


**Національний технічний університет України
«Київський політехнічний інститут ім. Ігоря Сікорського»**

Факультет електроніки
Кафедра конструювання електронно-обчислювальної апаратури

«До захисту допущено»

Завідувач кафедри


Лисенко О.М.
(підпис) (ініціали, прізвище)

“12” червня 2020р.

**Дипломний проект
на здобуття ступеня бакалавра**

зі спеціальності 172 Телекомунікації та радіотехніка
(код та назва напрямку підготовки або спеціальності)

на тему: Генератор QR-коду

Виконав: студент IV курсу, групи ДК-62

Ярослав Володимирович Сокол

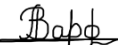
(прізвище, ім'я, по батькові)



(підпис)

Керівник доцент, к.т.н., А.Ю. Варфоломєєв

(посада, вчене звання, науковий ступінь, прізвище та ініціали)



(підпис)


Рецензент доц. каф. АМЕС, к.т.н. Павло Понович

(посада, вчене звання, науковий ступінь, прізвище та ініціали)



(підпис)

Засвідчую, що у цьому дипломному
проекті немає запозичень з праць інших
авторів без відповідних посилань.

Студент 
(підпис)

Київ - 2020 року


**Національний технічний університет України
«Київський політехнічний інститут ім. Ігоря Сікорського»**

Факультет електроніки
Кафедра конструювання електронно-обчислювальної апаратури

Освітньо-кваліфікаційний рівень – бакалавр телекомунікацій та радіотехніки
Спеціальність 172 Телекомунікації та радіотехніка
(код і назва)

ЗАТВЕРДЖУЮ

Завідувач кафедри



(підпис)

О.М. Лисенко
(ініціали, прізвище)

“09” _____ 03 _____ 2020р..

ЗАВДАННЯ
на дипломний проект студенту

Ярослав Володимирович Сокол

(прізвище, ім'я, по батькові)

1. Тема проекту Генератор QR-коду
керівник проекту Антон Юрійович Варфоломєєв, к.т.н., доцент
затверджені наказом по університету від _____ № 1196-с від 25.05.2020р
2. Термін подання студентом проекту 30 травня 2020 року
3. Вихідні дані до проекту Пристрій являє собою моноблочну конструкцію, кліматичне виконання УХЛ 5.1 по ГОСТ 15150-69. Пристрій повинен генерувати QR-коди. Габаритні розміри 80x80x20 мм, маса — не більше 200 г. Час напрацювання на відмову — не менше 10000 годин.

4. Зміст розрахунково-пояснювальної записки:
 - аналіз технічного завдання;
 - огляд існуючих засобів керування, патентний пошук;
 - розробка схеми електричної принципової;
 - обґрунтування вибору елементної бази та друкованої плати;
 - конструкторсько-технологічні розрахунки;
 - електричний розрахунок друкованої плати;
 - розрахунок надійності друкованої плати;
 - проектування у Altium Designer;
 - висновки.
5. Перелік графічного матеріалу (із зазначенням обов'язкових креслень, плакатів, презентацій тощо):
 - схема електрична принципова;
 - друкована плата;
 - складальне креслення друкованої плати.
6. Консультанти розділів проекту

Розділ	Прізвище, ініціали та посада консультанта	Підпис, дата	
		завдання видав	завдання прийняв

7. Дата видачі завдання 09.03.20

КАЛЕНДАРНИЙ ПЛАН

№ з/п	Назва етапів виконання Дипломного проекту	Термін виконання етапів проекту	Примітка
1	Аналіз технічного завдання	9.03.20-15.03.20	виконано
2	Вибір елементної бази та друкованої плати	16.03.20-29.03.20	виконано
3	Схемотехнічне проектування	30.03.20-08.04.20	виконано
4	Виконання креслень схеми електричної принципової	08.04.20-19.04.20	виконано
5	Конструкторсько-технологічні розрахунки	20.04.20-03.05.20	виконано
6	Електричний розрахунок друкованої плати	04.05.20-10.05.20	виконано
7	Проектування у Altium Designer	11.05.20-17.05.20	виконано
8	Виконання креслень друкованої плати та складального креслення друкованого вузла	18.05.20-24.05.20	виконано
9	Оформлення пояснювальної записки	25.05.20-30.05.20	виконано

Студент _____

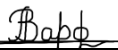


(підпис)

Я.В. Сокол

(прізвище та ініціали)

Керівник проекту _____



(підпис)

А.Ю. Варфоломєєв

(прізвище та ініціали)

АНОТАЦІЯ

Роботу викладено на 77 сторінках, вона містить 5 розділів, 37 ілюстрацій, 14 таблиць та 31 джерел в переліку посилань.

Об'єктом розробки є генератор QR-коду.

Предмет роботи — спосіб отримання згенерованого QR-коду.

Метою є створенням пристрою, який буде генерувати QR-код. Дана розробка призначена для ідентифікації товарів у торгівельному підприємстві.

