

КВАЛІФІКАЦІЙНА РОБОТА

на здобуття освітнього ступеня

бакалавр

(назва освітнього ступеня)

на тему: Генератор QR-коду на основі STM32F4

Виконав(ла): студент(ка) IV курсу, групи СІс-44

спеціальності 123 «Комп'ютерна інженерія»

(шифр і назва спеціальності)

	<u>Баршай Р.М.</u> (підпис)	<u>Баршай Р.М.</u> (прізвище та ініціали)
Керівник	<u>Осухівськ</u> (підпис)	<u>Осухівськ</u> (прізвище та ініціали)
Нормоконтроль	<u>Луцик Н.С.</u> (підпис)	<u>Луцик Н.С.</u> (прізвище та ініціали)
Завідувач кафедри	<u>Осухівськ</u> (підпис)	<u>Осухівськ</u> (прізвище та ініціали)
Рецензент	<u>Млинко Б</u> (підпис)	<u>Млинко Б</u> (прізвище та ініціали)

Міністерство освіти і науки України
Тернопільський національний технічний університет імені Івана Пулюя

Факультет комп'ютерно-інформаційних систем і програмної інженерії
(повна назва факультету)

Кафедра комп'ютерних систем та мереж
(повна назва кафедри)

ЗАТВЕРДЖУЮ

Завідувач кафедри

Осухівська Г.М.

(підпис)

(прізвище та ініціали)

« 12 » 02
2021 р.

**ЗАВДАННЯ
НА КВАЛІФІКАЦІЙНУ РОБОТУ**

на здобуття освітнього ступеня бакалавр
(назва освітнього ступеня)

за спеціальністю 123 «Комп'ютерна інженерія»
(шифр і назва спеціальності)

Студенту Баршаю Роману Михайловичу
(прізвище, ім'я, по батькові)

1. Тема роботи Генератор QR-коду на основі STM32F4

Керівник роботи Осухівська Галина Михаліївна, к.т.м., доцент
(прізвище, ім'я, по батькові, науковий ступінь, вчене звання)

Затверджені наказом ректора від « 10 » 02 2021 року № 4/7-97

2. Термін подання студентом завершеної роботи 25.06.2021

3. Вихідні дані до роботи Затвердження технічного завдання

4. Зміст роботи (перелік питань, які потрібно розробити)
Вступ

1. Аналіз технічного завдання
2. Проектна частина

3. Практична частина
4. Безпека життєдіяльності, основи охорони праці

Висновок

5. Перелік графічного матеріалу (з точним зазначенням обов'язкових креслень, слайдів)

1 Структурна схема (A1)

2 Схема електрична принципова (A1)

3 Блок-схема алгоритму (A1)

4 Схема з'єднання та схема моделювання (A1)

АНОТАЦІЯ

Генерування QR коду за допомогою STM32F407VGT6-Discovery // Кваліфікаційна робота на здобуття освітнього ступеня бакалавр // Баршай Роман Михайлович // ТНТУ, спеціальність 123 «Комп'ютерна інженерія»// Тернопіль, 2021 // с.– 57, рис. – 18, табл. – 2, аркушів А1 – 4, бібліогр. – 21.

Ключові слова: вбудована система, STM32F407VGT6-Discovery, UART, SPI, Keil uVision, STM32cubeMX, Proteus.

У кваліфікаційній роботі бакалавра розроблено вбудовану систему для генерування QR коду за допомогою STM32F407VGT6-Discovery. На основі аналізу предметної області побудовано структурну схему, блок-схему, алгоритм роботи системи. Окрім цього, було створено схему з'єднання, моделювання в середовищі proteus та побудову пристрою.

В першому розділі зроблений аналіз технічного завдання та огляд існуючих систем та вимог до системи.

В другому розділі відбувається обґрунтування структурної схеми наводиться обґрунтування вибору елементної бази відбувається опис шин, протоколів які використовуються в проекті та опис схеми електричної принципової.

В третьому розділі відбувається програмна реалізація проекту опис алгоритму роботи програми, створення проекту в середовищі STM32CubeMX, компіляція програмного проекту в середовищі Keil та побудова проекту в середовищі Proteus.

ABSTRACT

Generate QR code using STM32F407VGT6-Discovery // Qualification work for obtaining a bachelor's degree // Barshay Roman Mikhailovich // TNTU, specialty 123 "Computer Engineering" // Ternopil, 2021 // p.-57, Fig. -18 , table. -2, Sheets A1 -4, Bibliogr. -21.

Keywords: embedded system, STM32F407VGT6-Discovery, UART, SPI, Keil uVision, STM32cubeMX, Proteus.

In the bachelor's qualifications, a built-in system for generating a QR code using STM32F407VGT6-Discovery has been developed. Based on the analysis of the subject area, the structural scheme, block diagram, algorithm of the system operation are built. In addition, a connection diagram, proteus simulation, and device construction were created.

The first section analyzes the terms of reference and an overview of existing systems and system requirements.

In the second section there is a construction of the built-in system, the substantiation of the structural scheme the substantiation of a choice of element base there is a description of tires, the protocols used in the project and the description of the scheme of electric basic.

In the third section there is a software implementation of the project, a description of the algorithm of the program, project creation in the STM32CubeMX environment, compilation of the software project in the Keil environment and construction of the project in the Proteus environment.

ЗМІСТ

ВСТУП.....	8
РОЗДІЛ 1 АНАЛІЗ ТЕХНІЧНОГО ЗАВДАННЯ.....	9
1.1 Характеристика об'єкта проектування	9
1.2 Аналіз вимог до генератора QR-коду.....	11
РОЗДІЛ 2 ПРОЕКТНА ЧАСТИНА	15
2.1 Розробка узагальненої структури генератора QR-коду	15
2.2 Обґрунтування вибору апаратного забезпечення генератора QR-коду	17
2.2.1 Огляд платформи STM32F407VGT6-Discovery	17
2.2.2 Огляд OLED display SSD1351	20
2.3 Схема електрична принципова генератора QR-коду.....	22
2.3.1 UART	23
2.3.2 SPI	24
2.4 Обґрунтування вибору програмного забезпечення генератора QR-коду	27
2.4.1 STM32CubeMX	31
2.4.2 Keil uVision	32
2.5 Опис алгоритму роботи програми генератора QR-коду	30
2.5.1 Блок-схема алгоритму роботи програми генератора QR-коду.....	33
РОЗДІЛ 3 ПРАКТИЧНА ЧАСТИНА	37
3.1 Реалізація проектних рішень.....	37
3.1.1 Схема з'єднання мікроконтролера з дисплеєм генератора QR-коду	37
3.1.2 Моделювання проекту в середовищі Proteus	38
3.1.3 Результати моделювання в середовищі Proteus	44

<i>КС КРБ 123.158.00.00 ПЗ</i>					
<i>Змн.</i>	<i>Арк.</i>	<i>№ докум.</i>	<i>Підпис</i>	<i>Дата</i>	
<i>Розроб.</i>		<i>Барисай Р.М.</i>			
<i>Перевір.</i>		<i>Осухівська Г.М.</i>			
<i>Рецензент</i>					
<i>Н. Контр.</i>		<i>Луцик Н.С.</i>			
<i>Затверд.</i>		<i>Осухівська Г.М.</i>			
<i>Генератор QR-коду на основі STM32F4</i>			<i>Літ.</i>	<i>Арк.</i>	<i>Акрушів</i>
				6	57
<i>ТНТУ, каф. КС, гр. СІс-44</i>					

3.1.4 Налаштування виводів мікроконтролера в середовищі STM32CubeMX.....	46
3.2 Тестування.....	46
3.2.1 Тестування методом чорного ящика.....	47
3.2.2 Тестування ПЗ на працездатність.....	47
РОЗДІЛ 4 БЕЗПЕКА ЖИТТЄДІЯЛЬНОСТІ, ОСНОВИ ОХОРОНИ ПРАЦІ .	49
4.1 Охорона праці.....	49
4.2 Безпека в надзвичайних ситуаціях.....	51
ВИСНОВКИ.....	55
БІБЛІОГРАФІЯ.....	56
Додаток А. Технічне завдання.	
Додаток Б. Перелік елементів.	
Додаток В. Лістинг коду.	
Додаток Г. Таблиця тестових випадків методом чорного ящика.	

					КС КРБ 123.158.00.00 ПЗ	Арк
Зм	Арк	№ докум	Підпис	Дата		7

ВСТУП

QR-код (Quick Response) – це знайдений близько двадцять три роки тому в Токіо двомірний зчитувач коду, що використовувався для кодування будь-якої інформації: текстів, веб-посилань, номерів телефонів та чимало іншого. QR-код безумовно уявлений у вигляді темно-білосніжного квадрата, що нагадує всім відомі лабіринти. В будь-якому QR-коді є можливість кодувати: 7088 чисел, та 4297 символів (також у вигляді кирилиці), 1818 ієрогліфу. Код може бути в будь-якій текстовій комбінації, яка може містити цифри та символи [1].

Основною перевагою QR-кода над простим штрих-кодом є можливість здійснювати сканування простою камерою мобільного телефона, а також - сканування широкого обсягу інформацій, що кодується [1].

Відтворення QR-коду можна здійснювати програмно (з використанням мікропроцесора або мікроконтролера) або апаратно (за підтримкою спеціального приладу). В такому випадку QR-код відскановується сенсорами як двомірне зображення. Три квадратних візерунки по кутах штрих-коду і маленькі синхронізуючі квадрати по даному коду дають можливість стабілізувати розміри зображень і його орієнтованість, а також кут, на якому розгорнутий сенсор до верхньої частини візерунка коду. А програма-розпізнавач дозволяє користувачу миттєво внести в свій мобільний телефон будь-який текст для запису інформації, контактну інформацію в книгу для адрес, переходити по веб-посиланнях або відсилати SMS-повідомлення, тощо.

Метою цієї кваліфікаційної роботи є розробка пристрою, який буде генерувати QR-код, побудованого на базі STM32F407VGT6-Discovery.

					КС КРБ 123.158.00.00 ПЗ	Арк
Зм	Арк	№ докум	Підпис	Дата		8

РОЗДІЛ 1 АНАЛІЗ ТЕХНІЧНОГО ЗАВДАННЯ

1.1 Характеристика об'єкта проектування

Об'єктом проектування є прилад генерації QR коду, який буде застосовуватися для динамічної генерації в реальному часі та стрімкого читання QR-коду.

Генератори QR-кодів на сьогоднішній день є досить таки популярними та отримали велике поширення.

Сферою використання таких пристроїв є платіжні термінали на базі QR-кодів для невеликої кількості користувачів, а також вони використовуються в банківських сферах, міні-магазинах, продовольчих ринках, кафе та інших місцях як для оплати послуг, так і для отримання інформації.

Для ознайомлення з генеруванням QR коду нижче описаний простий алгоритм генерування QR коду, який використовується на простих сайтах, задля ознайомлення з QR кодом.

Алгоритм процесу генерування QR-коду [2]:

- Кодування даних.
- Додавання службової інформації та заповнення.
- Розділення інформації на блоки.
- Створення байтів корекції.
- Об'єднання блоків.
- Розміщення інформації на QR-коді.

На першому етапі здійснюється вибір типу кодування. Існують такі типи кодування як: цифровий (тільки цифри), літерально-цифровий (де можна кодувати літери і цифри), кендзі (китайські, японські ієрогліфи) і побайтове кодування.

На другому етапі - здійснюється вибір рівня корекції. Всього існує чотири рівня корекції: L (із допустимим максимальним пошкодженням - 7%), M

					КС КРБ 123.158.00.00 ПЗ	Арк
Зм	Арк	№ докум	Підпис	Дата		9

(із допустимим максимальним пошкодженням -15%), Q (25%) і H (30%).

Більшоптимальний і часто застосовується - це M.

Також на другому етапі потрібно вибрати рівень версії, чим більша версія тим більший його розмір. Всього існує сорок рівнів.

На третьому етапі потрібно зробити розбиття інформації на блоки. При цьому всю кількість байт поділити на кількість блоків даних. Якщо це число не ціле, то треба визначити залишок від ділення.

Для створення байтів корекції на цьому етапі використовується алгоритм Ріда-Соломона. Це цикл, в якому підготовлюється масив, довжина якого дорівнює максимуму з кількості байтів в поточному блоці і кількості байтів корекції, і при цьому необхідно заповнити його початок байтами з поточного блоку, а кінець - нулями.

Далі здійснюється об'єднання блоків. Оскільки вже існує блок даних та блок корекції байтів, тому необхідно їх об'єднати в одну послідовність.

Завершальний етап - це розміщення інформації на QR-коді.

Одним із найпопулярніших аналогів пристроїв генерування кодів є модель Mf66s з двома екранами, встроєними NFC та калькулятором [3].

Mf66s - це платіжний термінал на базі QR-коду, розроблений Fujian Morefun Tecnologia Eletronica Co Ltd для невеликої кількості розрахунків. Підтримує сканування QR-коду, відображення платежу, пароля та підписки-доставки платежу NFC для здійснення невеликої кількості платежів за допомогою мобільного телефону. Він може обмінюватися даними з головним комп'ютером через дротовий інтерфейс USB і безпосередньо стикуватися з бездротовою платіжною платформою GPRS / WIFI, таким чином швидко виконуючи функцію оплати.

Mf66s широко використовується в міні-магазинах, продовольчих ринках, кафе та інших невеликих сценаріях оплати. Є простим у експлуатації, а вбудоване вікно сканування QR-коду на робочому столі - зручне для продавців для отримання платежів.

					КС КРБ 123.158.00.00 ПЗ	Арк
Зм	Арк	№ докум	Підпис	Дата		10

Модель Mf66s дуже зручна для генерування QR коду, завдяки своїм габаритам.

На рисунку 1.1 показаний загальний вигляд POS TERMINAL генерування коду Mf66s [3].



Рисунок 1.1 — POS TERMINAL генерування коду Mf66s.

Також ця модель оснащена такими підтримками як: читання та відображення QR-коду, створення динамічного QR-коду в режимі реального часу; безконтактна IC-картки, мобільного телефону NFC та ін., можливість передачі голосу.

1.2 Аналіз вимог до генератора QR-коду

Відповідно до технічного завдання, поставленого у кваліфікаційній роботі (Додаток А), генератор QR-коду повинен бути побудований на базі STM32F407VGT6-Discovery (оскільки саме такого типу плата є в матеріально-технічному забезпеченні навчального процесу на кафедрі КС ТНТУ).

					КС КРБ 123.158.00.00 ПЗ	Арк
Зм	Арк	№ докум	Підпис	Дата		11

Розроблений генератор QR-коду повинен забезпечувати функціонування таких основних режимів:

- вивід результату генерації QR-коду на екран;
- підключення клавіатури;
- підключення джерела живлення;
- згенерований QR-код.

Вивід результату генерації коду на екран – перший етап цього режиму і передбачає занесення даних до матриці, тобто побітове перетворення, яке отримується в наслідок читання даних, заноситься до матриці. Другим етапом цього режиму є перетворення матриці в картинку, тобто переведення матриці до розміру 21x21 пікселів. Третій етап цього режиму - це вивід QR коду на екран, тобто вже згенеровані пікселі виводять картинку QR коду, яка потрібна користувачеві.

Підключення клавіатури виконується за підтримкою послідовного порта UART. UART - це (Universal asynchronous receiver / transmitter) або УАП (універсальний асинхронізуєчий приймач) - найстаріший і найпоширеніший на сьогоднішній день фізичний протокол передачі даних. Стандартна конфігурація представляє собою швидкість обміну пакетів 115200біт/с даними по 8біт, в стандарта не має біта парності, але існує один стоп-біт [4].

Подання джерела живлень реалізується за підтримкою практичної послідовної шини USB mini. Це послідовний інтерфейс для під'єднання периферійних приладів до обчислювального обладнання. Заробив велике поширення і став головним інтерфейсом під'єднання периферії до повсякденного цифрового обладнання, кабель який задовільняє живлення в +5В. Як джерело живлення застосовують РС.

Розроблені технічні рішення з створення пристрою генерування QR коду повинне забезпечувати його функціонування в +3,3В. Задля цього застосовується імпульсний перетворювач напруги, який призначений для перебудови напруги з одного рівня в інший рівень. Частіше вживають

					<i>КС КРБ 123.158.00.00 ПЗ</i>	Арк
Зм	Арк	№ докум	Підпис	Дата		12

імпульсні перетворювальні напруги з застосуванням індуктивних накопичувачів енергії. Такі перетворювачі виділяються великим ККД, іноді - до 95%, і є змога здобути підвищений, знижений або інвертований сигнал вихідної напруги. В нашому варіанті - перетворення напруги використовується для зниження напруги з +5В до +3,3В.

На цьому етапі користувач отримує згенерований монохромний QR-код, який являє собою графічний об'єкт високої точності відтворення, і який прикрашений відповідними візерунками. Користувачі можуть відсканувати його за допомогою камери смартфонів або спеціальних сканерів QR-коду.

Вимоги до складових частин виробу, вихідних й експлуатаційних матеріалів такі: для виробництва пристрою повинні використовуватися матеріали імпортного виробництва, які можливо купити та замінити в Україні, або власна елементна база, яка розробляється в Україні.

В таблиці 1.1 зображені деякі технічні характеристики пристрою генерації QR-коду.

Таблиця 1.1 – Технічні характеристики QR-генератора.

Характеристики	Значення
Вага	Орієнтовно 400г
Ширина	200x100мм
Висота	30 мм
Швидкість роботи	10с
Живлення	+3,3В

Генератор QR-коду виконаний за невеликих розмірах. Його довжина та ширина не перевищує 300мм, що підходить для наших умов використання.

					КС КРБ 123.158.00.00 ПЗ	Арк
						13
Зм	Арк	№ докум	Підпис	Дата		

Пристрій QR-коду оснащений живленням в +3,3В. Він може відсканувати при невеликому навантаженні зі швидкістю в 10 секунд, що є оптимальною швидкістю для мікроконтролера STM32F407VGT6-Discovery.

QR-код генератор підтримує інтерфейси комунікації з зовнішніми пристроями через широко розповсюджені стандарти USB.

					<i>КС КРБ 123.158.00.00 ПЗ</i>	Арк
<i>Зм</i>	<i>Арк</i>	<i>№ докум</i>	<i>Підпис</i>	<i>Дата</i>		14

РОЗДІЛ 2 ПРОЕКТНА ЧАСТИНА

2.1 Розробка узагальненої структури генератора QR-коду

Одним з перших етапів при створенні девайсу є створення його структурної схеми, що позначається як головна функціональна частина проекту, взаємовідношення проекту та його наміри. Структурна схема розроблена для відтворення основної схематики приладу, інакше кажучи, його головні вузли, частини, блоки та основні зв'язки між ними, не звертаючи увагу на схему реалізації для функціональної частини.

Структурна схема забезпечує швидкий огляд системи на високому рівні для швидкого виявлення визначних місць або проблемних місць. Структурна схема не буде детально відображати кожен провід і комутатор, це завдання принципової схеми.

З структурної схеми має бути зрозумілим, для чого необхідний цей апарат і як він використовується в основних режимах праці, як співпадають його частини.

Найменування елементів структурної схеми обираються довільно, хоча широкоприйнятих правил роботи схем слід додержуватись.

При розробці генератора QR-коду була створена структурна схема, яка зображена на рисунку 2.1.

					КСКП 123.158.01.00 ПЗ			
<i>Змн.</i>	<i>Арк.</i>	<i>№ докум.</i>	<i>Підпис</i>	<i>Дата</i>				
<i>Розроб.</i>		<i>Барисай Р.М.</i>			<i>Генератор QR-коду на основі STM32F4</i>	<i>Літ.</i>	<i>Арк.</i>	<i>Акрушів</i>
<i>Перевір.</i>		<i>Осухівська Г.М.</i>					15	22
<i>Н. Контр.</i>		<i>Луцик Н.С.</i>			<i>ТНТУ, каф. КС, гр. СІс-44</i>			
<i>Затверд.</i>		<i>Осухівська Г.М.</i>						

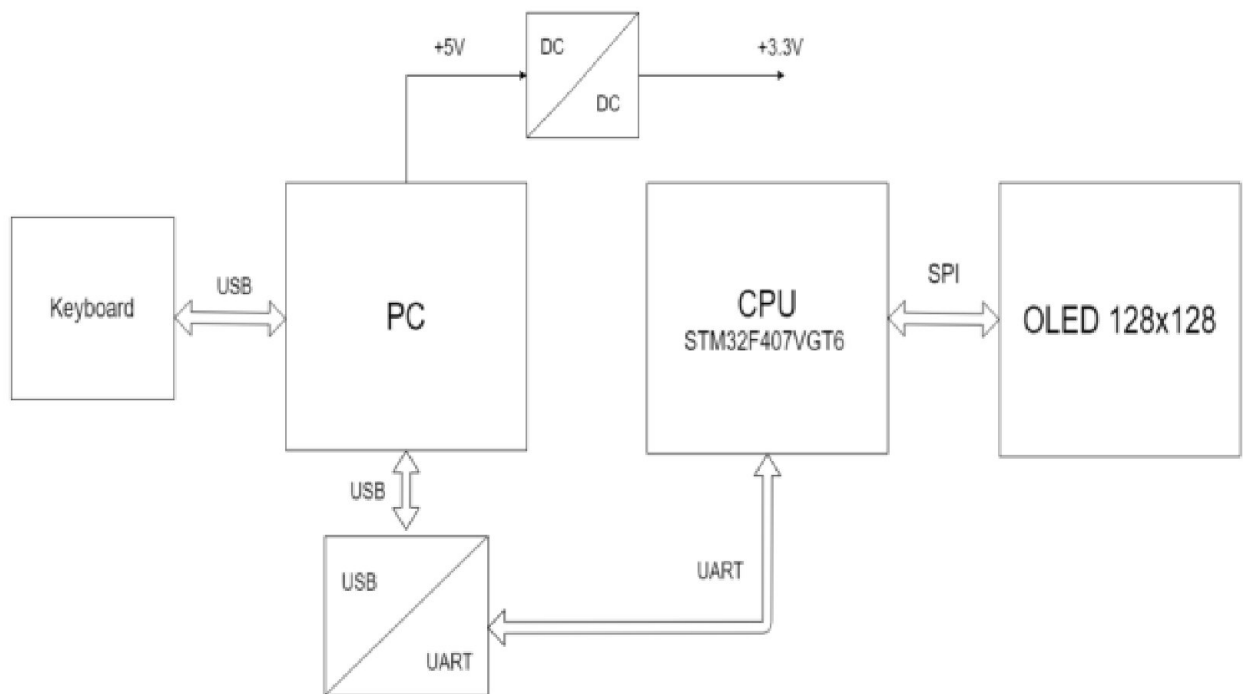


Рисунок 2.1 – Структура схема генератора QR-коду

Для правильної роботи потрібно визначити як повинен працювати кожен блок. Отже, відповідно до запропонованої структурної схеми:

– PC - блок який є основним ресурсом для живлення всієї системи (генератора QR-кодів), також використовується для підключення клавіатури, перетворювача інтерфейсів та подання живлення на мікроконтроллер.

– Keyboard - блок що використовується для подання цифрових даних на PC, тобто подання набору текстової інформації на нашу систему.

– DC/DC - це перетворювача інтерфейсів, що використовується для пониження поданого живлення з PC (з +5В до +3,3В).

– USB/UART - блок використовується для підключення клавіатури та джерела живлення до CPU. Для підключення клавіатури використовується послідовний порт UART, а для подання живлення - універсальна послідовна шина USB mini.

					КС КРБ 123.158.00.00 ПЗ	Арк
Зм	Арк	№ докум	Підпис	Дата		16

– CPU - головний блок, мікроконтролер що використовується для підключення дисплею, також він використовується для обробки пакету та подання їх на OLED-екран.

– OLED 128x128 - вихідний блок вміщає відображення результатів генерації QR-коду. Він повинен мати невеликі розміри і після його генерування, користувач повинен мати можливість відсканувати його.

2.2 Обґрунтування вибору апаратного забезпечення генератора QR-коду.

Для реалізації апаратної частини генератора QR-коду було обрано наступні елементи (детальніший список елементів наведено в додатку Б):

- Мікроконтролер STM32F407VGT6.
- Дисплей SSD1351.
- Роз'єм 47346-0001.
- Мікросхема MCP2200T-I/SS.
- Перетворювач LM37.

Для забезпечення нормального функціонування електричної схеми також були використані два кварцових резонаторів, конденсатори та резистори.

2.2.1 Огляд платформи STM32F407VGT6-Discovery

Для розробки нашого проекту вибраний мікроконтролер STM32F407VGT6, який виконує необхідні функції для підтримки передачі даних (підтримка UART, SPI або I2C інтерфейсів), які будуть оброблюватись за допомогою мікроконтролеру. А також необхідно зауважити, що саме такого типу плата є в матеріально-технічному забезпеченні навчального процесу на кафедрі КС ТНТУ, тому саме на базі неї передбачено реалізувати генератор QR-коду.

Серія STM32 F4 - це перша група мікроконтролерів STM32, в якій база ядра ARM Cortex-M4F. Серія F4 також є першою серією STM32, яка має DSP і інструкції з плаваючою комою. F4, за висновками, сумісний з серією STM32 F2

					<i>КС КРБ 123.158.00.00 ПЗ</i>	Арк
Зм	Арк	№ докум	Підпис	Дата		17

і додає вищу тактову частоту, статичне ОЗУ ССМ 64 КБ, повнодуплексний I²S, поліпшені годинник реального часу і більш швидкі АЦП.

STM32F407VGT6 включає високошвидкісну вбудовану пам'ять (flash-memoу до 1Мбайт, до 192Кбайт SRAM), до 4Кбайт зарезервованої SRAM, а також великий спектр розширеного входу та виходу на периферійних пристроях, що підключені по двох шинах APB, або трьох шин AHB з 32-бітною мультиматричною шиною AHB [5].

Мікроконтролер STM32F407VGT6 має наступні параметри:

- Максимальна тактова частота: 168МГц.
- Розмір програмної пам'яті: 1мБ.
- Розмір RAM пам'яті: 192кБ.
- Порт вводу-виводу тільки: 80.
- Таймери ШІМ: 17.
- Максимальний струм з піна або на пін: 2мА.
- Робоча температура: -40...+85°C.

Особливості мікроконтролеру STM32F407VGT6:

– Маленький та вишуканий зовнішній вигляд, зручний для різноманітних проектів, що виготовляються своїми руками.

– Сумісний з двома процесорами: STM32F407VGT6 та STM32F407VET6.

– Основна частота процесора досягає 168 МГц, що є швидким і потужним.

– Картка SD може розширити простір для зберігання.

– Підтримка завантаження SWD та послідовного порту, легка налагодження.

– Розширення порту GPIO, щоб зручно підключати різні периферійні пристрої.

– Підтримка використання шини FSMC для керування РК-дисплеєм, висока швидкість відображення.

					КС КРБ 123.158.00.00 ПЗ	Арк
Зм	Арк	№ докум	Підпис	Дата		18

Галузь, де використовують STM32F407VGT6 - це управління двигунами для медичного обладнання, промислових пристроїв: програмованих та логічних контролерів, інверторів і переривники ланцюгів, принтерів та сканерів, різні системи сигналізації, відеотехніка, в пристроях клімат-контролю, побутова аудіотехніка, концентратор датчиків мобільних телефонів та ін.

STM32F407VGT6-DISCOVERY – недорогий відладочний комплект для початку роботи з популярним сімейством мікроконтролерів STM32F1x на основі ядра ARM Cortex – M3 від одного з світових лідерів з виробництва електронних компонентів – STMicroelectronics. Внутрішньохемний програматор/відладчик для STM8L, STM8S и STM32 компанії STM.

Апаратна складова мікроконтролера STM32F407VGT6-Discovery зображена на рисунку 2.2 [6].

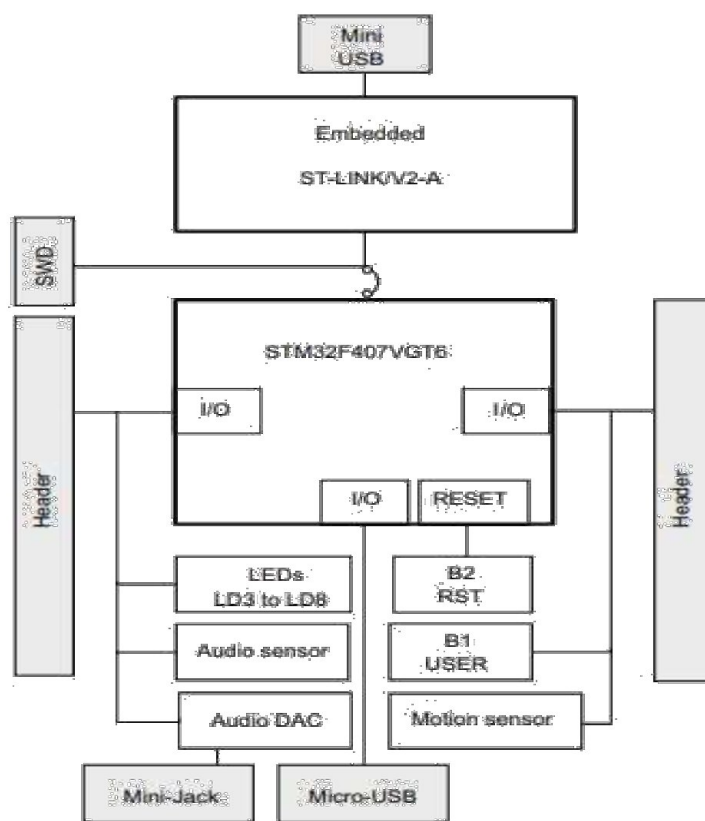


Рисунок 2.2 – Апаратна блок-схема STM32F407VGT6-Discovery

Засоби, які можуть застосовуватись для програмної реалізації мікроконтролера STM32F407VGT6-DISCOVERY:

- Keil nVision: MDK-ARM.
- IAR: EWARM.
- IDE на основі GCC, включаючи безкоштовний SW4STM32 від AC6.
- Arm® Mbed™ онлайн.

Програмне забезпечення, що застосовується для демонстрації, яке включене в пакет STM32Cube MCU, попередньо завантажується в флеш-пам'ять STM32 для легкого тактування периферійних пристроїв в автономному режимі. Новіші версії можна загрузити з веб-сайту виробника [5].

2.2.2 Огляд OLED-display SSD1351

SSD1351 - це CMOS-драйвер OLED/PLED з 384 сегментами та 128 загальними вихідними даними, що підтримує матричний дисплей до 128RGBx128 точок. Ця мікросхема призначена для загального катоду типу OLED/PLED панелі [7].

SSD1351 має вбудовану пам'ять графічного відображення даних (GDDRAM). Він підтримує 8, 16, 18 біт 8080/6800 паралельний інтерфейс, та послідовний периферійний інтерфейс. Він має 256-ступінчасту контрастність і 262К управління кольором, даючи яскравий кольоровий дисплей на OLED-панелях.

Роздільна здатність: 128RGBx128 матрична панель.

Глибина кольору 262тис. Кольорів, підтримувана вбудованим 128x128x18-бітним буфером відображення SRAM.

Вигляд OLED-display SSD1351 можна побачити на рисунку 2.3 [8].

					КС КРБ 123.158.00.00 ПЗ	Арк
Зм	Арк	№ докум	Підпис	Дата		20

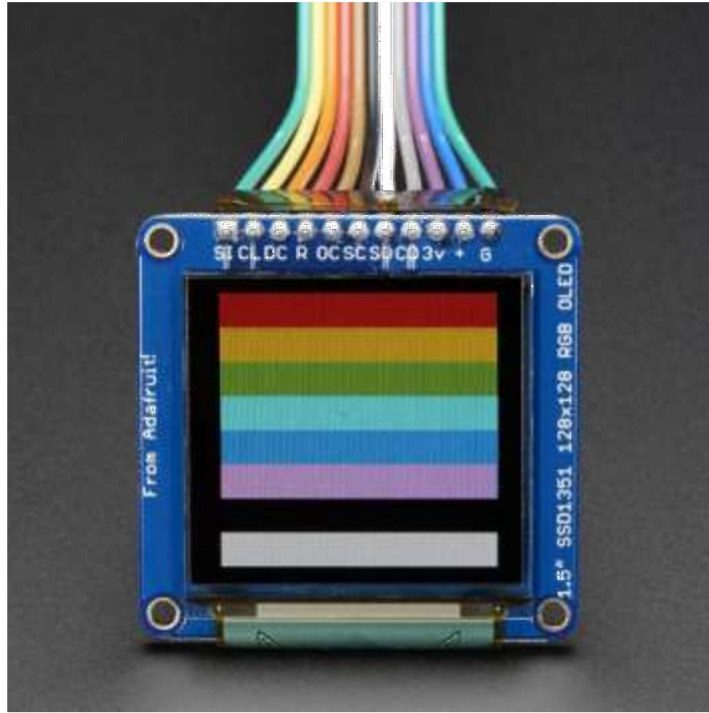


Рисунок 2.3 – SSD1351 OLED-display

MCU Serial Interface (3-провідниковий SPI) складається з:

- послідовного годинника SCLK;
- послідовних даних SDIN та CS #.

У 3-проводному режимі SPI D0 діє як SCLK, D1 - як SDIN. Для невикористаних контактів даних D2 слід залишити відкритим.

Виводи від D3 до D17, R / W # (WR #), E (RD #) та D / C # можуть бути підключені до зовнішнього заземлення.

Операція подібна до 4-провідного послідовного інтерфейсу, тоді як штифт D / C # не використовується. Всього є 9-бітові волі бути зміщеними в регістр зсуву на кожному дев'ятому годиннику послідовно: D / C # біт, D7 до D0 біт Біт D / C # (перший біт послідовних даних) визначатиме наступний байт даних у регістрі зсуву, який записується на дисплей. Детально можна побачити на рисунку 2.4 [7].

					КС КРБ 123.158.00.00 ПЗ	Арк
Зм	Арк	№ докум	Підпис	Дата		21

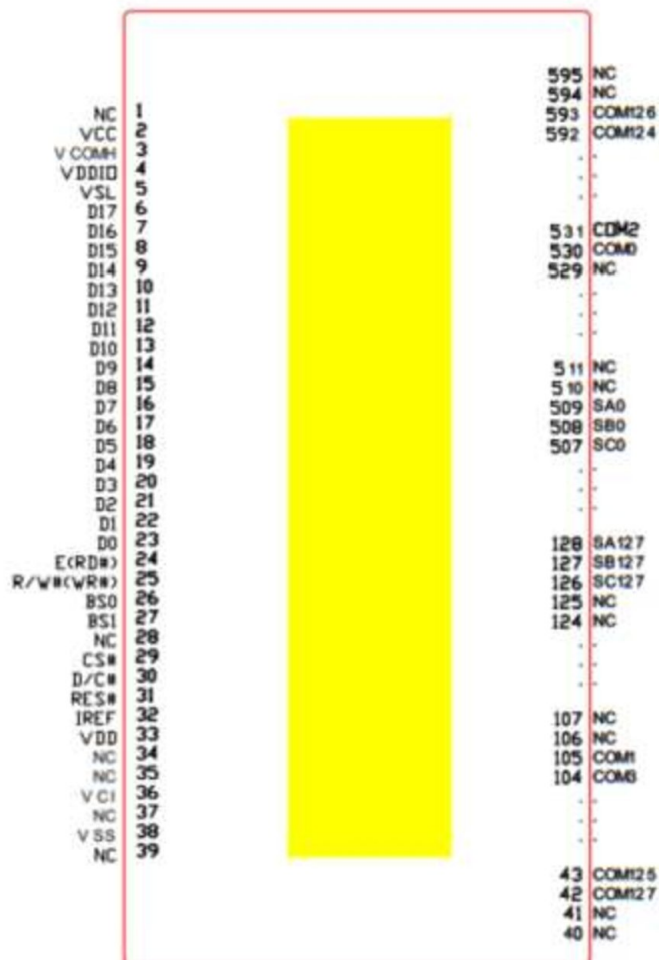


Рисунок 2.4 – Схема пінів дисплея SSD1351

Оперативна пам'ять даних ($D / C \# \text{ bit} = 1$) або регістр команд ($D / C \# \text{ bit} = 0$). У послідовному режимі виконуйте лише операції запису що дозволені в цій системі.

2.3 Схема електрична принципова генератора QR-коду.

Схема електрична принципова – графічна ілюстрація ескізу, за підтримкою умовного графічного та буквеного-цифрового позначення, взаємозв'язків серед елементів електронного гаджету. Вона показує на те, які компоненти з якими взаємозв'язуються. Як правило, при побудові приладу,

процес формування схеми електричної принципової є проміжною ланкою між стадіями розробки функціональної схеми і проектуванням друкованої плати.

Електрично-принципова схема є схожа з картою в якій є наявність електричних з'єднань. Застосування схеми електричної принципової не просто дає розуміння про проект, але і дає інформацію, як на її основі побудувати схеми відокремлених з'єднань, реалізувати вдосконалення конкретних вузлів. За нею ж і здійснюється перевірка постановки правильного монтажу для електронного обладнання [9].

Принципова електрична схема призначена для цілого відображення зв'язків між пристроями з врахуванням принципів їх дії і черговості роботи.

Розробка електричної принципової схеми QR генератора (креслення КС КРБ 123.158.00.00 ПЗ) здійснювалася відповідно до структурної схеми (див.рис.2.1) і зображена на рисунку 2.5. Перелік елементів наведено в ДодаткуБ.

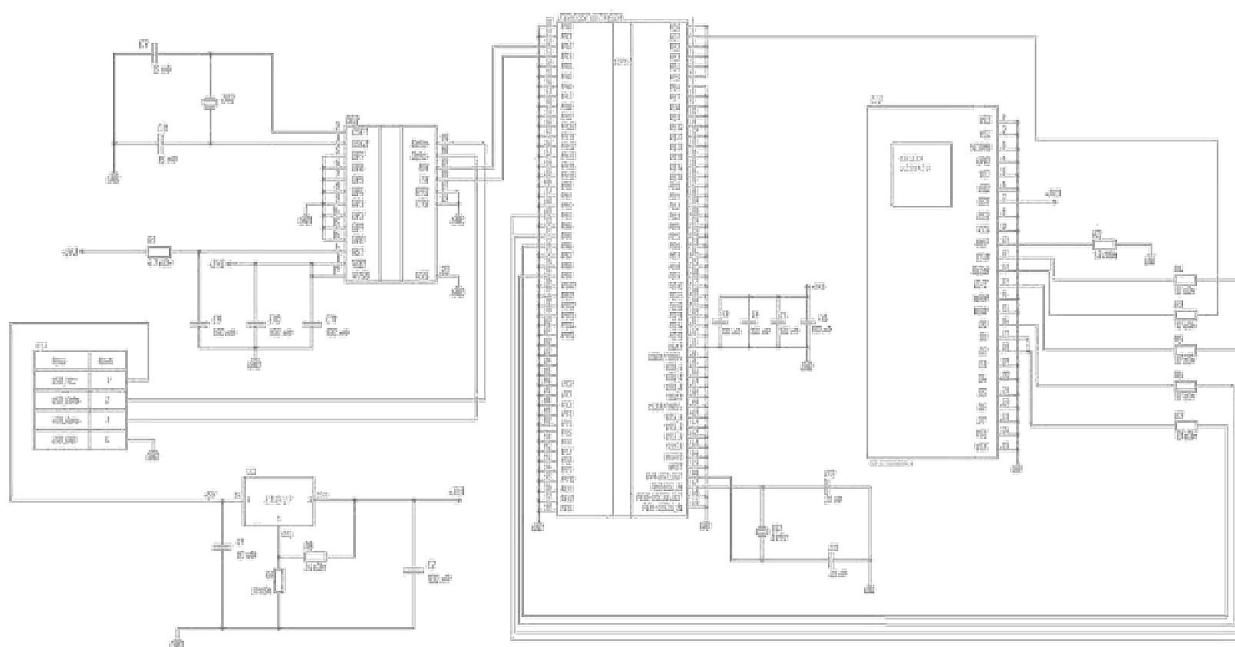


Рисунок 2.5 – Схема електрична принципова генератора QR-кода.