Дипломний проект складається із вступу, 5 розділів, висновків по роботі, списку використаних джерел та додатків.

В першому розділі зроблено пошук існуючих аналогів та патентний пошук.

В другому розділі розглядається принцип роботи схеми та наводиться обґрунтування вибору елементної бази.

В третьому розділі розглядається проектування друкованого вузлу.

В четвертому розділі наведено розрахунки, пов'язані з проектуванням друкованого вузлу та розрахунки, що підтверджують правильність прийнятих конструкторських рішень.

В п'ятому розділі виконується розробка програмної частини та отримання результатів.

ABSTRACT

The work is presented on 77 pages, it contains 5 sections, 37 illustrations, 14 tables and 31 sources in the list of references.

The object of development is a QR-code generator.

The subject of the work is a method of obtaining the generated QR-code.

The goal is to create a device that will generate a QR code. This development is designed to identify goods in trade.

The diploma project consists of an introduction, 5 sections, conclusions on work, the list of the used sources and appendices.

In the first section the search of existing analogues and patent search is made.

The second section considers the principle of operation of the scheme and provides justification for the choice of element base.

The third section discusses the design of the printed circuit board.

The fourth section provides calculations related to the design of the printed circuit board and calculations that confirm the correctness of the design decisions.

In the fifth section, the software is developed and the results are obtained.





**Пояснювальна записка
до дипломного проекту**

на тему: **Генератор QR-коду**

Київ – 2020 року

ЗМІСТ

ВСТУП	4
Розділ 1. АНАЛІЗ ТЕХНІЧНОГО ЗАВДАННЯ ТА ПАТЕНТНИЙ ПОШУК . 5	
1.1. Аналіз технічного завдання.....	5
1.2. Огляд існуючих систем та патентний пошук	5
Розділ 2. СХЕМОТЕХНІЧНЕ ПРОЕКТУВАННЯ	11
2.1. Розробка структурної схеми.....	11
2.2. Опис алгоритму пристрою та генерації QR-коду	12
2.3. Вибір елементної бази.....	17
2.4. Розробка та опис схеми електричної принципової	25
Розділ 3. Проектування друкованого вузлу	32
3.1. Вибір і обґрунтування типу друкованої плати	32
3.2. Вибір і обґрунтування матеріалу друкованої плати	32
3.3. Вибір і обґрунтування класу точності друкованої плати	33
3.4. Вибір і обґрунтування методу виготовлення друкованої плати.....	35
3.5. Розміщення КЕ в монтажному просторі	37
3.6. Проектування друкованої плати у середовищі Altium Designer.....	40
Розділ 4. Розрахунки, що підтверджують правильність конструкторського рішення.....	43
4.1. Конструкторсько-технологічний розрахунок друкованої плати	43
4.2. Електричний розрахунок друкованої плати	49
4.3. Розрахунок основних показників надійності друкованого вузлу	51
4.4. Розрахунок віброміцності друкованого вузла	56

<h1 style="margin: 0;">ДК62.469515.001 ПЗ</h1>				
<i>Змн.</i>	<i>Арк.</i>	<i>№ доким.</i>	<i>Підпис</i>	<i>Дата</i>
<i>Розробив</i>		<i>Я.В. Сокол</i>		
<i>Перевірив</i>		<i>О.І. Лисенко</i>		
<i>Реценз.</i>				
<i>Н. Контр.</i>		<i>О.І. Лисенко</i>		
<i>Затвердив</i>		<i>А.Ю. Варфоломеев</i>		
<h2 style="margin: 0;">Генератор QR-коду</h2>			<i>Літ.</i>	<i>Арк.</i>
			1	77
НТУУ «КПІ ім. Ігоря Сікорського», ФЕЛ, гр. ДК-62				

Розділ 5. РОЗРОБКА ПРОГРАМНОГО ЗАБЕЗПЕЧЕННЯ	61
5.1. Програмування мікроконтролера STM32F411RET6	61
5.2. Структура даних	61
5.3. Опис лістингу	63
5.4. Тестування роботи програмного забезпечення	69
Висновки	72
Література	74
Додаток А — Технічне завдання на проектування	
Додаток Б — Технічне завдання на проектування	
Додаток В — Матеріали конференції	

					<i>ДК62.469515.001 ПЗ</i>	Арк.
<i>Змн.</i>	<i>Арк.</i>	<i>№ докум.</i>	<i>Підпис</i>	<i>Дата</i>		2