					КС КРБ 123.158.00.00 ПЗ	Арк
Зм	Арк	№ докум	Підпис	Дата		23

STM32F407VGT6-DISCOVERY, який забезпечує необхідну функціональну підтримку передавання пакетів (також існує підтримка UART, SPI інтерфейсу), які будуть оброблюватись за допомогою мікроконтролера STM32F407VGT6.

2.3.1 UART

UART - узагальнений асинхронний приймач / передавач, в якого тип асинхронного приймання та передавачі, необхідних компонентів для РС та периферійних приладів, також має можливість передачі пакетів в послідовній та паралельній між собою формі.

UART це проста відокремлена частина мікросхеми або звичайна мікросхема, що застосовується для взаємозв'язку між комп'ютером або периферійним послідовним портом. Взагалі UART у цілому включений до мікроконтролерів.

У цілому UART формується з таких компонентів [4]:

- логіка для контролю читання або запису;
- тактові генератори, частота яких як правило може бути кратною бітрейту, щоб допустити цикл передавання біта на центральну частину дискретизації;
- зсувні регістри, що використовуються для вводу або виводу;
- схеми, що контролюють передачу або прийом пакетів;
- буферна пам'ять стеку під назвою FIFO (опціонально);
- буферне передавання або приймання пакету (опціонально);
- буфери паралельних шин даних (опціонально).

UART - це логічна схема, яка з одного боку використовує підключення до шини розрахункового приладу, а з іншої сторони - має два або більше виходів для поверхневого з'єднання.

UART може бути як окрема мікросхема (приклади схем Intel I8251, I8250) або бути великими частинами великої інтегральних схем (прикладом

					КС КРБ 123.158.00.00 ПЗ	Арк
Зм	Арк	№ докум	Підпис	Дата		24

можуть бути мікроконтролери), що застосовується для передавання пакетів через послідовний порт РС.

Передача і прийом послідовних даних здійснюється за так: UART бере байт пакету і передає окремі біти у вигляді певної послідовності. Кожен UART містить регістр зсуву, який є основним методом перетворення між послідовною і паралельною формою. Послідовна передача цифрової інформації (бітів) через один провід або інший носій є менш дорогим, ніж паралельна передача по декільком проводам.

UART зазвичай не створює або приймає безпосередньо зовнішні сигнали, що використовуються між різноманітними елементами устаткування. Окремі інтерфейсні пристрої використовуються для логічного переведення рівнів сигналів UART. Зовнішні сигнали можуть мати безліч різних форм. Приклади стандартів для передачі пакетів напруги є RS-232, RS-422 і RS-485 від EIA. Деякі мікросхеми сигналізації не застосовують електричні дроти. Прикладом може бути оптоволоконні кабелі, IrDA (інфрачервоні), і Bluetooth (бездротові) в Serial Port Profile (SPP). Деякі мікросхеми сигналізації маніпулюють несучими модуляціями сигналів (з або без проводів). Це може бути модуляція звукових сигналів з телефонної лінії модемів, ВЧ модуляції з радіостанціями даних, і DC-LIN для зв'язку по лінії електропередачі.

Контакт може бути симплексний (тільки в одному напрямку, без надання прав для приладу одержувача, щоб забрати інформацію назад на передавальний прилад), застосовується повний дуплекс (взаємодіють обидва пристрої передавання і приймання пакету одночасно) або полудуплекс (пристрій передає і приймає по черзі).

2.3.2 SPI

Послідовний периферійний інтерфейс (SPI) - це інтерфейсні шини, які зазвичай служать для передавання пакетів між CPU та невеликими периферійними приладами, такими як регістри зсуву, датчики та SD карти. Він застосовує окремі лінії таймера та даних разом із виділеною лінією для вибору пристрою, з яким ви хочете поговорити.

					КС КРБ 123.158.00.00 ПЗ	Арк
Зм	Арк	№ докум	Підпис	Дата		25

SPI працює дещо по-іншому. Це «синхронна» шина даних, що означає, що вона використовує окремі рядки для передачі даних та «таймер», який забезпечує ідеальну синхронізацію обох сторін. «Таймер» - це коливальний сигнал, який повідомляє приймачу, коли саме потрібно відбирати біти на лінії передачі даних. Це може бути зростаючий (від низького до високого) або спадаючий (від високого до нижчого) краю тактового сигналу; у таблиці буде вказано, яку з них використовувати. Коли приймач виявить цей край, він негайно перегляне рядок даних, щоб прочитати наступний біт. Оскільки час надсилається разом із даними, вказівка швидкості не є важливою, хоча пристрої матимуть максимальну швидкість, з якою вони можуть працювати.

Однією з причин популярності SPI є те, що приймальне обладнання може бути простим регістром зсуву. Це набагато простіше і дешевше апаратне забезпечення, ніж повноцінний UART (універсально асинхронний передавач або приймач), який вимагає асинхронний послідовний пристрій.

У SPI лише одна сторона генерує тактовий сигнал (зазвичай називається CLK або SCK для послідовного Clock). Сторона, яка генерує таймер, називається "контролером", а інша - "периферійною". Завжди є лише один контролер (який майже завжди є вашим мікроконтролером), але периферійних пристроїв може бути декілька.

Зверніть увагу, що SPI є "повнодуплексним" (має окремі лінії надсилання та прийому), і, отже, у певних ситуаціях можна одночасно передавати та отримувати пакети (наприклад, вимагаючи нового зчитування датчика під час отримання пакетів з попереднього).

В даній специфікації для передавання пакетів служать чотири сигнали:

- MOSI чи SI — ведучий вихід, ведучий вхід. Використовується для передавання пакетів від ведучого компоненту до головного.
- MISO чи SO — ведучий вхід, ведучий вихід. Застосовується для передавання пакетів від головного приладу до ведучого.

					КС КРБ 123.158.00.00 ПЗ	Арк
						26
Зм	Арк	№ докум	Підпис	Дата		

– SCLK чи SCK — це послідовно-тактовий сигнал. Використовується як передавач тактових сигналів для ведення до CPU.

– CS чи SS — сигнал, який відповідає початку завершення процедури зв'язку. По закінченню обміну - пакет повинен бути змінений, що дозволить приймачеві пакету піти з режима прочитаних пакетів або занесення, до режима опрацювання пакету [10].

2.4 Обґрунтування вибору програмного забезпечення генератора QR-коду.

Для попереднього налаштування мікроконтролера і ініціалізації початкового коду для різних середовищ розробки, використано середовище STM32CubeMX [11].

Для програмної розробки, що представляє собою набір утиліт для виконання повного комплексу заходів з написання програмного забезпечення для мікроконтролерів, та базу даних мікроконтролерів, що містять докладну інформацію про всі підтримувані пристрої було обрано середовище Keil uVisio [13].

2.4.1 STM32CubeMX

STM32CubeMX - це генератор вихідного коду для мікроконтролера STM32. Що дає змогу налаштувати периферію за допомогою графічного інтерфейсу [12].

Якщо створити проект в STM32CubeMX, вибрати мікроконтролер серії F1 і згенерувати вихідний код, не проводячи ніяких налаштувань, то цей код відключить інтерфейси програмування JTAG і SWD. Якщо таку програму скомпілювати і прошити, то в наступний раз прошити мікроконтролер буде важко.

Візуалізуючий редактор, що використовується для налаштувань STM32CubeMX і що є складовим компонентом інфраструктур STM32Cube,

					КС КРБ 123.158.00.00 ПЗ	Арк
Зм	Арк	№ докум	Підпис	Дата		27

який теж включає спеціальні бібліотечні компоненти для роботи з різноманітними серіями мікроконтролера STM32. Також відзначають, що інтегрування між компонентами виконуються вдало, оскільки встановити необхідну бібліотеку можливо прямо з головного редактора. Для цього потрібне підключення до Інтернету, оскільки завантаження проводиться з сайту компанії STMicroelectronics. Також є можливість відслідковувати оновлення версій вже завантажених бібліотек. Крім того, дані бібліотеки мають можливість застосовуватись самотужки при розробці в іншому програмному забезпеченні.

Програмний пакет має в себе такі головні компоненти [11]:

- бібліотека для взаємодії з периферійним пристроєм (Hardware
- компонент, що спрощує виконання всіх здатностей відомих мікросхем
- компонент бібліотек CMSIS;
- бібліотека високих прикладних рівнів, які дають можливість реалізувати спеціальні здатності мікросхеми такі як: системи реального часу, робота з файлами FAT та роботу з стеками протоколів TCP / IP за допомогою – компонента lwIP. Не всі бібліотеки даного рівня можуть бути доступні для підключення при використанні конкретного мікроконтролера.

Особливо можна відзначати, що є можливість роботи з бібліотекою HAL, що є доповнена версія популярної бібліотеки SPL, з якої HAL бере спільну концепцію для роботи з програмним кодом. Також архів з бібліотеками містить документацію та різні приклади кодів та корисні додатки.

Головні можливості, які надає візуальний редактор конфігурації, є:

- настройка використання висновків периферійними пристроями в складі мікроконтролера;
- настройка тактування;
- периферійне налагодження пристрою;

					<i>КС КРБ 123.158.00.00 ПЗ</i>	Арк
						28
<i>Зм</i>	<i>Арк</i>	<i>№ докум</i>	<i>Підпис</i>	<i>Дата</i>		

– споживання струму та його розрахунок за планованим режимом роботи.

Вигляд інтерфейсу ПЗ є дуже зрозумілий, простий та зручний. Без погляду на існування можливості розширеного налагодження, інтерфейс не має непотрібних функцій, а візуальна уява про різні системи МК дає повну картину про кінцеві результати виконаних редагувань [12].

Для імпорту роботи на середовище розробки користувач налаштовує проект в меню, також є можливість вибору ланцюжка утиліти для побудови проекту в середовищі IDE, для якого мусить бути створений програмний код C. STM32CubeMX гарантує, що дані проекту, що відповідають IDE, додаються до папки з проектом, щоб він міг бути безпосередньо імпортований як новий проект у сторонній IDE (IAR EWARM, Keil MDK ARM, Atollic TrueSTUDIO та AC6 System Workbench для STM32) [12].

2.4.2 Keil uVision

Keil uVision – це середовище розробки, що представляє собою набір утиліт для виконання повного комплексу заходів з написання програмного забезпечення для мікроконтролерів [13].

Keil uVision, популярна IDE від Keil Software, поєднує проект управління, редагування вихідного коду, налагодження програм та Flash програмування в єдиному потужному середовищі, управління проектами, налаштування пристрою та конфігурація інструменту. Налагодження файлів або моделювання процесора та периферії.

Для досвідчених користувачів μ Vision додає нові функції, такі як структура, навігація функціями, шаблони редакторів, додатковий пошук, майстер конфігурації, аналізатор логіки, моделювання CAN та I C, flash програмування та налагодження JTAG.

Можливості редактора для створення, модифікації та виправлення програм такі: Keil uVision забезпечує легкий та послідовний проект управління. Зберігаються файли одного проекту, такі як імена вихідних файлів, що зберігають конфігурацію інформації для compiler, Assembler, Linker, Flash

					КС КРБ 123.158.00.00 ПЗ	Арк
Зм	Арк	№ докум	Підпис	Дата		29

Loader та інших утиліт. Меню project забезпечує доступ до проекту файли та діалогові вікна для управління проектами. Open project ... завантажує існуючі файли проекту. Папка "example" містить багато корисного зразки додатків, як приклад Measure [13].

New Project ... дозволяє створити ваш власні проекти.

Платформа дозволяє переглядати та змінювати обраний цільовий процесор. Також дозволяє вказати процесор і параметри пам'яті та дозволяє налаштувати згенерований об'єкт та файли списку.

Keil uVision дозволяє встановити параметри інструменту за допомогою головного, групового та вихідного файлів. Варіанти діалогового вікна відкриваються через контекстне меню [13].

Проект робочої області та забезпечує вкладки для налаштування інструменту, включаючи лінкера, налагоджувача та симулятора. Як правило, це єдині налаштування необхідно зробити, щоб розпочати новий проект.

Налагодження - дозволяє налаштувати Vision Simulator та налагоджувач.

Службові програми - дозволяють налаштовувати конкретні проекти (наприклад, програмування Flash).

Редактор Keil uVision пропонує багато стандартних і розширені функції редагування програмного забезпечення. Колір виділення синтаксису та відступ тексту оптимізовано для редагування вихідного коду C. Більшість функцій редактора можна швидко отримати за допомогою панелі інструментів або в контекстному меню.

Програма налагодження безпосередньо контролюється в редакторі. Це середовище дозволяє швидко інтегрувати та перевіряти зміни вихідного коду.

Keil uVision містить потужні функції та інтерфейси до зовнішніх інструментів.

Вбудований браузер джерел (доступний для Keil Cx51, C251 та C166) підтримує базу даних програмних символів для швидкої навігації по вихідних

					<i>КС КРБ 123.158.00.00 ПЗ</i>	Арк
						30
<i>Зм</i>	<i>Арк</i>	<i>№ докум</i>	<i>Підпис</i>	<i>Дата</i>		

файлах. Можна використовувати детальну інформацію для оптимізації типів пам'яті змінних.

Keil uVision для ARM забезпечує виконання профілі, який записує статистику виконання для кожної інструкції процесора, а також дозволяє переглянути кількість або час виконання в редакторі та розбирання вікон. Вигляд редактора надає огляд вихідного коду, тобто можна використовувати інформацію, зібрану виконувачем профайлера, щоб визначити, де програма проводить більшу частину свого часу [13].

2.5 Опис алгоритму роботи програми генератора QR-коду.

Для комп'ютерних систем алгоритмами є списки детальних інструкцій, що втілюють різні процеси обчислень, який починається з першого стану, здійснюється крізь наступність логічних станів, і закінчується заключним станом. З попереднім переходом до наступного переходу стан не обов'язково повинен бути детермінований – декілька алгоритмів можуть містити елементи випадковості [14].

Алгоритм представляє собою спеціалізовану блок-схему високого рівня, що використовується в техніці, використовується для проектування нових систем або для опису та вдосконалення існуючих. Його структура забезпечує огляд на високому рівні основних компонентів системи, ключових елементів процесу та важливих взаємозв'язків.

Розроблена система зобов'язана мати зрозумілий, сформований алгоритм, який дозволяв би миттєво працювати в програмі та безперервно опрацьовувати вхідні і вихідні дані.

Алгоритму роботи програми генератора QR-коду такий: на початку відбувається опрацювання пакету від клавіатури та подання живлення; наступний етап - це завантаження даних на пристрій, тобто ввід тексту з

					КС КРБ 123.158.00.00 ПЗ	Арк
Зм	Арк	№ докум	Підпис	Дата		31

клавіатури; і завершальним етапом є генерація QR-коду, яка здійснюється натисненням на кнопку.

Алгоритм програми представлено у вигляді блок-схеми (креслення КС КРБ 123.158.00.00).

2.5.1 Блок-схема алгоритму роботи програми генератора QR-коду.

На початку відбувається опрацювання пакету від клавіатури, опрацювання пакету використовується для вводу даних з клавіатури.

Потім випадає можливість для вводу даних, тобто вже під'єднане живлення, подані дані з клавіатури і є можливість для вводу даних до рядку.

Наступним етапом відбувається перевірка на нецензурні слова, тобто в даній системі розроблений алгоритм, який робить перевірку на погані слова. Якщо введено нецензурне слово, то алгоритм перекидає користувача до початкового рядку вводу даних, якщо ж все вірно - програма пропускає до наступного етапу.

Далі здійснюється перевірка на кількість введених символів: процедура робить перевірку на правильність вводу символів. Якщо кількість введених символів більше 20, то викине повідомлення про коректність вводу символів, якщо ж кількість менша 20 - то цикл пропускає до наступного етапу.

На початковому етапі даний масив потрібно перетворити у послідовність, де перші 4 біти зберігають тип кодування, наступні 2 байти зберігають обсяг інформації, що зашифрована в коді — це звичайне двійкове число.

Головна блок схема поділена на дві частини, алгоритм програми зображений на рисунках 2.6 та 2.7.

					<i>КС КРБ 123.158.00.00 ПЗ</i>	Арк
<i>Зм</i>	<i>Арк</i>	<i>№ докум</i>	<i>Підпис</i>	<i>Дата</i>		32



Рисунок 2.6 — Блок-схема алгоритму генератора QR коду



Рисунок 2.7 — Блок-схема алгоритму генератора QR коду.

Другий етап - це визначення типу даних, оскільки наш пристрій читає тільки числові і стрічкові дані, то, відповідно, цих вимог користувач повинен дотримуватись. Відповідно цих вимог розроблений алгоритм, який повертає користувачів назад, якщо введений некоректно тип даних.

Побітове перетворення - етап на якому пристрій виконує переведення бітів до пікселя. Цей процес називається бітовою глибиною – що використовується для визначення обсягу даних про колір, присутній для всіх пікселів зображення. Велика чисельність бітів даних на піксель вказує на велику чисельність допустимих забарвлень та їх точне зображення на картинці, що можуть впливати на розміри зображень. Інакше кажучи, зі приростом бітової глибини розміри файла збільшують, тому що можна зберегти багато даних про колір в пікселі на даному зображенні [2].

Занесення даних до матриць - це процес відбувається так: повнокольорове зображення генерують за допомогою трьох світлодіодів, які містять такі забарвлення: червоне, зелене та синє, в матриці чи кластері, для утворення поодиноких пікселів (точок) [15].

Матриця світлодіодного дисплея - це схема, яка керується за інтегрованим світлодіодом, комутаційним контактером і захисним фільтром. Світлодіоди в матриці розміщені в суворій послідовності з рівними інтервалами в горизонтальному порядку (називають піксельним кроком), і рівними інтервалами (відстань між лініями світлодіодів по вертикалі також ідентичні).

Елементи світлодіодних екранів називають кластерами, вони демонструють окремі герметичні корпуси, в яких існують три кольори (зелений, червоний та синій) і більш світлодіодів. Із зовнішнього боку кластер прикритий пластиковою лінзою. Також потрібно зазначити, що в кластерних світлодіодних дисплеях кластером називають поодиначні пікселі [15].

Дозвіл світлодіодного екрану - щільний піксель, який використовує одиницю на поверхні. Чим кращі ці дані, тим якісніше зображення.

					<i>КС КРБ 123.158.00.00 ПЗ</i>	Арк
Зм	Арк	№ докум	Підпис	Дата		35

Перетворення матриці в QR зображення відбувається за набирання модулів, що створюють поле світлодіодного екрана.

Мікроконтролер, як система, яка керує світлодіодними екранами що "розбиває" цілу картинку на маленькі "ділянки". Кількість цих «ділянок» рівноцінна кількості світлодіодних модулів екрану і посилає крізь маленькі контролери, для відбиття, кожному "ділянку", яка відповідає світлодіоду модуля. Отже, після цих операцій, формують на світлодіодному дисплеї всю картинку.

Відокремлений модуль світлодіодів, як матриця, утворюється з піксельних світлодіодів, які і є випромінюючими елементами екрану світлодіоду, з яких і формується повне зображення.

В завершальному етапі пристрій виводить QR-код на екран, в результаті чого користувач може відсканувати його за підтримкою відскановувача QR-коду.

					<i>КС КРБ 123.158.00.00 ПЗ</i>	Арк
<i>Зм</i>	<i>Арк</i>	<i>№ докум</i>	<i>Підпис</i>	<i>Дата</i>		36

РОЗДІЛ 3 ПРАКТИЧНА ЧАСТИНА

3.1 Реалізація проектних рішень

3.1.1 Схема з'єднання мікроконтролера з дисплеєм генератора QR-коду.

Схема підключення - це просте візуальне зображення фізичного з'єднання та фізичної електричної схеми системи. Він показує, як електричні дроти взаємопов'язані, а також може показувати, де кріплення та компоненти можуть бути підключені до системи.

З'єднання мікроконтролера STM32F407VGT6-Discovery з дисплеєм SSD1351 пристроями зображено на рисунку 3.1

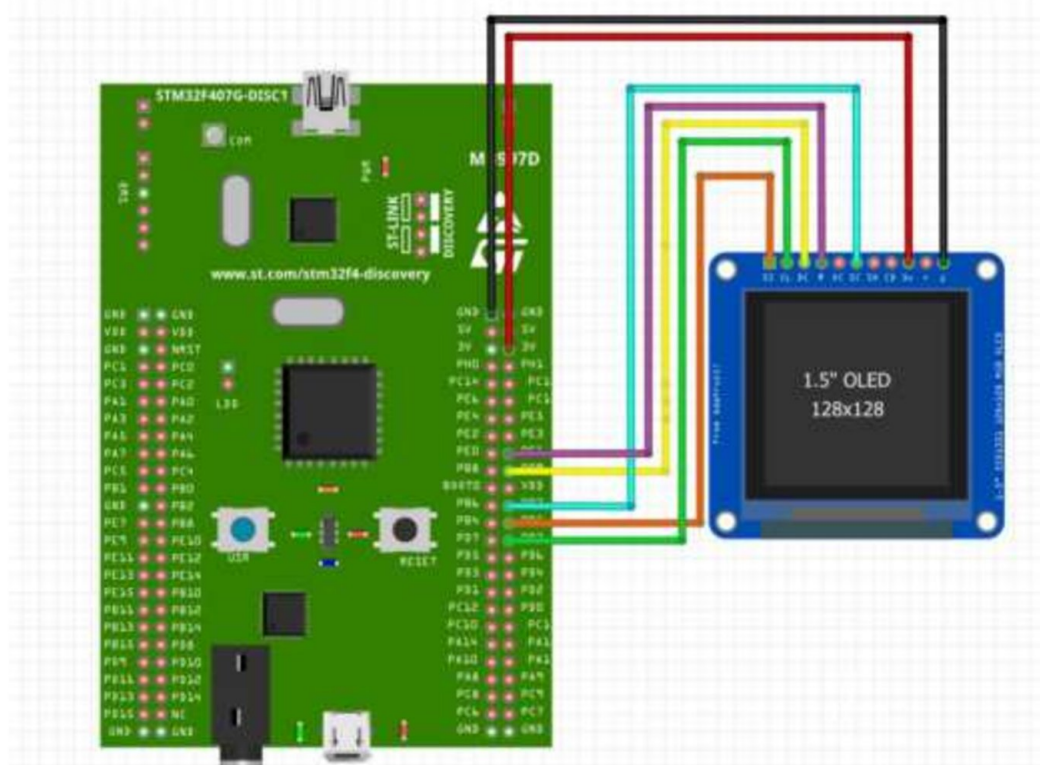


Рисунок 3.1 – Схема з'єднань мікроконтролера STM32F407VGT6-Discovery

					КС КРБ 123.158.01.00 ПЗ		
Змн.	Арк.	№ докум.	Підпис	Дата			
Розроб.		Барисай Р.М.					
Перевір.		Осухівська Г.М.					
Н. Контр.		Луцик Н.С.					
Затверд.		Осухівська Г.М.					
					<i>Генератор QR-коду на основі STM32F4</i>		
					Лім.	Арк.	Акрушів
						37	12
					ТНТУ, каф. КС, гр. СІс-44		

Порт PE1 з'єднується з портом дисплея RES. Де RES - це низьке скидання активу (Reset, low active).

Порт PB9 - з'єднуються з портом дисплея DC. Де DC використовується для 3-провідникового SPI: підключається до GND, залишається низьким.

Порт PB7- з'єднуються з портом дисплея CS. Де CS - використовується для вибору чіпів, рівень активності залишається низьким.

Порт PB5- з'єднуються з портом дисплея DIN. Де DIN - це MOSI проста лінійна передача даних від провідних пристрої до введених.

Порт PB3- з'єднуються з портом дисплея CLK. Де CLK - це SCL тактовий імпульс, що генерує за допомогою провідного пристрою синхронізацію передачі даних.

Vcc було під'єднано до +3,3В, порт GND - з'єднуються з портом дисплея GND. BS встановлюється до 1 підключення до VCC, а D/C слід підключити до GND. DIN - підключається до MOSI і CLK - підключається до to SCK.

3.1.2 Моделювання проекту в середовищі Proteus.

Proteus - це програма для схематичної зборки, що використовується для модуляції та виготовлення друкованих плат. Його можна застосовувати у багатьох випадках, залежно від розмірів конструкцій, що випускаються, та вимог до моделювання мікроконтролера. Продукти PCB Design включають авторутер та головні можливості моделювання SPICE в змішаному режимі.

Схематичне моделювання в Proteus застосовують як для моделінгу конструкцій, так і як етап проектування проекту друкованої плати. Тому він є головним компонентом і входить до складу всіх конфігурацій продукту [15].

Моделювання мікроконтролера в Proteus працює, застосовуючи або шістнадцятковий файл, або файл налагодження до частини мікроконтролера на схемі. Потім він моделюється разом із будь-якою підключеною до нього аналоговою та цифровою електронікою. Це дає користуватись ПЗ в повному спектрі прототипування проектів у таких областях як управління двигуном, контроль температури та дизайн інтерфейсу користувача. Він також знаходить

					<i>КС КРБ 123.158.00.00 ПЗ</i>	Арк
Зм	Арк	№ докум	Підпис	Дата		38

широке застосування, оскільки не потрібне жодне обладнання, його зручно використовувати як інструмент для вивчення чи навчання [16].

Доступна підтримка для співмоделювання з такими мікроконтролерами:

- Technologies Microchip dsPIC33, PIC12, PIC16, PIC18, PIC10, PIC24.
- Мікроконтролери Arduino та Atmel ARM Cortex-M3, 8051 та AVR.
- Мікроконтролери Texas Instruments MSP430, ARM Cortex-M3 та

- Freescale HC11, Parallax Basic Stamp.

Модулі компонують друковані плати їм автоматично надається інформація про підключення спискових з'єднань із модулів схематичного захоплення. Ця інформація застосовується разом із правилами проектування, визначеними користувачами, та різні інструменти для автоматизації проектування, щоб підсобити у дизайні плат без помилок. Друковані плати, що містять до 16 шарів міді, можуть бути виготовлені з конструктивними розмірами, обмеженими конфігурацією виробу [16].

PROTEUS сумісний з такими середовищами розробки вбудованого ПЗ, в тому числі [15]:

- Keil (8051 та ARM).
- ICC (МК AVR, msp430, ARM7).
- HiTECH (МК 8051 і PIC).
- IAR (будь-МК).
- CodeVisionAVR (тільки МК AVR).
- WinAVR (МК AVR).

Блок пристрою MCP2200T-I/SS використовується для перетворення протоколів USB в UART з додатковими лініями введення / виведення. Кінцевий користувач має можливість підключення до ПК для збору, передачі даних. Просто підключений інтерфейс UART мікросхем MCP2200 до інтерфейсу UART любого мікроконтролера – також може бути доданий інтерфейс USB [4].

					<i>КС КРБ 123.158.00.00 ПЗ</i>	Арк
Зм	Арк	№ докум	Підпис	Дата		39

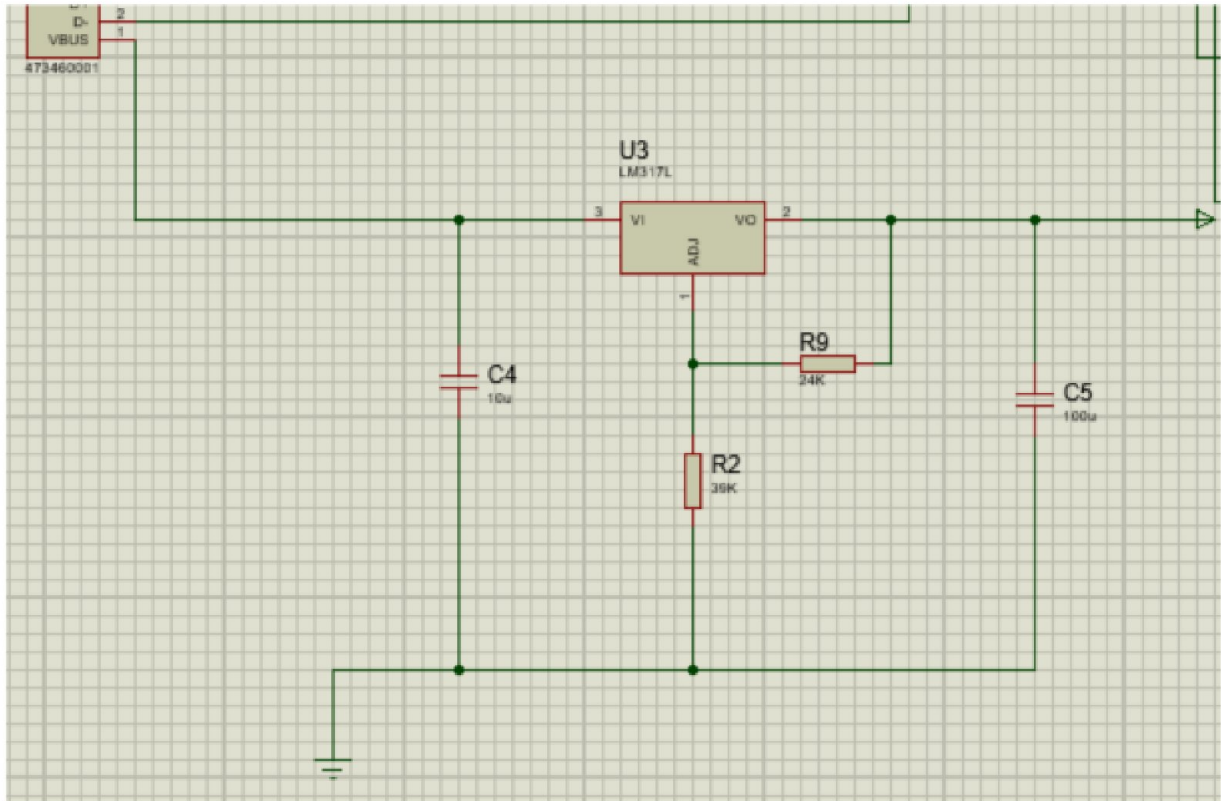


Рисунок 3.3 – Схематичне позначення LM317.

Порт VI призначений для отримання сигналу напруги від USB mini +5В.
 Порт VO призначений для виводу сигналу напруги від USB mini +3,3В.
 Порти AOJ призначений врегулювання виводу сигналу напруги від USB mini.

Блок пристрою USB mini 473460001 використовується для подання живлення від PC до CPU. Напруга через порти D+ та D- подається до мікросхеми MCP2200. Схематика блока показана на рисунку 3.4.

					КС КРБ 123.158.00.00 ПЗ	Арк
Зм	Арк	№ докум	Підпис	Дата		41

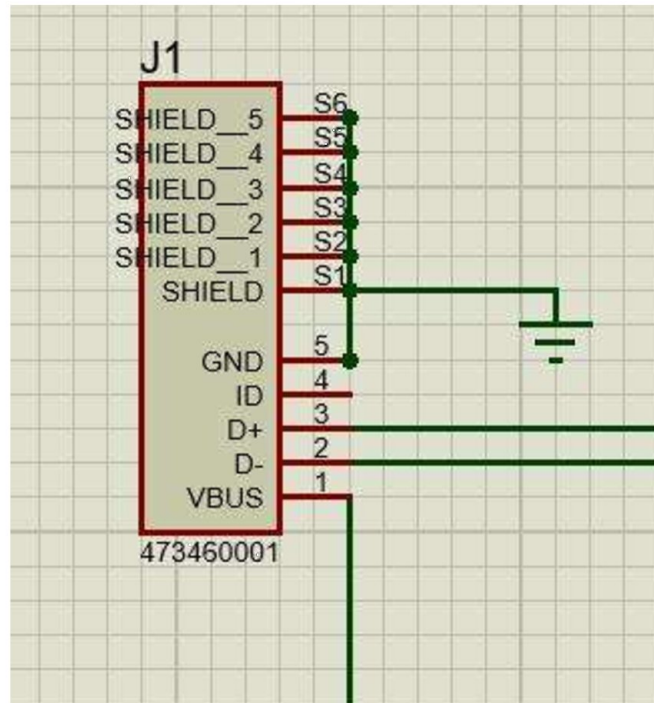


Рисунок 3.4 – Схематичне позначення USB mini

Порт GND призначений для заземлення, а VBUS - використовується для перетворення напруги.

В основному до мікроконтролера під'єднані такі компоненти як подання даних та живлення на дисплей, також CPU використовується для отримання живлення від PC по UART протоколу. Непотрібні порти подаються на заземлення.

					КС КРБ 123.158.00.00 ПЗ	Арк
						42
Зм	Арк	№ докум	Підпис	Дата		

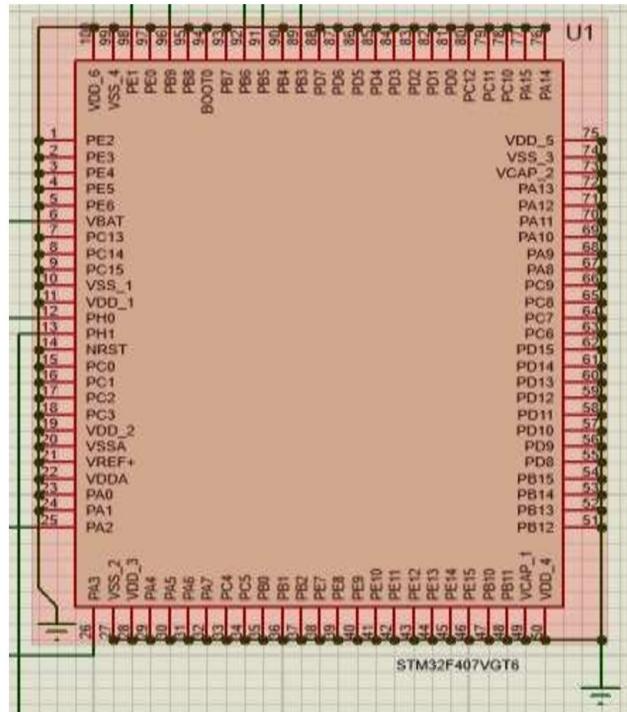


Рисунок 3.5 – Схематичне позначення STM32F407VGT6-Discovery

Для моделювання схеми в якості альтернативи SSD1357 був використаний SSD1336. Який має аналогічні функції та порти версії 1357.

Схематичне позначення блоку CPU показано на рисунку 3.5, а SSD1336 – на рисунку 3.6.

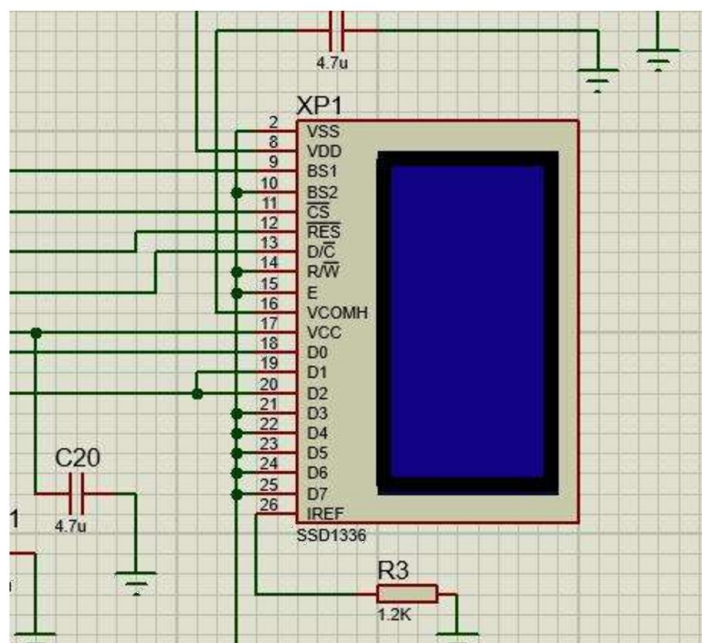


Рисунок 3.6 – Схематичне позначення SSD1336.

3.1.3 Результати моделювання в середовищі Proteus

Робочу модель пристрою генератора QR коду на STM32F407VGT6-Discovery показано на рисунку 3.7.

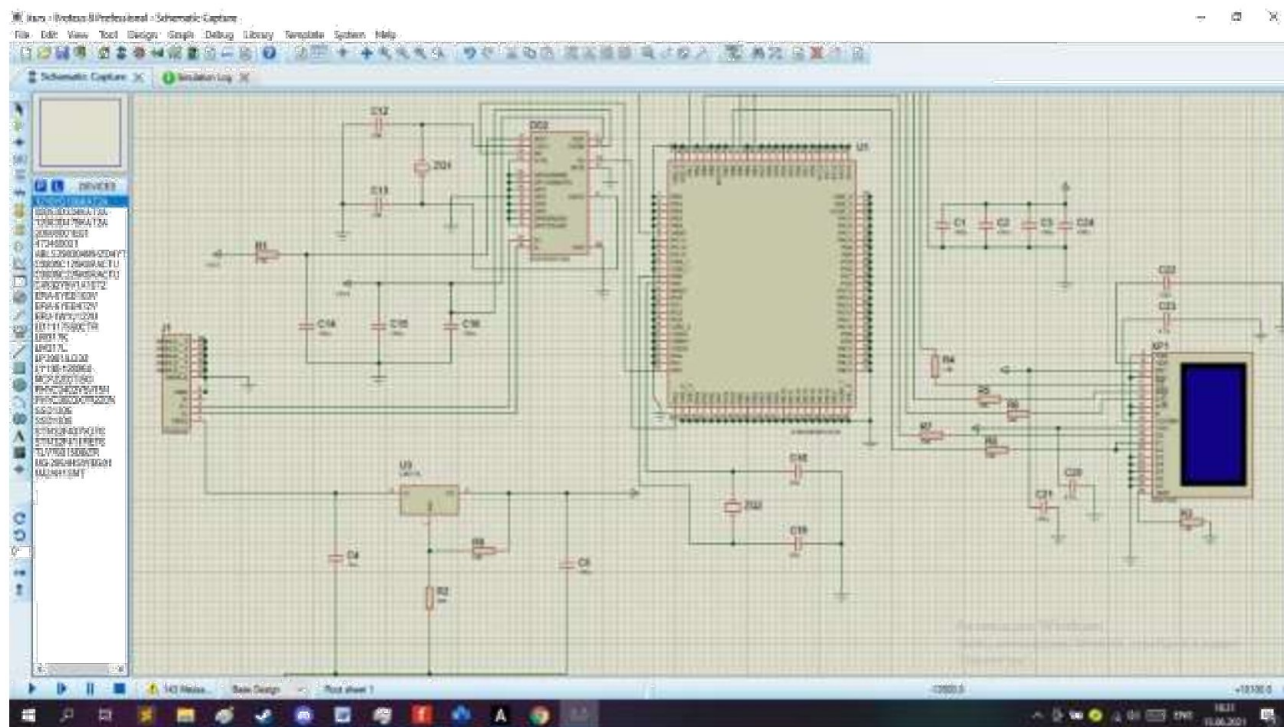


Рисунок 3.7 – Схема моделі пристрою генератора QR-коду на STM32F407VGT6-Discovery

На моделі (рисунок 3.7) з лівого боку розміщено USB mini, який подає живлення на мікросистему MCP. Мікросистема має виходи на два конденсатори та кварцовий резонатор, де конденсатори використовуються для обмеження сили змінного струму в електричному ланцюзі.

В центральній частині схеми зображений мікроконтроллер, до якого подається перетворене живлення та є можливість вводу з клавіатури за допомогою UART налаштування. Також подає дані до дисплею портами PE1, PB3, PB5, PB7, PB9.

В правому куті можна бачити екран 1336, в якого контакти RST, CS, DC, CLK, та DIN застосовуються для взаємодії з CPU і проходять через резистори

номіналом 10 кОм для обмеження струму. Інші контакти закорочені, оскільки вони не використовуються RST також підключаємо до живлення +3,3В через резистор для обмеження струму.

Результат моделювання зображено на рисунку 3.8.