Список використаних скорочень

QR	—	Quick Response
ПЛІС	—	Програмована Логічна Інтегральна Схема
VGA	—	Video Graphics Array
USB	—	Universal Serial Bus
ASCII	—	American Standard Code for Information Interchange
OLED	—	Organic Light-Emitting Diode
UART	—	Universal Asynchronous Receiver/Transmitter
SPI	—	Serial Peripheral Interface
I2C	—	Inter-Integrated Circuit
ЕОА	—	Електронно Обчислювальна Апаратура
ПМ	—	Посадкове Місце
КЕ	—	Конструктивні Елементи
ТЗ	—	Технічне Завдання
AD	—	Altium Designer
ІС	—	Інтегральна Схема
ДП	—	Друкована Плата
ОДП	—	Одностороння Друкована Плата
ДДП	—	Двостороння Друкована Плата
БДП	—	Багатошарова Друкована Плата
ГДП	—	Гнучка Друкована Плата
КМ	—	контактний майданчик
ДВ	—	друкований вузол
МО	—	Монтажний Отвір
ЕРЕ	—	ЕлектроРадіо-Елементи

					<i>ДК62.469515.001 ПЗ</i>	<i>Арк.</i>
<i>Змн.</i>	<i>Арк.</i>	<i>№ докум.</i>	<i>Підпис</i>	<i>Дата</i>		<i>3</i>

ВСТУП

QR-код (Quick Response) є двомірним матричним штрих-кодом, який призначений для зберігання невеликих обсягів інформації, головним чином на друківаних носіях, яку можна швидко та зручно зчитати і розпізнати як за допомогою спеціальних сканерів, так і камерою мобільного телефону. В загальному випадку у QR-коді можна закодувати будь-яку інформацію, проте найбільш часто у ньому зберігають текстову інформацію, таку як номери телефонів, посилання на сайти, геометки, номери криптовалютних гаманців тощо [1, 2].

Не дивлячись на існування альтернативних версій двомірних штрих-кодів, таких як Aztec code, Data Matrix, саме QR-коди наразі набули найбільшого розповсюдження, тому в цій роботі мова йдеться саме про них.

Створення QR-коду може здійснюватись програмно — із використанням мікропроцесора чи мікроконтролера або апаратно — за допомогою спеціалізованого пристрою. Кожен з цих підходів має свої переваги та недоліки. Так, програмні реалізації є більш гнучкими і простими, водночас, будучи розгорнутими на неспецифічних контролерах чи процесорах, вони потребують більшого енергоспоживання. Апаратні реалізації навпаки є більш складними, не володіють значною гнучкістю, проте можуть бути більш швидкодіючими та споживати значно менше енергії, що робить їх використання більш доцільним у різноманітних портативних пристроях з батарейним живленням.

Метою роботи є створення пристрою, який буде генерувати QR-код. Такий пристрій повинен мати свої особливості. Для досягнення даної мети виконано наступні задачі, серед яких головними являються:

- Аналіз існуючих систем;
- Визначення алгоритму генерування QR-коду;
- Розробка схемотехнічних та рішень апаратно-програмного комплексу та їх реалізація;
- Розрахунок елементів друкованої плати пристрою.

					<i>ДК62.469515.001 ПЗ</i>	<i>Арк.</i>
<i>Змн.</i>	<i>Арк.</i>	<i>№ докум.</i>	<i>Підпис</i>	<i>Дата</i>		4

Розділ 1. АНАЛІЗ ТЕХНІЧНОГО ЗАВДАННЯ ТА ПАТЕНТНИЙ ПОШУК

1.1. Аналіз технічного завдання

На даний момент існує багато способів використання генератора QR-коду, але в основному він являється частиною певного пристрою. Проте незважаючи на це, задачею даного дипломного проекту — розробити продукт, який буде генерувати QR-код. Для цього потрібно вирішити наступні задачі:

- Що являється джерелом даних?
- За допомогою чого або як буде виведений результат?

Для вирішення даних питань спочатку потрібно розглянути які існують пристрої, щоб було зрозуміло чи доцільно використовувати саме такий метод вирішення питання.

1.2. Огляд існуючих систем та патентний пошук

На сьогодні існує наступний пристрій, який має властивість генерування QR-коду і має наступний вигляд (Рис.1.1) [3]:



Рис.1.1 — Зовнішній вигляд терміналу з генератором QR-коду

					ДК62.469515.001 ПЗ	Арк. 5
Змн.	Арк.	№ докум.	Підпис	Дата		

Даний пристрій призначений для банківських справ з мобільним платіжками та авторизацією. Генерування QR-коду виконується автоматично на основі налаштованих даних. Виконавши генерування QR-коду, для сплати товарів необхідно за допомогою телефону відсканувати згенерований QR-код і виконати платіжку. Також даний пристрій має клавіатуру на випадок, якщо у покупця немає з собою смартфона для сканування QR-коду, і таким чином виконується платіжка через банківську карту вводячи свої дані. Також мається достатньо схожий пристрій який зображено нижче на Рис.1.2 [4]:



Рис.1.2 — Зовнішній вигляд POS терміналу

Даний пристрій на відмінну від попереднього має ще властивість сплати за допомогою банківської карти, яку можна вставити в пристрій.