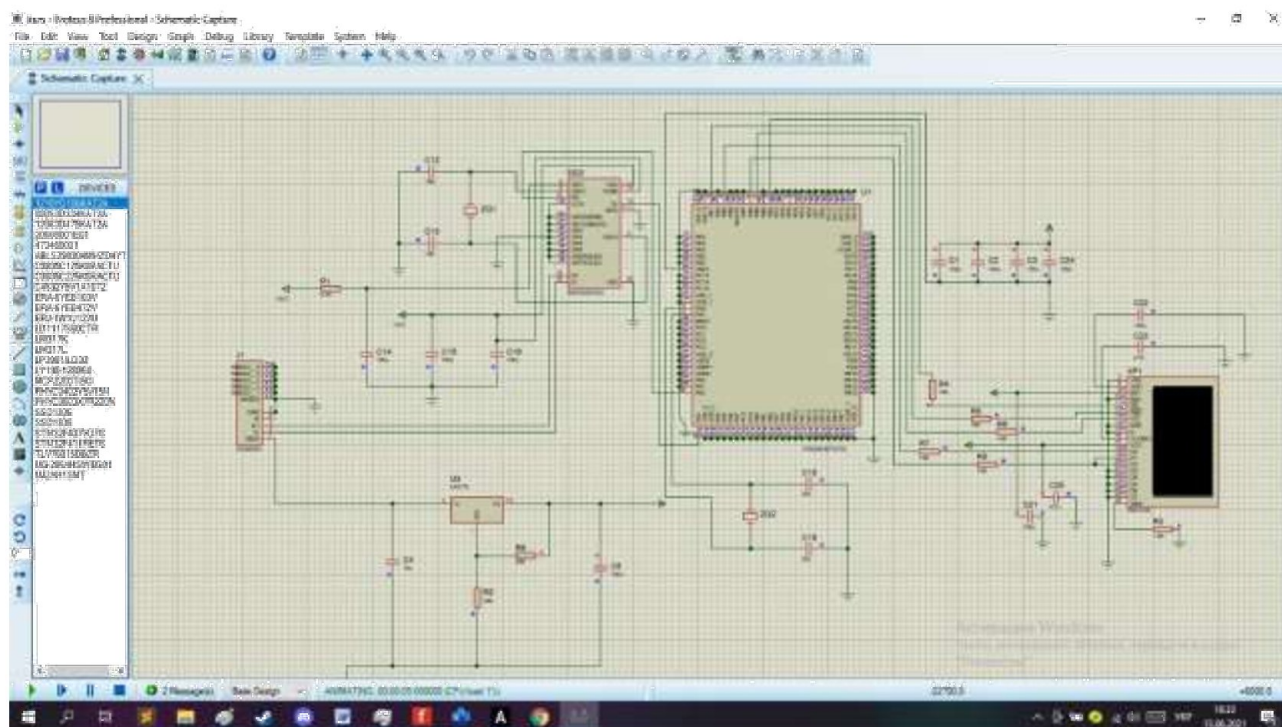


Рисунок 3.8 – Схема моделі пристрою генератора QR коду на STM32F407VGT6-Discovery

На схемі можна побачити, що всі резистори, конденсатори та кварцеві резонатори при модуляції працюють вірно.

В результаті видно, що частини мікроконтролера STM32F407VGT6-Discovery, USB mini, мікросхеми MCP та дисплея підсвічуються синім кольором, що засвідчує що симуляція виконана безпомилково. Також можна побачити, що дисплей ввімкнувся, тобто можна зафіксувати, що живлення +3,3В надійшло до дисплею.

3.1.4 Налаштування виводів мікроконтролера в середовищі STM32CubeMX

					<i>КС КРБ 123.158.00.00 ПЗ</i>	Арк
Зм	Арк	№ докум	Підпис	Дата		45

Виводи мікроконтролера STM32F407VGT6-Discovery, позначення з'єднання між SSD1351 дисплеєм зображено на таблиці 3.1 (зліва зображені виводи CPU, а справа - виводи OLED display).

Таблиця 3.1 - З'єднання між SSD1351 та STM32F407VGT6

+3,3В	Підключення живлення
GND	GND
PE1	RESET
PB9	DC (режим даних/команд)
PB7	CS (дозвіл роботи з CPU)
PB3	SCK (тактова частота)
PB5	MOSI (відправка даних)
PA2	Відправка пакету
PA3	Отримання пакету

3.2 Тестування

Тестування продукту (англ. Software Testing) – можна сказати що це технічний огляд для вияву всіляких інформацій про якість продукту порівняно його контексту, в якому він повинен використовуватись. В тестуванні є два процеси, що називають: дослідження дефектів та пошук різних помилок. Також можна сказати, що тестування є випробуванням для всіляких складових програмних забезпечень з метою їх оцінки [17].

До тестування входять такі частини контролю якості як [17]:

					КС КРБ 123.158.00.00 ПЗ	Арк
Зм	Арк	№ докум	Підпис	Дата		46

- Аналіз отриманого результату (його називають, Test Analysis).
- Проектування тесту (його трактують як, Test Design).
- Проектування робіт (його назва, Test Management).
- Виконання тесту (його назва, Test Execution).

3.2.1 Тестування методом чорного ящика

Тестування компонентів пристрою було проведено в лабораторії кафедри КС ТНТУ, де знаходилось спеціальне обладнання для тестів, таке як: індикатор напруги, мультиметр, тестер ізоляції та мікрометр.

Методом чорного ящика, перевірів функціональну придатність пристрою.

Тестові випадки наведені в таблицях що знаходяться в додатку В.

3.2.2 Тестування ПЗ на працездатність

При тестуванні програмного забезпечення використано плату STM32F407VGT6-DISCOVERY та OLED дисплей SSD1351. Для цього виконано наступне:

1. Збір макета згідно схеми з'єднань зображеної на рисунку 3.1.
2. Підключення плати за допомогою кабелю miniUSB до PC.
3. Створення та запуск проекту в Keil uVisio.
4. Загрузка код на плату STM32F407VGT6-DISCOVERY.
5. Налаштування для швидкості передачі в 115200.
6. Ввід тексту з клавіатури та натиснення на клавіші, що відповідає за генерацію.

Після того як ми занесли необхідні данні до стрічки можна побачити згенерований QR-код що показаний на рисунку 3.12.

					КС КРБ 123.158.00.00 ПЗ	Арк
						47
Зм	Арк	№ докум	Підпис	Дата		



Рисунок 3.12 — Згенерований QR-код

Таким чином ми отримуємо згенерований QR-код який після сканування робить посиланн на сторінку кафедри комп'ютерної інженерії “ <http://kaf-ks.tntu.edu.ua/> “.

Отже, в кваліфікаційній роботі бакалавра було змодельовано генератор QR-коду в середовищі Proteus 8.8. Та побудовано генератор QR-коду на основі STM32F4VGT6-Discovery.

					КС КРБ 123.158.00.00 ПЗ	Арк
Зм	Арк	№ докум	Підпис	Дата		48

РОЗДІЛ 4 БЕЗПЕКА ЖИТТЄДІЯЛЬНОСТІ, ОСНОВИ ОХОРОНИ ПРАЦІ

4.1. Охорона праці

У кваліфікаційній роботі бакалавра здійснено огляд існуючих систем та створено проект з використанням мікроконтролера STM32F407-Discovery.

Оскільки, розробка комп'ютерної систем проводиться на ПК, то важливим аспектом роботи користувача є його безпека. У зв'язку з цим, необхідно врахувати і проаналізувати вимоги і норми охорони праці, а також правила техніки безпеки при експлуатації електронно-обчислювальних засобів і периферійних пристроїв.

Основним нормативним документом, який регламентує норми і правила експлуатації електронно-обчислювальної техніки є НПАОП 0.00-7.15-18 «Загальні вимоги безпеки з охорони праці для користувачів ПК». Правила встановлюють вимоги безпеки до обладнання робочих місць операторів ЕОМ. Правила НПАОП 0.00-7.15-18 є обов'язковим для виконання роботодавцями, операторами комп'ютерного набору, операторами електронно-обчислювальних машин та працівників інших професій, які у своїй роботі застосовують ЕОМ з ВДТ і ПП.

Для забезпечення електробезпеки користувачів ПК при розробці проекту проекту на мікроконтролері STM32F407-Discovery необхідно, щоб комп'ютери і периферійні пристрої відповідали I-му класу захисту, або були заземленими відповідно до вимог НПАОП 40.1-1.32-01. Як зазначається у НПАОП 0.00-7.15-18 електронно-обчислювальні засоби повинні відповідати вимогам чинних в Україні стандартів і пройти державну санітарно-епідеміологічну експертизу у Порядку проведення державної санітарно-епідеміологічної експертизи.

					КС КРБ 123.158.00.00 ПЗ			
<i>Змн.</i>	<i>Арк.</i>	<i>№ докум.</i>	<i>Підпис</i>	<i>Дата</i>				
<i>Розроб.</i>		<i>Баршай Р.М.</i>			<i>Безпека життєдіяльності, основи охорони праці</i>	<i>Літ.</i>	<i>Арк.</i>	<i>Акрушів</i>
<i>Перевір.</i>		<i>Пилипець М.І.</i>					49	6
<i>Реценз.</i>						<i>ТНТУ, каф. КС, гр. СІс-44</i>		
<i>Н. Контр.</i>		<i>Луцик Н.С.</i>						
<i>Затверд.</i>		<i>Осухівська Г.М.</i>						

Для організації робочих місць користувачів розробки проекту в приміщенні, необхідно забезпечити дотримання вимог щодо їх розташування, а саме відстань робочого місця від стіни повинна складати 1м, відстань між робочими місцями повинна становити 1,7 м. Площа, яка виділяється на одне робоче місце, обладнане ПК має становити не менше 6.0 м. а об'єм – $\geq 20 \text{ м}^3$. При недостатньому освітленні знижується продуктивність праці користувачів ПК, і можливі негативні впливи на здоров'я такі, як короткозорість, швидка втомлюваність, тому всі приміщення, які облаштовані робочими місцями з ПК, мають природне і штучне освітлення [19].

При виборі кімнат для розміщення робочих місць ПК враховано ступінь заломлення світла на екранах дисплеїв, яке проходить через вікна і яке може викликати значне осліплення в тих, хто сидить перед ними, особливо влітку та в сонячні дні. Самі робочі місця з ПК при виконанні творчої роботи, яка потребує значної розумової концентрації, зокрема при розробці проекту на мікроконтролері STM32F407-Discovery, ізольовані одне від одного перегородкою висотою 1,6 м. Поверхня підлоги у приміщеннях повинна бути оздоблена керамічною плиткою і бути рівною та зручною для очищення. У приміщеннях штучне освітлення повинно бути виконано у вигляді комбінованої системи освітлення з використанням люмінесцентних джерел світла у світильниках загального освітлення, які розташовувати над робочими поверхнями у рівномірному-прямокутному порядку.

Штучне освітлення забезпечує на робочих місцях з ПК освітленість 300 – 500 Лк. Щоб недопустити засвітлення екранів ПК прямими світловими потоками лінії, світильники розташовані з достатнім бічним зміщенням відносно рядів робочих місць, а також паралельно до світлових отворів. Кожне вікно повинно мати світлорозсіюючі штори з коефіцієнтом відбивання 0,7 м. У робочому приміщенні необхідно забезпечити і природне освітлення, при цьому на кожному вікні закріплені жалюзі з вертикальними ламелями, що можуть регулюватися для зменшення прямого попадання сонячного світла на екран

					<i>КС КРБ 123.158.00.00 ПЗ</i>	Арк
Зм	Арк	№ докум	Підпис	Дата		50

комп'ютерів. Для запобігання нещасних випадків та організації охорони праці на виробництві розробляються інструкції з охорони праці і техніки безпеки при використанні комп'ютерної техніки. Дія інструкції поширюється на всі структурні підрозділи установи.

Для виконання роботи на ПК допускаються особи, які пройшли спеціальне навчання, медичне обстеження, вступний інструктаж з охорони праці, інструктаж на робочому місці та інструктаж з пожежної безпеки. З економічної точки зору, при розташуванні елементів робочого місця враховано наступні фактори: - можливість робити записи, розміщення документації і матеріалів, які використовує користувач; - можливість огляду елементів робочого місця; - простір для розміщення користувача; - можливість огляду простору за межами робочого місця. При дослідженні та симуляції проекту на основі середовища proteus проаналізовано та враховано необхідні вимоги охорони праці і правила техніки безпеки, що дозволило забезпечити зручні умови для ефективної роботи користувачів комп'ютерів.

4.2. Безпека в надзвичайних ситуаціях

Правовою основою організації оповіщення населення області при загрози чи виникненні надзвичайних ситуацій (НС) є Конституція України, Кодекс Цивільного захисту України, Постанови Кабінету Міністрів "Про затвердження Положення про організацію оповіщення про загрозу виникнення або виникнення надзвичайних ситуацій та зв'язку у сфері цивільного захисту", "Положення про єдину державну систему цивільного захисту", накази центрального органу виконавчої влади з питань НС, відповідні розпорядження обласної державної адміністрації та інші акти [20].

Одним із основних завдань Цивільного захисту України, як державної системи органів управління, сил і засобів, які створені для організації і забезпечення захисту населення від наслідків надзвичайних ситуацій

					КС КРБ 123.158.00.00 ПЗ	Арк
Зм	Арк	№ докум	Підпис	Дата		51

техногенного, екологічного, природного та воєнного характеру, є оповіщення населення про загрозу і виникнення надзвичайних ситуацій у мирний і воєнний часи та постійне інформування його про наявну обстановку.

Система централізованого оповіщення області представляє собою комплекс організаційно-технічних заходів, апаратури і технічних засобів оповіщення, засобів та каналів зв'язку, мереж проводового, радіо, телевізійного мовлення призначених для своєчасного доведення сигналів та інформації з питань цивільної оборони (цивільного захисту) до центральних і місцевих органів виконавчої влади, підприємств, установ, організацій і населення. Для зосередження уваги громадян перед передачею інформації вмикаються сирени, інші сигнальні засоби. Їх звук означає попереджувальний сигнал "УВАГА ВСІМ".

Взагалі система оповіщення складається із загальнодержавної, регіональних і спеціальних систем централізованого оповіщення; локальних та об'єктових систем б8 оповіщення, систем циркулярного виклику. Ці системи забезпечують оповіщення і подальше інформування:

- чергових служб міністерств та інших центральних органів виконавчої влади по службових телефонах;
- чергових служб місцевих органів виконавчої влади;
- чергових аварійно-рятувальних служб.

Для виконання основних завдань оповіщення, які визначені керівними документами, а саме: забезпечення своєчасного проходження інформації між органами управління щодо ступенів готовності; оповіщення керівного складу, населення про загрозу радіоактивного, хімічного і бактеріологічного ураження, про загрозу і виникнення надзвичайних ситуацій у мирний і особливий період та постійне інформування його про наявну обстановку.

Система оповіщення працює за принципом відбору каналів з єдиної національної системи зв'язку. Апаратура оповіщення розташована на відповідних об'єктах органів управління, електрозв'язку, чергових відділах

					<i>КС КРБ 123.158.00.00 ПЗ</i>	Арк
<i>Зм</i>	<i>Арк</i>	<i>№ докум</i>	<i>Підпис</i>	<i>Дата</i>		52

МВС, на радіо-телепередавальних центрах та інших визначених підприємствах і установах [18].

Для оперативного доведення відповідної інформації до керівного складу по телефонам застосовуються стійки циркулярного виклику та апаратура автоматизованого багатоканального оповіщення.

Для передачі попереджувального сигналу "УВАГА ВСІМ" застосовуються електричні сирени централізованого і автономного включення, наявна кількість яких в основному забезпечує озвучення території де проживає населення області.

Інформація до населення доводиться через радіотрансляційні вузли, радіотеле-передавальні центри по проводовому мовленню до якого підключено радіоточки і вуличні гучномовці, по визначеним радіо та телевізійним каналам.

На випадок виникнення надзвичайної ситуації безпосередньо на потенційно небезпечних підприємствах за їх рахунок створюються об'єктові системи оповіщення.

Локальні системи оповіщення створюються на потенційно небезпечних об'єктах, зона ураження від яких, у разі виникнення на них надзвичайної ситуації, досягає заселених територій або інших підприємств, установ, організацій. До їх складу входять абонентські радіоточки мережі радіомовлення та відомчих радіотрансляційних вузлів, вуличні гучномовці, пристрої запуску електросирен та самі електросирени, система централізованого виклику, магнітофони, магнітні стрічки із записаними текстами звернень.

Готовність систем оповіщення забезпечено шляхом:

- організованої цілодобової чергової відповідних служб;
- налагодження телефонного зв'язку чергових служб потенційно небезпечних підприємств,
- зона ураження яких може поширюватися на заселені території або території інших підприємств, установ, організацій з оперативно-черговою

					<i>КС КРБ 123.158.00.00 ПЗ</i>	Арк
						53
Зм	Арк	№ докум	Підпис	Дата		

службою пункту управління облдержадміністрації, чергових служб органів МВС в містах та районах області;

- завчасної підготовки персоналу чергових служб до дій у надзвичайних ситуаціях;

- впровадження автоматизованих систем оповіщення з використанням сучасних технологій;

- якісного експлуатаційно-технічного обслуговування апаратури і технічних засобів оповіщення та системи зв'язку.

Забороняється відключати радіотрансляційні точки та абонентські лінії, через які здійснюється запуск електросирен від мереж радіомовлення, демонтувати вуличні гучномовці без погодження з відповідними органами управління з питань ЦЗН.

					<i>КС КРБ 123.158.00.00 ПЗ</i>	Арк
<i>Зм</i>	<i>Арк</i>	<i>№ докум</i>	<i>Підпис</i>	<i>Дата</i>		54

ВИСНОВКИ

В результаті виконання кваліфікаційної роботи був розроблений генератор QR-коду на базі STM32F407VGT6-Discovery.

В процесі виконання поставленого завдання було:

– описано вимоги до генератора QR-коду на базі STM32F407VGT6-Discovery;

– розглянуто існуючий аналог;

– розроблено структурну схему, блок-схему алгоритму та схему з'єднань генератора QR-коду на базі STM32F407VGT6-Discovery;

– розроблено та описано схему електричну принципову генератора QR-коду на базі STM32F407VGT6-Discovery; \

- описано вимоги до апаратного та програмного забезпечення;

– виконано моделювання генератора QR-коду на базі STM32F407VGT6-Discovery в середовищі Proteus;

– виконано тестування;

– описано вимоги безпеки життєдіяльності та охорони праці.

Розроблена система дозволяє генерувати QR код. Програмне забезпечення дозволяє зчитувати та візуалізувати дані, тобто приймає дані з PC опрацьовує на CPU та відправляє на OLED дисплей.

Під час розробки генератора QR-коду на базі STM32F407VGT6-Discovery були задіяні навички роботи в таких середовищах як Proteus, STM32CubeMX, keil, Fritzing та Diagrams.

					КС КРБ 123.158.00.00 ПЗ	Арк
Зм	Арк	№ докум	Підпис	Дата		55

БІБЛІОГРАФІЯ

1. QR коди, як вони використовуються. URL: <http://teach-hub.com/scho-take-qr-kod-ta-yak-joho-vykorystovuvaty-vchytelyu/>
2. Алгоритм генерації QR-коду URL: <https://habr.com/ru/post/172525/> (дата звернення 20.04.2021).
3. POS-terminal. URL: https://morefunet.en.alibaba.com/product/62551663688805899120/MF66S_2_Screens_QR_code_POS_Terminal_with_NFC_and_Calculator.html. (дата звернення: 22.04.2021).
4. UART. Асинхронний приймач. URL: <https://uk.wikipedia.org/wiki/UART>
5. STM32F407VGT6. URL: <https://liven.com.ua/uk/stm32f103rct6> (дата звернення: 18.05.2021).
6. STM32F407VGT6 Datasheet. URL: <https://www.st.com/resource/en/datasheet/dm00037051.pdf> (дата звернення: 21.05.2021).
7. OLED дисплей SSD1351. URL: https://www.newhavendisplay.com/app_notes/SSD1351.pdf (дата звернення: 24.05.2021).
8. SSD1351 опис. URL: <https://www.adafruit.com/product/1431>
9. Схема електрична принципова. URL: <https://wiki.tntu.edu.ua/>
10. Шина SPI. https://uk.wikipedia.org/wiki/Serial_Peripheral_Interface
11. STM32CubeMX manual. URL: https://www.st.com/resource/en/user_manual/dm00104712-stm32cubemx-for-stm32-configuration-and-initialization-c-code-generation-stmicroelectronics.pdf (дата звернення: 24.05.2021).
12. STM32CubeMX description. URL: <https://bravikov.wordpress.com/tag/stm32cubemx/> (дата звернення: 12.05.2021).
13. KeilVision5. URL: <http://b.sumy.net.ua/swich/TF-320/uv3.pdf> (дата звернення: 12.05.2021).
14. Алгоритм. URL: <https://uk.wikipedia.org/wiki/>
15. Світлодіодний дисплей. URL: <https://uk.wikipedia.org/wiki/>

					КС КРБ 123.158.00.00 ПЗ	Арк
Зм	Арк	№ докум	Підпис	Дата		56

16. Proteus 8.8. URL: <http://www.gaw.ru/html.cgi/txt/soft/avr/Proteus.htm> (дата звернення: 30.05.2021).

17. Тестування продукту. URL: <https://quizlet.com/306571379/qa-flash-cards/>

18. Безпека життєдіяльності (забезпечення соціальної, техногенної та природної безпеки: Навч. посібник/ В.В. Бегун, І.М. Науменко. К.: Видавництво УАННП “Фенікс”, 2004. 328с.

19. Жидецький В. Ц. Основи охорони праці [Текст] : підруч. / В. Ц. Жидецький; М-во освіти і науки України. Наук.-метод. центр вищої освіти. Укр. акад. друкарства ; рец.: Г. Г. Гогіташвілі, І. І. Даценко, Б. С. Штангрет. 3-тє вид., перероб. і доп. — Львів : Укр. акад. друкарства, 2006. 336 с. Бібліогр.: с. 329–330. ISBN 966-8013-11-5.

20. ГОСТ 22.0.05-97 Безпека в надзвичайних ситуаціях. — URL: <https://docs.cntd.ru/document/1200001536> (дата звернення: 03.06.2021).

21. Охорона праці. / За ред. В.П.Кучерявого. – Львів: Оріяна – Нова, 2007. 368 с.

					КС КРБ 123.158.00.00 ПЗ	Арк
Зм	Арк	№ докум	Підпис	Дата		57

Додаток А
Технічне завдання

МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ

Тернопільський національний технічний університет імені Івана Пулюя Факультет
комп'ютерно-інформаційних систем і програмної інженерії

Кафедра комп'ютерних систем та мереж

“Затверджую”

Завідувач кафедри КС

_____ Осухівська Г.М.

“ ____ ” _____ 2021 р

ГЕНЕРАТОР QR-КОДУ НА ОСНОВІ STM32F4

ТЕХНІЧНЕ ЗАВДАННЯ

на 6 листках

Вид робіт:

Кваліфікаційна робота

На здобуття освітнього ступеня «Бакалавр»

Спеціальність 123 «Комп'ютерна інженерія»

«УЗГОДЖЕНО»

«ВИКОНАВЕЦЬ»

Керівник кваліфікаційної роботи

Студент групи СІс-44

_____ Завідувач кафедри КС
Осухівська Г.М.

_____ Баршай Р.М.

« ____ » _____ 2021 р.

« ____ » _____ 2021 р.

Тернопіль 2021

1. Назва та підстава для виконання роботи.

1.1. Генератор QR-коду на основі STM32F4.

1.2. Підставою для виконання кваліфікаційної роботи бакалавра (КРБ) є

Наказ по Університету (№ 4/7-59 від 01.02.2021 р.).

2. Виконавець.

2.1. Студента групи СІс-44 кафедри КС

Тернопільського національного технічного університету ім. І.

Пулюя Баршай Роман Михайлович.

3. Мета роботи.

3.1. Метою роботи є розробка пристрою генерування QR-коду
STM32F407VGT6-Discovery/

4. Склад виробу.

4.1. До складу системи повинні входити:

- 1) мікроконтролер;
- 2) мікросхема;
- 3) перетворювачі напруги;
- 4) виконавчі пристрої;
- 5) клавіші керування;
- 6) USB живлення;
- 7) OLED-дисплей.

5. Технічні вимоги.

5.1. Вимоги по призначенню.

5.1.1. Комп'ютерна система повинна мати наступні параметри:

- | | |
|--------------------------------------|-----------|
| 1) Діапазон температури пристрою, °C | 0...+20 |
| 2) Швидкість передачі даних, с | 0...10 |
| 2) Діапазон подання живлення, В | +3,3...+5 |

5.1.2. Швидкість обміну даними по шині SPI, не менше кбіт/с 1000

5.1.3. Пристрій повинен живитись напругою постійного струму, В +3,3

5.2. Вимоги до умов експлуатації:

5.2.1. По умовам експлуатації виріб повинен відповідати вимогам ГОСТ 15150 для УХЛ4.1

5.2.2. Температура експлуатації від 0 до +20°C

5.2.3. Відносна вологість до 100% при $t=25^{\circ}\text{C}$

5.3. Конструктивні вимоги.

5.3.1. Конструювання корпусу приладу в КРБ не передбачено.

5.3.2. Для побудови системи мають бути використані сучасні компоненти з можливістю поверхневого монтажу друкованого вузла.

5.3.3. При побудові системи необхідно передбачити розміщення роз'ємів живлення і обміну даними.

5.3.4. Габаритні розміри при макетуванні, мм, не більше:

довжина	200 мм
ширина	300 мм
висота	120 мм

5.3.5. Маса макету, кг, не більше 2,5

5.3.6. Конструкція макету повинна забезпечувати доступ до всіх комплектуючих виробів при тестуванні.

5.4. Вимоги до надійності.

5.4.1. Система повинна відповідати вимогам ДСТУ 2862-94.

5.4.2. Напрацювання на відмову, не менше 4500 год.

5.5. Вимоги метрології.

5.5.1. Вимірювання параметрів системи при моделюванні повинно виконуватись на універсальних вимірювальних приладах.

6. Економічні показники.

6.1. Собівартість системи повинна бути не більше 7000 грн.

7. Вимоги до документації.

7.1. Конструкторська документація повинна відповідати вимогам ЄСКД, ЄСПД, ДСТУ та ГОСТ.

7.2. До складу документації повинно входити:

- 1) Пояснювальна записка
- 2) Структурна схема Е1
- 3) Електрична схема Е3
- 4) Схема з'єднань Е4
- 5) Блок схема алгоритму роботи

8. Стадії та етапи розробки КРБ

8.1 Стадії та етапи виконання КРБ наведенні в таблиці 1

Таблиця 1

№	Назва етапу	Строк виконання	
		початок	кінець
1	Технічне завдання	—	до 31.03.21
2	РОЗДІЛ 1 АНАЛІЗ ТЕХНІЧНОГО ЗАВДАННЯ	01.04. 21	09.04. 21
3	РОЗДІЛ 2 ПРОЕКТНА ЧАСТИНА	09.04. 21	01.05. 21
4	РОЗДІЛ 3 ТЕХНІЧНИЙ ПРОЕКТ	02.05. 21	15.05. 21
5	РОЗДІЛ 4 БЕЗПЕКА ЖИТТЄДІЯЛЬНОСТІ, ОСНОВИ ОХОРОНИ ПРАЦІ	16.05. 21	24.05. 21
6	Нормоконтроль	22.06. 21	14.06. 21
7	Попередній захист	14.06. 21	22.06.21
8	Захист	з 25.06. 21	—

9. В дане ТЗ можуть вноситись зміни по узгодженню сторін

ДОДАТОК Б ПЕРЕЛІК ЕЛЕМЕНТІВ

<i>Поз. Позначення</i>	<i>Найменування</i>	<i>К- сть</i>	<i>Примітка</i>
	Конденсатори		
C1	0805-X5R-15V-10 нФ±10% Murata EI	1	
C2..C6	0805-X5R-15V-100 нФ±10% Murata EI	5	
C7, C8	0805-X5R-15V-15 лФ±10% Murata EI	2	
C9..C11	0805-X5R-15V-100 нФ±10% Murata EI	3	
C12,C13	0805-X5R-15V-2,2 лФ±10% Murata EI	2	
	Мікроконтролери		
DD1	STM32F407VGT6-DISCOVERY	1	
DD2	MCP2200T-I/SS	1	
	Резистори		
R1	0805-4.7 кОм±5% Murata EI	1	
R2	0805-1,2 мОм±5% Murata EI	1	
R3..R7	0805-10 кОм±5% Murata EI	5	
R8	0805-24 кОм±5% Murata EI	1	
R9	0805-39 кОм±5% Murata EI	1	
	Дисплей		
U1	UG-2828GDEDF11	1	
	Імпульсний перетворювач		
U2	LM317	1	
	Роз'єми		
XS1	47346-0001	1	
	Кварцеві резонатори		
ZQ1	LFXTAL031826Bulk	1	
ZQ2	LFXTAL031823Bulk	1	

					КСКРБ 123.158.00.00 ПЕ			
<i>Змн.</i>	<i>Арк.</i>	<i>№ докум.</i>	<i>Підпис</i>	<i>Дата</i>				
Розроб.		Баршай Р.М.			Генератор QR-коду на основі STM32F4	Літ.	Арк.	Акрушіє
Перевір.		Осухівська Г.М.						
Реценз.								
Н. контр.		Луцук Н.С.						
Затверд.		Осухівська Г.М.						
						ТНТУ, каф. КС, гр. СІс-44		

ДОДАТОК В
Лістинг програми

main.c

```
#include "stm32f4xx.h"
#include "system_stm32f4xx.h"
#include "stm32f4xx_syscfg.h"
#include "stm32f4xx_gpio.h"
#include "stm32f4xx_conf.h"
#include "stm32f4xx_exti.h"
#include "stm32f4xx_tim.h"
#include "stm32f4xx_rcc.h"
#include "stm32f4xx_rng.h"
#include "misc.h"
#include "delay.h"
#include "List.h"
#include "ff.h"
#include <stdbool.h>
#include "tm_stm32f4_pcd8544.h"
#include "bmp.h"
#include "QRLib\qr_encode.h"

FATFS fatfs;
FIL file;

volatile s8 num_of_switch=-1;
volatile u8 error_state=0;
volatile bool action_button_state=0;
volatile u32 numberOfFiles=0;
struct List *first=0,*last=0,*pointer; //list of .txt file names
bool RadicalsOrText=false;

#define maxFileNameLength 64

char* readtxtFile();
bool isBMP(FILINFO fileInfo)
{
    int i=0;
    for (i=0;i<maxFileNameLength-3;i++)
    {
        if(fileInfo.fname[i]=='.')
        {
            if((fileInfo.fname[i+1]=='B' ||
            fileInfo.fname[i+1]=='b')
            && (fileInfo.fname[i+2]=='M' ||
            fileInfo.fname[i+2]=='m')
            && (fileInfo.fname[i+3]=='P' ||
            fileInfo.fname[i+3]=='p'))
            {
```

```

        fileInfo.fname[i+4]='\0';
        return 1;
    }
}
return 0;
}
bool isTXT(FILINFO fileInfo)
{
    int i=0;
    for (i=0;i<maxFileNameLength-3;i++)
    {
        if(fileInfo.fname[i]=='.')
        {
            if((fileInfo.fname[i+1]=='T' ||
fileInfo.fname[i+1]=='t')
                && (fileInfo.fname[i+2]=='X' ||
fileInfo.fname[i+2]=='x')
                && (fileInfo.fname[i+3]=='T' ||
fileInfo.fname[i+3]=='t'))
            {
                fileInfo.fname[i+4]='\0';
                return 1;
            }
        }
    }
    return 0;
}
bool isTXT(FILINFO fileInfo)
{
    int i=0;
    for (i=0;i<maxFileNameLength-3;i++)
    {
        if(fileInfo.fname[i]=='.')
        {
            if((fileInfo.fname[i+1]=='T' ||
fileInfo.fname[i+1]=='t')
                && (fileInfo.fname[i+2]=='X' ||
fileInfo.fname[i+2]=='x')
                && (fileInfo.fname[i+3]=='T' ||
fileInfo.fname[i+3]=='t'))
            {
                fileInfo.fname[i+4]='\0';
                return 1;
            }
        }
    }
    return 0;
}
/* Flashcards */
void displayFlashcard()
{

```

```

PCD8544_Clear();
PCD8544_Refresh();
FRESULT fr;
FILINFO fno;

struct List *temporary_txt=pointer;
TCHAR* name=(TCHAR *)calloc(maxFileNameLength+13,1);
strcpy(name, "\\Radicals\\");
strcat(name, temporary_txt->file.fname);
//strcpy(name+11,);
fr = f_stat(name, &fno);
if( fr == FR_OK )
{
    TCHAR* proper=(TCHAR *)calloc(maxFileNameLength,1);
    const char *p;

    p = strchr(temporary_txt->file.fname, '.');
    if (p)
    {
        int bytes=p-temporary_txt->file.fname;
        TCHAR* name2=temporary_txt->file.fname;
        memmove(proper,name2 ,bytes);
    }

    //int radicalNameLength=strlen(temporary_txt->file.fname);
    //memmove (properName, temporary_txt->file.fname,
radicalNameLength);
    QRGenerator (proper);
    ReadAndDisplayBMP(43, 2,name);
}
free(name);
}
void listManager() //Radicals->>true, Text->>false
{
    first=last=remove_all(first);
    first->next->file.fname;
    pointer=0;
    FRESULT fresult;
    DIR Dir;

    FILINFO fileInfo;
    numberOfFiles=0;
    //choose Radicals or Text QR Code
    if (RadicalsOrText==true)
    {

```

```

        fresult = f_opendir(&Dir, "\\Radicals");
    }
    else
    {
        fresult = f_opendir(&Dir, "\\");
    }

    if(fresult != FR_OK)
    {
        return(fresult);
    }

u8 iter=0;
bool extResult=false;
for(;;)
{
    fresult = f_readdir(&Dir, &fileInfo);

    if(fresult != FR_OK)
    {
        return(fresult);
    }
    if(!fileInfo.fname[0])
    {
        break;
    }

    if (RadicalsOrText==true)
    {
        extResult=isBMP(fileInfo);
    }
    else
    {
        extResult=isTXT(fileInfo);
    }
    if(extResult==1)
    {
        if(numberOfFiles==0)
        {
            first=last=add_last(last, fileInfo);
        }
        else
        {
            last=add_last(last, fileInfo);
        }
        numberOfFiles++;
    }

    for(iter=0; iter<maxFileNameLength; ++iter)
    {
        fileInfo.fname[iter]='0';
    }
}

```

```

if (first==0)//
{
    error_state=3;
    PCD8544_GotoXY(12, 17);
    PCD8544_Puts("Brak plikow", PCD8544_Pixel_Set,
PCD8544_FontSize_5x7);
    PCD8544_GotoXY(3, 25);
    PCD8544_Puts(".txt na karcie", PCD8544_Pixel_Set,
PCD8544_FontSize_5x7);
    PCD8544_Refresh();
    TIM_Cmd(TIM4, ENABLE);
    for(;;)

}
last->next=first;
first->previous=last;
pointer=first;
}
void display_const()
{
    //Go to x=1, y=2 position
    PCD8544_GotoXY(15, 0);
    //Print data with Pixel Set mode and Fontsize of 5x7px
    PCD8544_Puts("STMQRCode", PCD8544_Pixel_Set,
PCD8544_FontSize_5x7);
    PCD8544_GotoXY(32, 40);
    PCD8544_Puts("2017", PCD8544_Pixel_Set,
PCD8544_FontSize_5x7);
    PCD8544_Refresh();
}
/* QRCode */
void display_filename()
{
    PCD8544_ClearFilename();
    //Print data with Pixel Set mode and Fontsize of 5x7px
    PCD8544_Puts(pointer->file.fname, PCD8544_Pixel_Set,
PCD8544_FontSize_5x7);
    PCD8544_Refresh();
}
void displayQRCode(int side, uint8_t *bitdata)
{
    PCD8544_Clear();
    int i=0;
    int j=0;
    int a=0;
    int l=0;
    int n=0;
    int OUT_FILE_PIXEL_PRESCALER=1;
    if (side==21)

```

```

{
    OUT_FILE_PIXEL_PRESCALER=2;
}
for (i = 0; i < side; i++) {
    for (j = 0; j < side; j++) {
        a = j * side + i;

        if ((bitdata[a / 8] & (1 << (7 - a % 8))))
        {
            for (l = 0; l <
OUT_FILE_PIXEL_PRESCALER; l++)
            {
                for (n = 0; n <
OUT_FILE_PIXEL_PRESCALER; n++)
                {

                                                                    /*(pDestData + n * 3 +
unWidthAdjusted * l) = PIXEL_COLOR_B;

                    PCD8544_DrawPixel(OUT_FILE_PIXEL_PRESCALER*i+l,OUT_FILE_PIXEL_PRESCALER*(j)+n,PCD8544_Pixel_Set);
                }
            }
        }
    }
}

PCD8544_Refresh();
}
void ReadAndDisplayBMP(uint16_t x, uint16_t y, char* filename)
{
    bmpDrawBitmap(x, y, filename);
}
void QRGenerator(char *input)
{
    int side, i, j, a;
    uint8_t bitdata[QR_MAX_BITDATA];

    // remove newline
    if (input[strlen(input) - 1] == '\n')
    {
        input[strlen(input) - 1] = 0;
    }
}

```

```

        side = qr_encode(QR_LEVEL_L, 0, input, 0, bitdata);
        displayQRCode(side, bitdata);
    }
    void EXTI0_IRQHandler(void)
    {
        // drgania stykow
        if(EXTI_GetITStatus(EXTI_Line0) != RESET)

        {
            num_of_switch=0;
            TIM_Cmd(TIM5, ENABLE);
            EXTI_ClearITPendingBit(EXTI_Line0);
        }
    }
    void EXTI9_5_IRQHandler(void)
    {
        if(EXTI_GetITStatus(EXTI_Line5) != RESET)
        {
            num_of_switch=5;
            TIM_Cmd(TIM5, ENABLE);
            EXTI_ClearITPendingBit(EXTI_Line5);
        }
        else if(EXTI_GetITStatus(EXTI_Line7) != RESET)
        {
            num_of_switch=7;
            TIM_Cmd(TIM5, ENABLE);

            EXTI_ClearITPendingBit(EXTI_Line7);
        }
        else if(EXTI_GetITStatus(EXTI_Line8) != RESET)
        {
            num_of_switch=8;
            TIM_Cmd(TIM5, ENABLE);
            EXTI_ClearITPendingBit(EXTI_Line8);
        }
        else if(EXTI_GetITStatus(EXTI_Line9) != RESET)
        {
            num_of_switch=9;
            TIM_Cmd(TIM5, ENABLE);
            EXTI_ClearITPendingBit(EXTI_Line9);
        }
    }
    void TIM4_IRQHandler(void)
    {
        if(TIM_GetITStatus(TIM4, TIM_IT_Update) != RESET)
        {
            if (error_state==1)//
            {
                GPIO_ToggleBits(GPIOD, GPIO_Pin_12);
            }
        }
    }

```



```

        }
        if (error_state==2) //
        {
            GPIO_ToggleBits(GPIOD, GPIO_Pin_13);
        }
        if (error_state==3)//.txt
        {
            GPIO_ToggleBits(GPIOD, GPIO_Pin_14);
        }
        /*if (error_state==4)//
        {
            GPIO_ToggleBits(GPIOD, GPIO_Pin_15);
        }*/

        // wyzerowanie flagi wyzwolonego przerwania
        TIM_ClearITPendingBit(TIM4, TIM_IT_Update);
    }
}
void TIM5_IRQHandler(void)
{
    if(TIM_GetITStatus(TIM5, TIM_IT_Update) != RESET)
    {
        int i_loop=0;
        u8 rand_number=0;

        if (num_of_switch==0)//user button
        {
            //ReadAndDisplayBMP(24,0,"test.bmp");
            //generuj bitmape, zapisz do pliku

        }
        else if (num_of_switch==5)// switch 5 - previous file
        {
            action_button_state=0;
            if(RadicalsOrText==true)
            {
                rand_number=RNG_GetRandomNumber()%(numberOfFiles-1)+1;
            }
            else
            {
                rand_number=1;
            }

            for(i_loop=0;i_loop<rand_number;i_loop++)
            {
                pointer=pointer->previous;
            }
        }
    }
}