Аналізуючи їх переваги та недоліки можна сказати наступне: так, дані пристрої мають по 2-3 види сплачування товарів, але в більшості людей мається смартфон і з кожним наступним роком така необхідність, як сплата вручну буде поступово зменшуватись. Тому тут можна запропонувати створити пристрій, за допомогою якого можна сплачувати лише через мобільний телефон, а тобто створити звичайний генератор QR-коду.

Також було здійснено патентний пошук аналогів та прототипів пристрою, що розробляється. В основному всі пристрої пов'язані з

					<i>ДК62.469515.001 ПЗ</i>	<i>Арк.</i>
<i>Змн.</i>	<i>Арк.</i>	<i>№ докум.</i>	<i>Підпис</i>	<i>Дата</i>		6

генеруванням QR-кодом, виконують генерацію без допомоги певного портативного пристрою (наприклад, за допомогою смартфона або на сервері). Тому така розробка втрачає певну актуальність. Розглянемо за приклад наступний патент [5], де зображення конструкції представлено нижче на Рис.1.3:

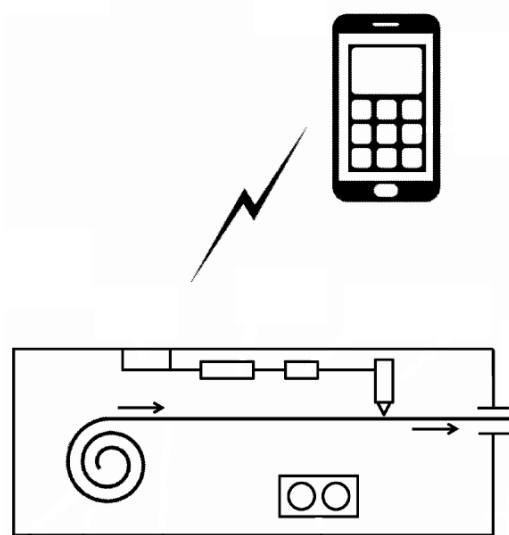


Рис.1.3 — Схема портативного принтера етикеток QR-коду

Дана розробка отримує доступ до смартфона, який відіграє роль контролера, за допомогою якого можна ввести потрібну інформацію. Після чого введена інформація перетворюється в QR-код у вигляді зображення і в кінці дане зображення друкується. Такий пристрій використовується для ідентифікації товарів у торговельних закладах. Такий пристрій має достатньо переваг, але за рахунок того, що він може мати як бездротове так і дротове підключення, тому його вартість буде значно збільшуватись за рахунок кількості периферій. Тому тут можна запропонувати наступне: замість смартфона підключити до нього генератор QR-коду з клавіатурою. Таким чином вартість даного пристрою можна зменшити і також габаритні розміри пристрою потенційно теж можуть зменшитись.

Також було розглянуто ще один патент [6], принцип роботи якого зображено нижче на Рис.1.4:

					<i>ДК62.469515.001 ПЗ</i>	Арк.
Змн.	Арк.	№ докум.	Підпис	Дата		7

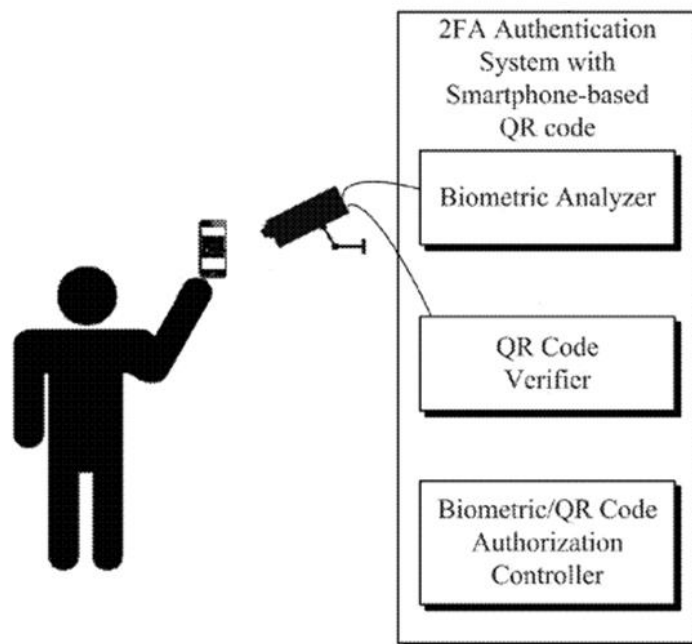


Рис.1.4 — Схема користування візуальної біометричної автентифікації

Дана розробка складається з двофакторної аутентифікації, але нам важлива лише перший етап. На даному етапі виконується наступне: спочатку користувач за допомогою смартфона генерує QR-код, після чого дане зображення зі смартфона підноситься до камери та розпізнається. Такий метод аутентифікації має свої зручності та переваги. Для того, щоб вирішити ... але тут стоїть наступне питання “Які дані використовуються для генерації QR-коду?”. Зловмисникам достатньо отримати доступ до даного смартфона і можна вважати, що дана аутентифікація є зламанною. Тому тут пропонується розробити портативний пристрій, подібним до смартфона, який буде генерувати QR-код. Єдине що потрібно уточнити, якими вхідними даними повинні бути (вхідними даними рекомендується робити змінними з часом). Тому за рахунок того, що вхідними даними можна створити на аналоговій частині і отримати доступ до такого пристрою не буде можливості (лише за допомогою крадіжки даного пристрою), тому такий пристрій може ускладнити систему захисту.