```

```

if (RadicalsOrText==true) //flashcard mode
{
    displayFlashcard();
}
else
{
    PCD8544_Clear();
    display_const();
    display_filename();
}
}
else if (num_of_switch==7)// switch 7 - generate QRCode
{
    if (RadicalsOrText==false)
    {
        if(action_button_state==0)
        {
            action_button_state=1;

            FRESULT fr;
            struct List *temporary_txt=pointer;
            char fileContent[1];
            UINT *readBytes=0;

            fr = f_open( &file, temporary_txt-
>file.fname, FA_READ );
            if( fr == FR_OK )
            {
                f_read(&file,fileContent,7089,readBytes);
                f_close(&file);
            }
            if (fileContent[0]!='\200')
            {
                QGenerator(fileContent);
            }

            else
            {
                error_state=3;
                GPIO_ResetBits(GPIOD,
GPIO_Pin_12|GPIO_Pin_13|GPIO_Pin_14|GPIO_Pin_15);
                TIM_Cmd(TIM4, ENABLE);
                PCD8544_ClearFilename();
                PCD8544_GotoXY(0, 13);
                PCD8544_Puts("file is not",
PCD8544_Pixel_Set, PCD8544_FontSize_5x7);
                PCD8544_GotoXY(5, 21);
                PCD8544_Puts("or problem z",
PCD8544_Pixel_Set, PCD8544_FontSize_5x7);
                PCD8544_GotoXY(3, 29);
            }
        }
    }
}
}

```

```

        PCD8544_Puts("I read it",
PCD8544_Pixel_Set, PCD8544_FontSize_5x7);
        PCD8544_Refresh();
    }
}
else
{
    GPIO_ResetBits(GPIOD,
GPIO_Pin_12|GPIO_Pin_13|GPIO_Pin_14|GPIO_Pin_15);
    TIM_Cmd(TIM4, DISABLE);
    TIM_SetCounter(TIM4, 0);
    action_button_state=0;
    PCD8544_Clear();
    display_const();
    display_filename();
}
}

else if (num_of_switch==8)// switch 8 - next file
{
    action_button_state=0;
    if(RadicalsOrText==true)
    {
        rand_number=RNG_GetRandomNumber()%(numberOfFiles-1)+1
    }
    else
    {
        rand_number=1;
    }

    for(i_loop=0;i_loop<rand_number;i_loop++)
    {
        pointer=pointer->next;
    }
    if (RadicalsOrText==true) //flashcard mode
    {
        displayFlashcard();
    }
    else
    {
        PCD8544_Clear();
        display_const();
        display_filename();
    }
}
}

```

```

}
else if (num_of_switch==9)// flashcards or text
{
    action_button_state=0;
    RadicalsOrText=!RadicalsOrText;

    listManager();

    if (RadicalsOrText==true) //flashcard mode
    {
rand_number=RNG_GetRandomNumber()%(numberOfFiles-1)+1;
        for(i_loop=0;i_loop<rand_number;i_loop++)
        {
            pointer=pointer->next;
        }
        displayFlashcard();
    }
    else //text mode
    {
        PCD8544_Clear();
        PCD8544_Refresh();

        display_const();
        display_filename();
    }
}
num_of_switch=-1;
TIM_Cmd(TIM5, DISABLE);
TIM_SetCounter(TIM5, 0);

        TIM_ClearITPendingBit(TIM5, TIM_IT_Update);
}
}
void ERROR_TIM_4()
{
    // TIMER DO OBSLUGI DIOD W PRZYPADKU BLEDU
    RCC_APB1PeriphClockCmd(RCC_APB1Periph_TIM4, ENABLE);
    TIM_TimeBaseInitTypeDef TIMER_4;
    /* Time base configuration */
    TIMER_4.TIM_Period = 24000-1;
    TIMER_4.TIM_Prescaler = 1000-1;
    TIMER_4.TIM_ClockDivision = TIM_CKD_DIV1;
    TIMER_4.TIM_CounterMode = TIM_CounterMode_Up;
    TIM_TimeBaseInit(TIM4, &TIMER_4);
    TIM_Cmd(TIM4,DISABLE);

    // KONFIGURACJA PRZERWAN - TIMER/COUNTER
    NVIC_InitTypeDef NVIC_InitStructure;

```

```

    NVIC_InitStructure.NVIC_IRQChannel = TIM4_IRQn;// call
number
    NVIC_InitStructure.NVIC_IRQChannelPreemptionPriority =
0x00;// priorytet
    NVIC_InitStructure.NVIC_IRQChannelSubPriority = 0x00;//
    NVIC_InitStructure.NVIC_IRQChannelCmd = ENABLE;
    NVIC_Init(&NVIC_InitStructure);// write down
    TIM_ClearITPendingBit(TIM4, TIM_IT_Update);// wyczyszczenie
przerwania od timera 4 (wystapilo przy konfiguracji timera)
    TIM_ITConfig(TIM4, TIM_IT_Update, ENABLE);// zezwolenie na
przerwania od przepelnienia dla timera 4
}
void JOINT_VIBRATION()
{
    // TIMER DO ELIMINACJI DRGAN STYKOW, TIM5
    RCC_APB1PeriphClockCmd(RCC_APB1Periph_TIM5, ENABLE);

    TIM_TimeBaseInitTypeDef TIMER;
    /* Time base configuration */
    TIMER.TIM_Period = 8400-1;
    TIMER.TIM_Prescaler = 3000-1;
    TIMER.TIM_ClockDivision = TIM_CKD_DIV1;
    TIMER.TIM_CounterMode = TIM_CounterMode_Up;

    TIM_TimeBaseInit(TIM5, &TIMER);
    TIM_Cmd(TIM5,DISABLE);

    // KONFIGURACJA PRZERWAN - TIMER/COUNTER
    NVIC_InitTypeDef NVIC_InitStructure;
    NVIC_InitStructure.NVIC_IRQChannel = TIM5_IRQn;// numer
przerwania
    NVIC_InitStructure.NVIC_IRQChannelPreemptionPriority =
0x00;// priorytet glowny
    NVIC_InitStructure.NVIC_IRQChannelSubPriority = 0x00;//
subpriorytet

    NVIC_InitStructure.NVIC_IRQChannelCmd = ENABLE;// uruchom
dany kanal
    NVIC_Init(&NVIC_InitStructure);// write the complete
structure to the register
    TIM_ClearITPendingBit(TIM5, TIM_IT_Update);// wyczyszczenie
przerwania od timera 5 (wystapilo przy konfiguracji timera)
    TIM_ITConfig(TIM5, TIM_IT_Update, ENABLE);// zezwolenie na
przerwania od przepelnienia dla timera 5
}
void DIODES_init()
{
    RCC_AHB1PeriphClockCmd(RCC_AHB1Periph_GPIOD, ENABLE);
    GPIO_InitTypeDef DIODES;
    /* Configure PD12, PD13, PD14 and PD15 in output pushpull
mode */
    DIODES.GPIO_Pin = GPIO_Pin_12 | GPIO_Pin_13| GPIO_Pin_14|
GPIO_Pin_15;

```

```

    DIODES.GPIO_Mode = GPIO_Mode_OUT;// type of introduction,
    DIODES.GPIO_OType = GPIO_OType_PP;// complementary outputs
    DIODES.GPIO_Speed = GPIO_Speed_100MHz;// max. V translation
    DIODES.GPIO_PuPd = GPIO_PuPd_NOPULL;// no participation
    GPIO_Init(GPIOA, &DIODES);
}
void BUTTON_init()
{
    RCC_AHB1PeriphClockCmd(RCC_AHB1Periph_GPIOA , ENABLE);
    GPIO_InitTypeDef USER_BUTTON;
    USER_BUTTON.GPIO_Pin = GPIO_Pin_0;
    USER_BUTTON.GPIO_Mode = GPIO_Mode_IN;
    USER_BUTTON.GPIO_PuPd = GPIO_PuPd_NOPULL;
    GPIO_Init(GPIOA, &USER_BUTTON);

    USER_BUTTON.GPIO_Pin = GPIO_Pin_5 | GPIO_Pin_9 | GPIO_Pin_7 |
GPIO_Pin_8;
    USER_BUTTON.GPIO_PuPd = GPIO_PuPd_DOWN;
    GPIO_Init(GPIOA, &USER_BUTTON);
}
void INTERRUPT_init()
{
    // KONFIGURACJA KONTROLERA PRZERWAN
    NVIC_InitTypeDef NVIC_InitStructure;
    NVIC_InitStructure.NVIC_IRQChannel = EXTI0_IRQn; // numer
przerwania
    NVIC_InitStructure.NVIC_IRQChannelPreemptionPriority =
0x00;// priorytet glowny
    NVIC_InitStructure.NVIC_IRQChannelSubPriority = 0x00;//
subpriorytet
    NVIC_InitStructure.NVIC_IRQChannelCmd = ENABLE;// uruchom
dany kanal
    NVIC_Init(&NVIC_InitStructure);// zapisz wypelniona
struktura do rejestrow

    EXTI_InitTypeDef EXTI_InitStructure;
    EXTI_InitStructure.EXTI_Line = EXTI_Line0;// a selection of
the numbers of the currently configured line
    EXTI_InitStructure.EXTI_Mode = EXTI_Mode_Interrupt;// try
selection - search for success
    EXTI_InitStructure.EXTI_Trigger = EXTI_Trigger_Rising;// the
selection of the response to which the response responds
    EXTI_InitStructure.EXTI_LineCmd = ENABLE;//
    EXTI_Init(&EXTI_InitStructure);// write down the
configuration structure of the external of the external
airfields

    SYSCFG_EXTILineConfig(EXTI_PortSourceGPIOA,
EXTI_PinSource0);

```

```

NVIC_InitStructure.NVIC_IRQChannel = EXTI9_5_IRQn;
EXTI_InitStructure.EXTI_Line = EXTI_Line5 |EXTI_Line9|
EXTI_Line7 | EXTI_Line8;
NVIC_Init(&NVIC_InitStructure);
EXTI_Init(&EXTI_InitStructure);
    SYSCFG_EXTILineConfig(EXTI_PortSourceGPIOA,
EXTI_PinSource5);
    SYSCFG_EXTILineConfig(EXTI_PortSourceGPIOA,
EXTI_PinSource9);
    SYSCFG_EXTILineConfig(EXTI_PortSourceGPIOA,
EXTI_PinSource7);
    SYSCFG_EXTILineConfig(EXTI_PortSourceGPIOA,
EXTI_PinSource8);
}

int main( void )
{
    SystemInit();
    DIODES_init();// diode initialization
    ERROR_TIM_4();
    delay_init( 80 );// sending 80 watch pulses; to initiate SPI
    SPI_SD_Init();// initialization SPI under SD

    SysTick_CLKSourceConfig(SysTick_CLKSource_HCLK_Div8);//
zegar 24-bitowy
    SysTick_Config(90000);

    //Initialize LCD with 0x38 software contrast
    PCD8544_Init(0x38);
    display_const();

    // SD CARD
    FRESULT fresult;

    disk_initialize(0);// cart initialization
    Fresult = f_mount( &fatfs, 1,1 );// recording logical disk
in the system
    if(fresult != FR_OK) //if the device is bladed STM32

```

```

insertion without a card in the module badly supported cables
{
    error_state=1;
    PCD8544_GotoXY(21, 13);
    PCD8544_Puts("Problem", PCD8544_Pixel_Set,
PCD8544_FontSize_5x7);
    PCD8544_GotoXY(17, 21);
    PCD8544_Puts("with cables", PCD8544_Pixel_Set,
PCD8544_FontSize_5x7);
    PCD8544_GotoXY(17, 29);
    PCD8544_Puts("or card", PCD8544_Pixel_Set,
PCD8544_FontSize_5x7);
    PCD8544_Refresh();
    TIM_Cmd(TIM4, ENABLE);
    for(;;)
    { }
}
listManager();
RCC_AHB2PeriphClockCmd(RCC_AHB2Periph_RNG, ENABLE);

BUTTON_init();
JOINT_VIBRATION();
INTERRUPT_init();
u32 i_loop=0;

display_filename();

for(;;)
{

}

return 0;
}
void SysTick_Handler()
{
    disk_timerproc();
}

```


tm_stm32f4_pcd8544.h

#include "tm_stm32f4_pcd8544.h"

```
unsigned char PCD8544_Buffer[PCD8544_BUFFER_SIZE];
unsigned char PCD8544_UpdateXmin = 0, PCD8544_UpdateXmax = 0,
PCD8544_UpdateYmin = 0, PCD8544_UpdateYmax = 0;
unsigned char PCD8544_x;
unsigned char PCD8544_y;
```

//Fonts 5x7

```
const uint8_t PCD8544_Font5x7 [97][PCD8544_CHAR5x7_WIDTH] = {
    { 0x00, 0x00, 0x00, 0x00, 0x00 }, // sp
    { 0x00, 0x00, 0x2f, 0x00, 0x00 }, // !
    { 0x00, 0x07, 0x00, 0x07, 0x00 }, // "
    { 0x14, 0x7f, 0x14, 0x7f, 0x14 }, // #
    { 0x24, 0x2a, 0x7f, 0x2a, 0x12 }, // $
    { 0x32, 0x34, 0x08, 0x16, 0x26 }, // %
    { 0x36, 0x49, 0x55, 0x22, 0x50 }, // &
    { 0x00, 0x05, 0x03, 0x00, 0x00 }, // '
    { 0x00, 0x1c, 0x22, 0x41, 0x00 }, // (
    { 0x00, 0x41, 0x22, 0x1c, 0x00 }, // )
    { 0x14, 0x08, 0x3E, 0x08, 0x14 }, // *
    { 0x08, 0x08, 0x3E, 0x08, 0x08 }, // +
    { 0x00, 0x00, 0x50, 0x30, 0x00 }, // ,
    { 0x10, 0x10, 0x10, 0x10, 0x10 }, // -
    { 0x00, 0x60, 0x60, 0x00, 0x00 }, // .
    { 0x20, 0x10, 0x08, 0x04, 0x02 }, // /
    { 0x3E, 0x51, 0x49, 0x45, 0x3E }, // 0
    { 0x00, 0x42, 0x7F, 0x40, 0x00 }, // 1
    { 0x42, 0x61, 0x51, 0x49, 0x46 }, // 2
    { 0x21, 0x41, 0x45, 0x4B, 0x31 }, // 3
    { 0x18, 0x14, 0x12, 0x7F, 0x10 }, // 4

    { 0x27, 0x45, 0x45, 0x45, 0x39 }, // 5
    { 0x3C, 0x4A, 0x49, 0x49, 0x30 }, // 6
    { 0x01, 0x71, 0x09, 0x05, 0x03 }, // 7
    { 0x36, 0x49, 0x49, 0x49, 0x36 }, // 8 LA SWITCH Pin_5, Pin_6,
    { 0x06, 0x49, 0x49, 0x29, 0x1E }, // 9
    { 0x00, 0x36, 0x36, 0x00, 0x00 }, // :
    { 0x00, 0x56, 0x36, 0x00, 0x00 }, // ;
    { 0x08, 0x14, 0x22, 0x41, 0x00 }, // <
    { 0x14, 0x14, 0x14, 0x14, 0x14 }, // =
    { 0x00, 0x41, 0x22, 0x14, 0x08 }, // >
    { 0x02, 0x01, 0x51, 0x09, 0x06 }, // ?
    { 0x32, 0x49, 0x59, 0x51, 0x3E }, // @
    { 0x7E, 0x11, 0x11, 0x11, 0x7E }, // A
```

```

{ 0x7F, 0x49, 0x49, 0x49, 0x36 }, // B
{ 0x3E, 0x41, 0x41, 0x41, 0x22 }, // C
{ 0x7F, 0x41, 0x41, 0x22, 0x1C }, // D
{ 0x7F, 0x49, 0x49, 0x49, 0x41 }, // E
{ 0x7F, 0x09, 0x09, 0x09, 0x01 }, // F
{ 0x3E, 0x41, 0x49, 0x49, 0x7A }, // G
{ 0x7F, 0x08, 0x08, 0x08, 0x7F }, // H
{ 0x00, 0x41, 0x7F, 0x41, 0x00 }, // I
{ 0x20, 0x40, 0x41, 0x3F, 0x01 }, // J
{ 0x7F, 0x08, 0x14, 0x22, 0x41 }, // K
{ 0x7F, 0x40, 0x40, 0x40, 0x40 }, // L
{ 0x7F, 0x02, 0x0C, 0x02, 0x7F }, // M
{ 0x7F, 0x04, 0x08, 0x10, 0x7F }, // N
{ 0x3E, 0x41, 0x41, 0x41, 0x3E }, // O
{ 0x7F, 0x09, 0x09, 0x09, 0x06 }, // P
{ 0x3E, 0x41, 0x51, 0x21, 0x5E }, // Q
{ 0x7F, 0x09, 0x19, 0x29, 0x46 }, // R
{ 0x46, 0x49, 0x49, 0x49, 0x31 }, // S
{ 0x01, 0x01, 0x7F, 0x01, 0x01 }, // T
{ 0x3F, 0x40, 0x40, 0x40, 0x3F }, // U
{ 0x1F, 0x20, 0x40, 0x20, 0x1F }, // V
{ 0x3F, 0x40, 0x38, 0x40, 0x3F }, // W

{ 0x63, 0x14, 0x08, 0x14, 0x63 }, // X
{ 0x07, 0x08, 0x70, 0x08, 0x07 }, // Y
{ 0x61, 0x51, 0x49, 0x45, 0x43 }, // Z
{ 0x00, 0x7F, 0x41, 0x41, 0x00 }, // [
{ 0x55, 0x2A, 0x55, 0x2A, 0x55 }, // 55
{ 0x00, 0x41, 0x41, 0x7F, 0x00 }, // ]
{ 0x04, 0x02, 0x01, 0x02, 0x04 }, // ^
{ 0x40, 0x40, 0x40, 0x40, 0x40 }, // _
{ 0x00, 0x01, 0x02, 0x04, 0x00 }, // `
{ 0x20, 0x54, 0x54, 0x54, 0x78 }, // a
{ 0x7F, 0x48, 0x44, 0x44, 0x38 }, // b
{ 0x38, 0x44, 0x44, 0x44, 0x20 }, // c
{ 0x38, 0x44, 0x44, 0x48, 0x7F }, // d
{ 0x38, 0x54, 0x54, 0x54, 0x18 }, // e
{ 0x08, 0x7E, 0x09, 0x01, 0x02 }, // f
{ 0x0C, 0x52, 0x52, 0x52, 0x3E }, // g
{ 0x7F, 0x08, 0x04, 0x04, 0x78 }, // h
{ 0x00, 0x44, 0x7D, 0x40, 0x00 }, // i
{ 0x20, 0x40, 0x44, 0x3D, 0x00 }, // j

{ 0x7F, 0x10, 0x28, 0x44, 0x00 }, // k
{ 0x00, 0x41, 0x7F, 0x40, 0x00 }, // l
{ 0x7C, 0x04, 0x18, 0x04, 0x78 }, // m
{ 0x7C, 0x08, 0x04, 0x04, 0x78 }, // n
{ 0x38, 0x44, 0x44, 0x44, 0x38 }, // o
{ 0x7C, 0x14, 0x14, 0x14, 0x08 }, // p
{ 0x08, 0x14, 0x14, 0x18, 0x7C }, // q