Даний пристрій було спробовано реалізувати за допомогою технології ПЛІС. Для цього було розроблено генератор QR-коду на поширеній платі DE1-SoC. Дана плата має інтерфейси для підключення клавіатури та VGA —

					<i>ДК62.469515.001 ПЗ</i>	Арк.
Змн.	Арк.	№ докум.	Підпис	Дата		8

монітору, тому введення та виведення інформації про QR-код виконувалося саме за допомогою них. Його принципом роботи був наступним:
 користувач вводить необхідний текст з клавіатури; даний текст паралельно відображається на моніторі;
 при натисканні на кнопку “Enter” виконується генерування QR-коду та його виведення у вигляді зображення на монітор;
 за необхідності повторне введення коду здійснюється при натисканні на клавішу «Esc».

Загальна структурна схема пристрою та структурна схема IP-блоку генерування QR-кодів представлені на Рис.1.5 а) та б) відповідно:

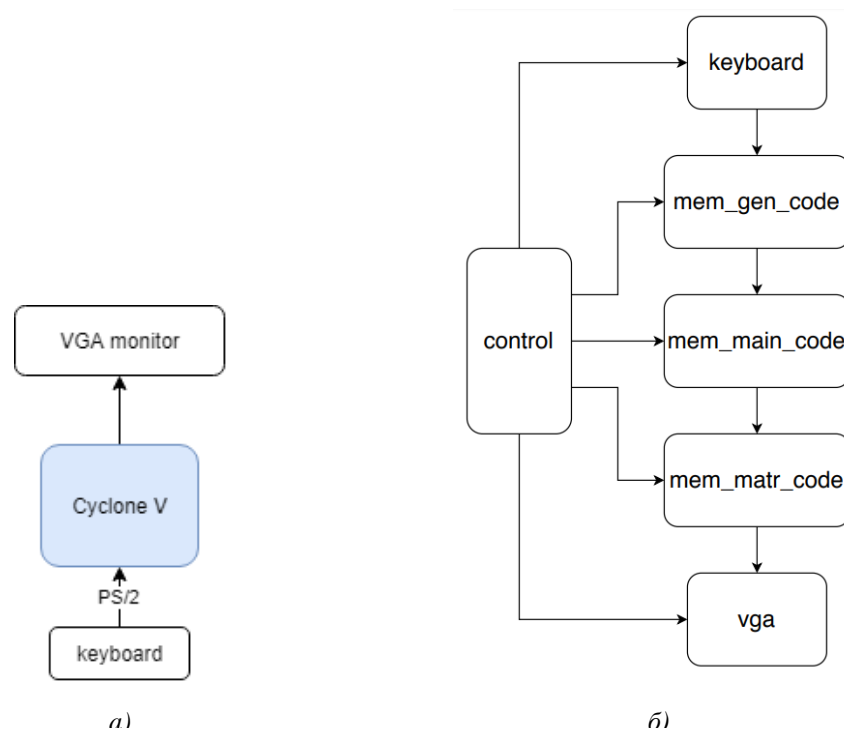


Рис.1.5 — (а) Загальна структурна схема пристрою та (б) структурна схема IP-блоку генерування QR-кодів

IP-блок для генерування QR-кодів реалізовано на мові Verilog [7] і складається з наступних модулів: control, keyboard, mem_gen_code, mem_main_code, mem_matr_code, VGA. Модуль control здійснюється для керування скінченного автомата, який має свої стани для генерації QR-коду. Модуль keyboard виконується для зчитування сигналів клавіатури. Модулі mem_gen_code, mem_main_code та mem_matr_code виконують саму

генерацію QR-коду. Модуль vga формує необхідні синхросигнали для правильної роботи VGA монітору [8]. Результат роботи створеного генератора QR-коду показано нижче на Рис.1.6:

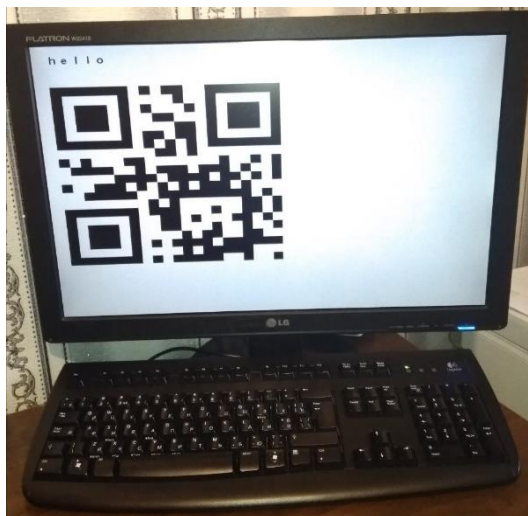


Рис.1.6 — Результат генерування QR-коду

Як для портативного пристрою мається головний недолік — плата під час розробки буде мати більші габарити. Тому розробку генератора QR-коду було вирішено побудувати на мікроконтролері.

Висновок: Було проведено аналіз існуючих пристроїв та виконано патентний пошук, під час чого було встановлено, що розроблюючий пристрій може знадобитись в наступному: для сплати покупок, ідентифікації товарів у торговельних закладах або для охоронних систем. З даних необхідностей можна сказати наступне: у якості джерела вхідних даних може бути клавіатура або генератор даних, а результат генерування може виводитись за допомогою дисплея або принтера. Також було розроблено та протестовано генератор QR-коду за допомогою технології ПЛІС. Опираючись на необхідність та актуальність пристроїв було вирішено розробити генератор QR-коду на основі мікроконтролера, на вході якого будуть прийматись певні дані (можна використати клавіатуру), і виведення результату на екран.

					<i>ДК62.469515.001 ПЗ</i>	<i>Арк.</i>
<i>Змн.</i>	<i>Арк.</i>	<i>№ докум.</i>	<i>Підпис</i>	<i>Дата</i>		<i>10</i>

Розділ 2. СХЕМОТЕХНІЧНЕ ПРОЕКТУВАННЯ

2.1. Розробка структурної схеми

Структурна схема генератора QR-коду зображено нижче на Рис.2.1:

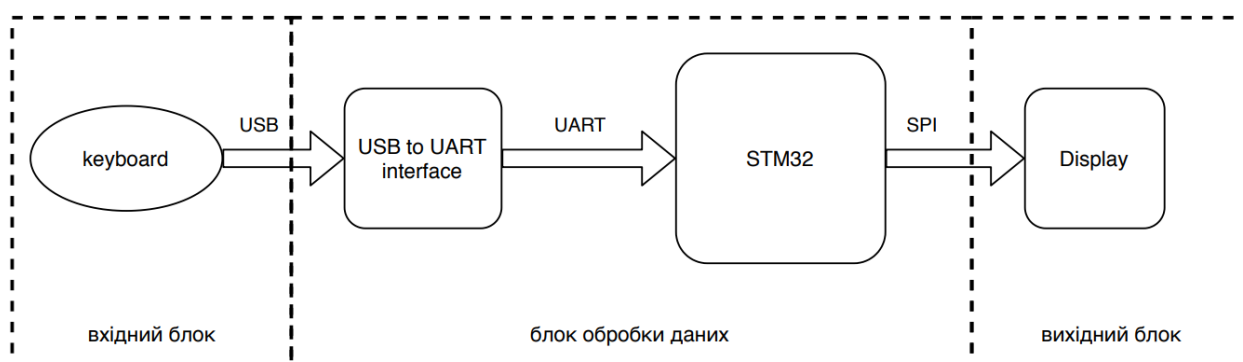


Рис.2.1 — Структурна схема пристрою

Дана схема складається з трьох частин:

- 1) Вхідний блок;
- 2) Блок обробки даних;
- 3) Вихідний блок.

Для розробки такої схеми спочатку необхідно визначити з чого повинні складатися кожен блок.

Вхідний блок:

Даний блок повинен мати:

- підключення до джерела живлення;
- мати підключення до вхідних даних (до клавіатури).

Згідно технічного завдання напруга живлення схеми повинно мати напругу 5 або 3,3 В. Тому, в тому числі для зменшення габаритних розмірів доцільно використовувати у якості роз'єму для живлення USB micro B.

Наступний роз'єм розрахований на використання у якості передачі даних з клавіатури. Оскільки наразі найбільш розповсюджені клавіатури мають штекери USB A типу, тому такого ж типу і буде мати даний роз'єм.

Блок обробки даних:

Даний блок повинен мати:

- Перетворювачі напруги;

									Арк.
									11
Змн.	Арк.	№ докум.	Підпис	Дата	ДК62.469515.001 ПЗ				

- Мікроконтролер.
- Перетворювач інтерфейсу.

Перший перетворювач напруги обов'язково повинен бути на вході схеми, який буде приймати вхідну напругу живлення. Дану напругу доцільно перетворити у 5 В, оскільки дана напруга необхідно для живлення клавіатури. Також можуть знадобитись додаткові перетворювачі напруги для мікроконтролеру, перетворювачу інтерфейсу та OLED-дисплею.

Мікроконтролер буде обрано опираючись на аналіз розповсюджених версій та маючих інтерфейсів. Перетворювач інтерфейсів повинен перетворити отримані дані з клавіатури для мікроконтролера, тобто перетворити USB інтерфейс в UART.

Вихідний блок:

Даний блок повинен відображати результат генерації QR-коду. Він повинен мати невеликі розміри і після його генерування повинен мати можливість відсканувати його. Для такої задачі найкраще може підійти OLED дисплей.