```

```

{ 0x7C, 0x08, 0x04, 0x04, 0x08 }, // r
{ 0x48, 0x54, 0x54, 0x54, 0x20 }, // s
{ 0x04, 0x3F, 0x44, 0x40, 0x20 }, // t
{ 0x3C, 0x40, 0x40, 0x20, 0x7C }, // u
{ 0x1C, 0x20, 0x40, 0x20, 0x1C }, // v
{ 0x3C, 0x40, 0x30, 0x40, 0x3C }, // w
{ 0x44, 0x28, 0x10, 0x28, 0x44 }, // x
{ 0x0C, 0x50, 0x50, 0x50, 0x3C }, // y
{ 0x44, 0x64, 0x54, 0x4C, 0x44 }, // z
{ 0x00, 0x7F, 0x3E, 0x1C, 0x08 }, // > Filled      123
{ 0x08, 0x1C, 0x3E, 0x7F, 0x00 }, // < Filled      124
{ 0x08, 0x7C, 0x7E, 0x7C, 0x08 }, // Arrow up      125
{ 0x10, 0x3E, 0x7E, 0x3E, 0x10 }, // Arrow down    126
{ 0x3E, 0x3E, 0x3E, 0x3E, 0x3E }, // Stop          127
{ 0x00, 0x7F, 0x3E, 0x1C, 0x08 } // Play          128

```

```
};
```

```

const uint8_t PCD8544_Font3x5[106][3] = {
  { 0x00, 0x00, 0x00 }, // sp - 32
  { 0x00, 0x17, 0x00 }, // ! - 33
  { 0x03, 0x00, 0x03 }, // " - 34
  { 0x1F, 0x0A, 0x1F }, // # - 35
  { 0x0A, 0x1F, 0x05 }, // $
  { 0x09, 0x04, 0x12 }, // %
  { 0x0F, 0x17, 0x1C }, // &
  { 0x00, 0x03, 0x00 }, // '

  { 0x00, 0x0E, 0x11 }, // ( - 40
  { 0x11, 0x0E, 0x00 }, // )
  { 0x05, 0x02, 0x05 }, // *
  { 0x04, 0x0E, 0x04 }, // +
  { 0x10, 0x08, 0x00 }, // ,
  { 0x04, 0x04, 0x04 }, // - - 45
  { 0x00, 0x10, 0x00 }, // .
  { 0x08, 0x04, 0x02 }, // /
  { 0x1F, 0x11, 0x1F }, // 0
  { 0x12, 0x1F, 0x10 }, // 1
  { 0x1D, 0x15, 0x17 }, // 2 - 50
  { 0x11, 0x15, 0x1F }, // 3
  { 0x07, 0x04, 0x1F }, // 4
  { 0x17, 0x15, 0x1D }, // 5
  { 0x1F, 0x15, 0x1D }, // 6
  { 0x01, 0x01, 0x1F }, // 7 - 55
  { 0x1F, 0x15, 0x1F }, // 8
  { 0x17, 0x15, 0x1F }, // 9 - 57
  { 0x00, 0x0A, 0x00 }, // :
  { 0x10, 0x0A, 0x00 }, // ;
  { 0x04, 0x0A, 0x11 }, // < - 60

```

```

{ 0x0A, 0x0A, 0x0A }, // =
{ 0x11, 0x0A, 0x04 }, // >
{ 0x01, 0x15, 0x03 }, // ?
{ 0x0E, 0x15, 0x16 }, // @
{ 0x1E, 0x05, 0x1E }, // A - 65
{ 0x1F, 0x15, 0x0A }, // B
{ 0x0E, 0x11, 0x11 }, // C
{ 0x1F, 0x11, 0x0E }, // D
{ 0x1F, 0x15, 0x15 }, // E
{ 0x1F, 0x05, 0x05 }, // F - 70
{ 0x0E, 0x15, 0x1D }, // G
{ 0x1F, 0x04, 0x1F }, // H
{ 0x11, 0x1F, 0x11 }, // I
{ 0x08, 0x10, 0x0F }, // J
{ 0x1F, 0x04, 0x1B }, // K - 75
{ 0x1F, 0x10, 0x10 }, // L
{ 0x1F, 0x06, 0x1F }, // M
{ 0x1F, 0x0E, 0x1F }, // N
{ 0x0E, 0x11, 0x0E }, // O
{ 0x1F, 0x05, 0x02 }, // P - 80
{ 0x0E, 0x11, 0x1E }, // Q
{ 0x1F, 0x0D, 0x16 }, // R
{ 0x12, 0x15, 0x09 }, // S
{ 0x01, 0x1F, 0x01 }, // T
{ 0x0F, 0x10, 0x0F }, // U - 85
{ 0x07, 0x18, 0x07 }, // V
{ 0x1F, 0x0C, 0x1F }, // W
{ 0x1B, 0x04, 0x1B }, // X
{ 0x03, 0x1C, 0x03 }, // Y
{ 0x19, 0x15, 0x13 }, // Z - 90
{ 0x1F, 0x11, 0x00 }, // [
{ 0x02, 0x04, 0x08 }, // 55 - backspace - 92
{ 0x00, 0x11, 0x1F }, // ]
{ 0x02, 0x01, 0x02 }, // ^
{ 0x10, 0x10, 0x10 }, // _ - 95

{ 0x1A, 0x16, 0x1C }, // a
{ 0x1F, 0x12, 0x0C }, // b
{ 0x0C, 0x12, 0x12 }, // c
{ 0x0C, 0x12, 0x1F }, // d - 100
{ 0x0C, 0x1A, 0x16 }, // e
{ 0x04, 0x1E, 0x05 }, // f
{ 0x06, 0x15, 0x0F }, // g
{ 0x1F, 0x02, 0x1C }, // h
{ 0x00, 0x1D, 0x00 }, // i - 105
{ 0x10, 0x10, 0x0D }, // j
{ 0x1F, 0x0C, 0x12 }, // k
{ 0x11, 0x1F, 0x10 }, // l
{ 0x1E, 0x0E, 0x1E }, // m

```

```

{ 0x1E, 0x02, 0x1C }, // n - 110
{ 0x0C, 0x12, 0x0C }, // o
{ 0x1E, 0x0A, 0x04 }, // p
{ 0x04, 0x0A, 0x1E }, // q
{ 0x1C, 0x02, 0x02 }, // r
{ 0x14, 0x1E, 0x0A }, // s - 115

{ 0x02, 0x1F, 0x12 }, // t
{ 0x0E, 0x10, 0x1E }, // u
{ 0x0E, 0x10, 0x0E }, // v
{ 0x1E, 0x1C, 0x1E }, // w
{ 0x12, 0x0C, 0x12 }, // x - 120
{ 0x02, 0x14, 0x1E }, // y
{ 0x1A, 0x1E, 0x16 }, // z
{ 0x04, 0x1B, 0x11 }, // {
{ 0x00, 0x1F, 0x00 }, // |
{ 0x11, 0x1B, 0x04 }, // }
{ 0x04, 0x06, 0x02 }, // ~
{ 0x1F, 0x1F, 0x1F }, // delete
};

void PCD8544_InitIO(void) {
    GPIO_InitTypeDef GPIO_InitStructure;
    //Enable clock for all pins
    RCC_AHB1PeriphClockCmd(RCC_AHB1Periph_GPIOC, ENABLE);

    //Common settings for all pins
    GPIO_InitStructure.GPIO_OType = GPIO_OType_PP;
    GPIO_InitStructure.GPIO_PuPd = GPIO_PuPd_NOPULL;
    GPIO_InitStructure.GPIO_Speed = GPIO_Speed_100MHz;
    GPIO_InitStructure.GPIO_Mode = GPIO_Mode_OUT;

    //RST pin
    GPIO_InitStructure.GPIO_Pin = PCD8544_RST_PIN;
    GPIO_Init(PCD8544_RST_PORT, &GPIO_InitStructure);

    //DC pin
    GPIO_InitStructure.GPIO_Pin = PCD8544_DC_PIN;
    GPIO_Init(PCD8544_DC_PORT, &GPIO_InitStructure);

    //CE pin
    GPIO_InitStructure.GPIO_Pin = PCD8544_CE_PIN;
    GPIO_Init(PCD8544_CE_PORT, &GPIO_InitStructure);

    //Reset pin HIGH
    GPIO_SetBits(PCD8544_RST_PORT, PCD8544_RST_PIN);

    //CE HIGH
    PCD8544_CE_HIGH;
}

```

```

//Initialize SPI
SPI_InitTypeDef SPI_InitStruct;

/* Enable SPI clock */
RCC->APB1ENR |= RCC_APB1ENR_SPI2EN;

/* Init pins */
GPIO_InitStruct.GPIO_OType = GPIO_OType_PP;
GPIO_InitStruct.GPIO_PuPd = GPIO_PuPd_NOPULL;
GPIO_InitStruct.GPIO_Speed = GPIO_Speed_100MHz;

GPIO_InitStruct.GPIO_Mode = GPIO_Mode_AF;
GPIO_InitStruct.GPIO_Pin = GPIO_Pin_3;
GPIO_Init(GPIOC, &GPIO_InitStruct);
GPIO_PinAFConfig(GPIOC, GPIO_PinSource3, GPIO_AF_SPI2);

GPIO_InitStruct.GPIO_Pin = GPIO_Pin_10;
GPIO_Init(GPIOB, &GPIO_InitStruct);
GPIO_PinAFConfig(GPIOB, GPIO_PinSource10, GPIO_AF_SPI2);

/* Fill SPI settings */
SPI_InitStruct.SPI_DataSize = SPI_DataSize_8b;
SPI_InitStruct.SPI_BaudRatePrescaler =
SPI_BaudRatePrescaler_16;
SPI_InitStruct.SPI_Direction =
SPI_Direction_2Lines_FullDuplex;
SPI_InitStruct.SPI_FirstBit = SPI_FirstBit_MSB;
SPI_InitStruct.SPI_Mode = SPI_Mode_Master;

/* SPI mode */
SPI_InitStruct.SPI_CPOL = SPI_CPOL_Low;
SPI_InitStruct.SPI_CPHA = SPI_CPHA_1Edge;
SPI_InitStruct.SPI_NSS = SPI_NSS_Soft;

/* Disable first */
SPI_Cmd(SPI2, DISABLE);

/* Init SPI */
SPI_Init(SPI2, &SPI_InitStruct);

/* Enable SPI */
SPI_Cmd(SPI2, ENABLE);
}

```

```

void PCD8544_DrawCircle(char x0, char y0, char r,
PCD8544_Pixel_t color) {
    int16_t f = 1 - r;
    int16_t ddF_x = 1;
    int16_t ddF_y = -2 * r;

```

```

    int16_t x = 0;
    int16_t y = r;

    PCD8544_DrawPixel(x0, y0 + r, color);
    PCD8544_DrawPixel(x0, y0 - r, color);
    PCD8544_DrawPixel(x0 + r, y0, color);
    PCD8544_DrawPixel(x0 - r, y0, color);

    while (x < y) {
        if (f >= 0) {
            y--;
            ddF_y += 2;
            f += ddF_y;
        }
        x++;
        ddF_x += 2;

        PCD8544_DrawPixel(x0 + x, y0 + y, color);
        PCD8544_DrawPixel(x0 - x, y0 + y, color);
        PCD8544_DrawPixel(x0 + x, y0 - y, color);
        PCD8544_DrawPixel(x0 - x, y0 - y, color);

        PCD8544_DrawPixel(x0 + y, y0 + x, color);
        PCD8544_DrawPixel(x0 - y, y0 + x, color);
        PCD8544_DrawPixel(x0 + y, y0 - x, color);
        PCD8544_DrawPixel(x0 - y, y0 - x, color);
    }
}

void PCD8544_DrawFilledCircle(char x0, char y0, char r,
PCD8544_Pixel_t color) {
    int16_t f = 1 - r;
    int16_t ddF_x = 1;
    int16_t ddF_y = -2 * r;
    int16_t x = 0;
    int16_t y = r;

    PCD8544_DrawPixel(x0, y0 + r, color);
    PCD8544_DrawPixel(x0, y0 - r, color);
    PCD8544_DrawPixel(x0 + r, y0, color);
    PCD8544_DrawPixel(x0 - r, y0, color);
    PCD8544_DrawLine(x0 - r, y0, x0 + r, y0, color);

    while (x < y) {
        if (f >= 0) {
            y--;
            ddF_y += 2;
            f += ddF_y;
        }
        x++;
        ddF_x += 2;
        f += ddF_x;
    }
}

```

```
PCD8544_DrawLine(x0 - x, y0 + y, x0 + x, y0 + y, color);
PCD8544_DrawLine(x0 + x, y0 - y, x0 - x, y0 - y, color);

PCD8544_DrawLine(x0 + y, y0 + x, x0 - y, y0 + x, color);
PCD8544_DrawLine(x0 + y, y0 - x, x0 - y, y0 - x, color);
```

E

J

ДОДАТОК Г

Тестування системи та її компонентів

Таблиця - Тестових випадків

Номер теста	Назва теста	Значення вихідних даних	Очікуваний результат	Реакція	Висновок
Тест 1	Чорний ящик	Перевірка дисплея SSD1351 на биті пікселі	При виводі білого кольору, всі пікселі працюють вірно.	Битих пікселів немає	Пристрій працює вірно
Тест 2	Чорний ящик	Перевірка на вхід напруги до дисплея SSD1351	При підключення дисплея до ПК, живлення повинно подаватись в +3,3В.	Подача живлення на дисплей працює вірно.	Пристрій працює вірно
Тест 3	Чорний ящик	Перевірка на час відклику дисплея SSD1351 в с.	При виводі зображення дисплей відклик повинен бути не більше 2 с.	При виводі зображення відклик мінімальний	Пристрій працює вірно
Тест 4	Чорний ящик	Перевірка матриці дисплею SSD1351 на працездатність	При включення дисплею до живлення матриця повинна працювати стабільно	При включенні матриця дисплею, він працює стабільно	Пристрій працює вірно
Тест 5	Чорний ящик	Перевірка перетворювача інтерфейсів LM317 на працездатність	При підключення перетворювача інтерфейсів до живлення результат повинен перевищувати 10В	При підключення перетворювача інтерфейсів LM317 результат задовільняє	Пристрій працює вірно
Тест 6	Чорний ящик	Перевірка тока перетворювача інтерфейсів LM317	При підключення перетворювача інтерфейсів до мультиметра результат повинен перевищувати 3мл. Амп.	При підключення перетворювача інтерфейсів LM317 результат задовільняє	Пристрій працює вірно

Тест 7	Чорний ящик	Перевірка температури перетворювача інтерфейсів LM317	При підключення перетворювача інтерфейсів до інфрачервоного термометра результат повинен перевищувати 50 градусів.	При підключення перетворювача інтерфейсів LM317 результат задовільняє	Пристрій працює вірно
Тест 8	Чорний ящик	Перевірка напруги мікроконтролера STM32F407	При підключення мікроконтролера до ПК, живлення повинно подаватись більше +8В.	Подача живлення на мікроконтролера задовольняє вимогам.	Пристрій працює вірно
Тест 9	Чорний ящик	Перевірка роботи порту UART в мікроконтролері STM32F407	При підключення мікроконтролера STM32F407 до ПК, порт UART повинен передавати пакети які запланував користувач	Передача даних по порту UART проведена успішно	Пристрій працює вірно
Тест 10	Чорний ящик	Перевірка з'єднання пінів між мікроконтролером STM32F407 та дисплеєм SSD1351	При підключення мікроконтролера STM32F407 до SSD1351, піни повинні відправляти дані на дисплей	Передача даних за допомогою пінів мікроконтролером STM32F407 проведена успішно	Пристрій працює вірно
Тест 11	Чорний ящик	Перевірка температури мікроконтролером STM32F407	При підключення мікроконтролером STM32F407 до інфрачервоного термометра результат повинен перевищувати 50 градусів.	При підключення мікроконтролером STM32F407 результат задовільняє	Пристрій працює вірно
Тест 12	Чорний ящик	Перевірка збереження параметрів в енергонезалежній пам'яті за допомогою ПК по UART-інтерфейсу	При підключенні мікроконтролера STM32F407 до ПК, порт UART повинен зберігати параметри в енергонезалежній пам'яті	При збереження параметрів даних пам'яті за допомогою ПК по UART-інтерфейсу проведено успішно	Пристрій працює вірно

Тест 13	Чорний ящик	Перевірка встроєних світлодіодів мікроконтролері STM32F407	При підключення мікроконтролера STM32F407 до PC, та запису відповідного коду до IDE для підсвідки світлодіода він повинен засвітитись зеленим кольором	При підключення мікроконтролера STM32F407 до PC, світлодіод загорівся зеленим кольором	Пристрій працює вірно
Тест 14	Чорний ящик	Перевірка мікросхеми MCP2200T-I на паразитиздатність	При підключення мікросхеми MCP2200T-I до живлення по USB mini результат не повинен перевищувати 10В	При підключення мікросхеми MCP2200T-I результат задовільняє	Пристрій працює вірно
Тест 15	Чорний ящик	Перевірка на відклик клавіатури по порту UART	При підключення PC до мікроконтролера STM32F407 порт UART повинен передавати пакети від клавіатури з мінімальним відкликом	Передача даних з клавіатури по порту UART проведена задовільним відкликом	Пристрій працює вірно
Тест 16	Чорний ящик	Перевірка згенерованого пристроєм QR коду на достовірність	При вводу стрічки на екран, нам повинен згенеруватись QR код, який потім ми зможемо відсканувати	При сканування ми отримали стрічку, отже QR код згенерувався правильно	Пристрій працює вірно

Опис кроків кейс-тестів:

Тест 1:

За допомогою кода користувач задає параметри білого кольору, задля перевірки чи все працюють вірно, та не має битих пікселів.

Тест 2:

При підключення дисплея SSD1351 за допомогою блока напруги до живлення при тестуванні тестером результати вказують стабільні в 2,9В.

Тест 3:

Користувач виводить зображення на дисплей SSD1351 відклик дорввноє 1 с. що перевищує допустиме значення в 2 с.

Тест 4 :

Користувач вмикає матрицю після перевірок декілька хвилин помилок з матрицею не виявлено отже вано працює стабільно.

Тест 5:

Користувач робить підключення перетворювача інтерфейсів до живлення результат н становить 5В що не перевищувати стандартні 10В.

Тест 6:

При підключення перетворювача інтерфейсів за допомогою блока напруги до живлення, при тестуванні мультиметром що використовується для перевірки току, результати вказують ток в розмірі 0.57 мА.Амп що є допустимим значенням для перетворювача напруги.

Тест 7:

При підключення перетворювача інтерфейсів за допомогою блока напруги до живлення, при тестуванні інфрачервоним термометром що використовується для перевірки температури, результати вказують температуру в розмірі 33 градусів, що є допустимим значенням для перетворювача напруги.

Тест 8:

При підключення мікроконтроллера STM32F407 до ПК, відносно виводи D+ D- підключені за вимогами, за допомогою програмного коду перевірили напругу мікроконтроллера яка становить біля 5В.

Тест 9:

При підключення мікроконтролера STM32F407 до ПК, порт UART передає пакети які запланував користувач.

Тест 10:

Користувач підключає мікроконтролер STM32F407 до SSD1351, піни відправляють дані по порту SPI на дисплей вчасно.

Тест 11:

При підключення STM32F407 за допомогою блока напруги до живлення, при тестуванні інфрачервоним термометром що використовується для перевірки температури, результати вказують температуру в розмірі 3 градусів, що є допустимим значенням для мікроконтролера.

Тест 12:

Користувач підключає мікроконтролер STM32F407 до ПК, порт щоб зберегти параметри в енергонезалежній пам'яті використовується протокол передачі даних UART.

Тест 13:

Користувач підключає мікроконтролер STM32F407 до ПК, та записує відповідний коду до IDE для підсвідки світо діода, результатом є засвічений зеленим кольором світодіод.

Тест 14:

Користувач робить підключення мікросхеми MCP2200T-I до живлення по по USB mini результат становить 5В що не перевищувати стандартні 10В.

Тест 15:

Користувач зробив підключення РС до мікроконтроллера STM32F407 по порту UART, в результаті чого отримав переданий пекет від клавіатури з мінімальним відкликом.

Тест 16:

Користувач вводить до стрічки данні які потрібно закодувати, в результаті чого пристрій згенерує QR код, який потім можна відсканувати. Сканування проведено за допомогою смартфона, і результатом сканування є відповідна закодована стрічка данних.

Test-report:

Кількість пройдених позитивних тестів – 16

Кількість пройдених негативних тестів – 0

Загальна кількість пройдених тестів - 16

Кількість провалених позитивних тестів – 0

Кількість провалених негативних тестів – 0

Загальна кількість провалених тестів - 0