2.2. Опис алгоритму пристрою та генерації QR-коду

Алгоритм даного пристрою закладається в наступному:

- 1) Підключити пристрій до джерела живлення та клавіатури;
- 2) Натиснути на кнопку для перезавантаження пристрою;
- 3) Ввести текст з клавіатури;
- 4) Натиснути на кнопку "Enter" (після чого виконується генерація QR-коду);
- 5) За необхідності натиснути знову на кнопку "Enter" для повторного введення тексту.

Для спрощення алгоритму використано обмеження, які полягають в наступному:

- Генерування здійснюється лише для першої версії коду, яка має розмір матриці 21x21 піксель.

					ДК62.469515.001 ПЗ	<i>Арк.</i>
<i>Змн.</i>	<i>Арк.</i>	<i>№ докум.</i>	<i>Підпис</i>	<i>Дата</i>		12

- Застосовується рівень корекції помилок M типу. Даний тип корекції передбачає використання у кодї надлишкової інформації, що дозволяє відновити вміст закодованих даних у випадку пошкодження до 15% зображення коду (такий тип є найбільш розповсюдженим).

Виходячи з даних обмежень, код може шифрувати не більше ніж 14 символів. Для генерації QR-коду спочатку необхідно мати масив даних у вигляді ASCII коду. Генерація QR-коду виконується з чотирьох етапів, структурна схема алгоритму якого зображено нижче на Рис.2.2:



Рис.2.2 — Структурна схема алгоритму генерації QR-коду

На першому етапі даний масив потрібно перетворити у наступну послідовність:

- 1) Перші 4 біти зберігають тип кодування. Даний генератор використовує лише побайтове кодування, тому дане поле містить послідовність «0100». Загалом існують ще цифрове та літерно-числове кодування,

для яких перше поле має заповнюватись бітами «0001» та «0010» відповідно.

- 2) Наступні 2 байти зберігають обсяг інформації, що зашифрована в кодї — це звичайне двійкове число.
- 3) Наступні N байт (кількість залежить від типу корекції та номеру версії) зберігають корисну інформацію (власне дані, які були отримані з клавіатури). Даний генератор може мати до 14 байтів даних. Невикористані байти заповнюються наступними константами по черзі: «11101100» та «00010001».
- 4) Останні 4 біти заповнюються нулями для того, щоб послідовність містила цілу кількість байт.

На другому етапі виконується генерування байтів корекції. Для цього використовується алгоритм Ріда-Соломона [9]. Для застосування даного алгоритму необхідно спочатку підготувати масив байтів корекції. Даний масив обирається за рівнем корекції та номером версії коду. Оскільки використовується рівень корекції M та перший номер версії, то масив байтів корекції містить наступні 10 елементів: {251, 67, 46, 61, 118, 70, 64, 94, 32, 45}. Для даного алгоритму також необхідні таблиці прямого та зворотного полів Галуа, які зберігають значення від 0 до 255 [10]. Сам алгоритм передбачає циклічне виконання наступних дій (кількість ітерацій циклу визначається кількістю байт даних, що мають бути закодовані) [9]:

- 1) Перший елемент масиву даних зберігається у змінній a і видаляється з масиву (всі наступні значення зсуваються на одну клітинку вліво, останній елемент заповнюється нулем);
- 2) Якщо a дорівнює нулю, то пропустити наступні дії і перейти до наступної ітерації циклу;
- 3) У таблиці зворотного поля Галуа визначається значення комірки, що відповідає числу a і її значення заноситься у змінну b : $b = GF^{-1}[a]$;
- 4) Далі для N перших елементів, де N — кількість байтів корекції, i — лічильник циклу;

					ДК62.469515.001 ПЗ	Арк. 14
Змн.	Арк.	№ докум.	Підпис	Дата		

- 5) до i -го значення генеруючого многочлена додається значення b і зберігається у змінній c ;
- 6) якщо b більше 254, використовується залишок від ділення b на 255;
- 7) знайти відповідне c значення в таблиці поля Галуа і виконати побітову операцію суми по модулю 2 (XOR) з i -м значенням підготованого масиву і записати отримане значення в i -у комірку підготовленого масиву.

На третьому етапі виконується генерування матриці: розміщуючи узори для вирівнювання, код маски, рівень корекції та смуги синхронізації відповідно з Рис.2.3 [11]:



Рис.2.3 — Розміщення службової інформації на QR-кодi

Дана інформація містить наступне [9]:

- Узори для вирівнювання завжди розташовуються в 3 кряях (верхній лівий кут, верхній правий кут та нижній лівий кут). Матриця коду також може містити додаткові узори для вирівнювання, що розміщуються через кожен 21 комірку матриці по висоті та ширині. Додаткові узори наявні у версіях QR-кодів вище 2-ї, тому в даному алгоритмі вони не застосовуються.
- Тип маски та рівень корекції задаються 15-бітним числом. Оскільки в даному алгоритмі використовується рівень корекції типу M та тип маски під номером 1, то код типу маски та рівня корекції містить бітову послідовність наступного вигляду: 101000100100101.

- Полоси синхронізації розміщуються між узорами для вирівнювання і по чергово заповнюються нулями та одиницями.
- Номер версії коду складається з 15 бітів, обирається згідно номеру версії QR-коду і використовується в 7 версії та більше.

На останньому етапі виконується заповнення байтів з певною послідовністю, які були згенеровані на другому етапі, відповідно до Рис.2.4 [9]:

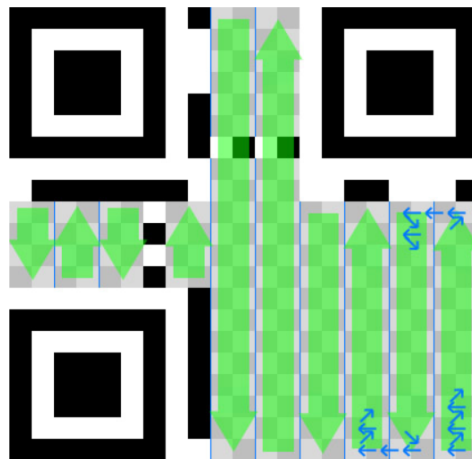


Рис.2.4 — Розміщення даних на QR-коді

Після такого розташування для цих даних накладається маска, відповідно до Рис.2.5 [12] (використовується маска 1-го типу, її код 001).

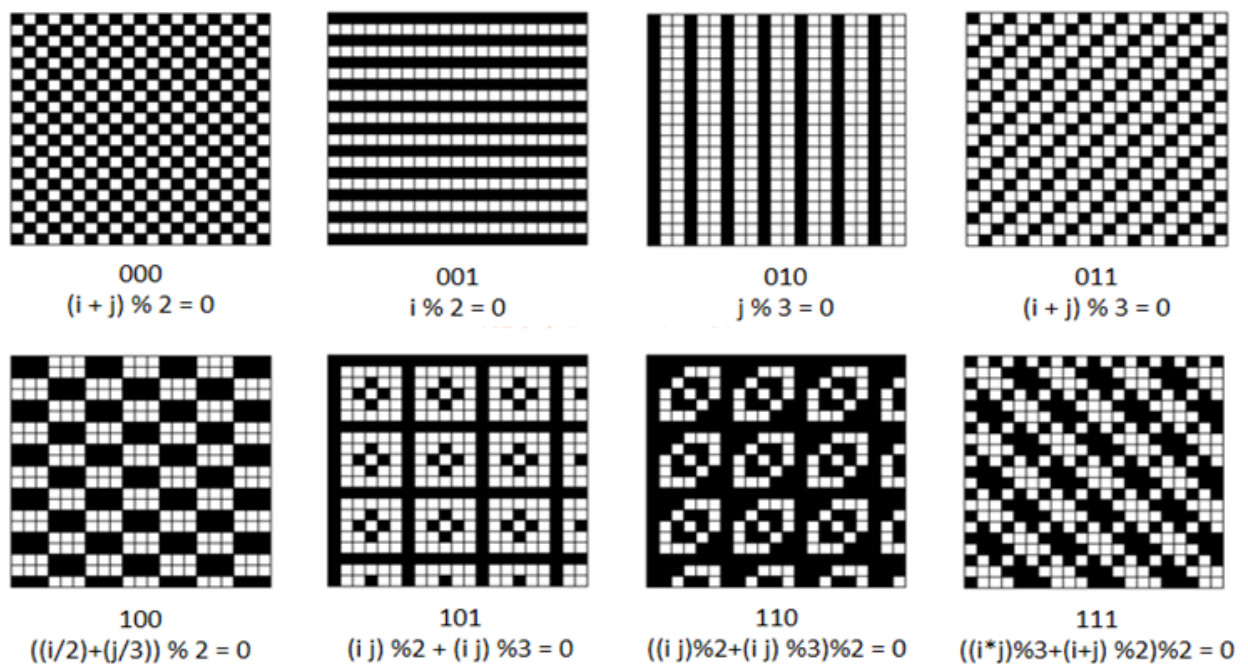


Рис.2.5 — Типи масок: операція «%» позначає остачу від ділення

2.3. Вибір елементної бази

Перелік елементів включає наступні типи електронних компонентів:

- Мікроконтролер
- Дисплей
- Роз'єми
- Кварцевий резонатор
- ІС
- Конденсатори
- Резистори

Для початку оберемо мікроконтролер. Він повинен виконувати необхідні функції для підтримки передачі даних (підтримка UART, SPI або I2C інтерфейсів), які будуть оброблюватись за допомогою мікроконтролеру. Найбільш об'єктивним рішенням являється обирання 32-бітних мікроконтролерів ARM, адже вони наразі є найбільш розповсюдженими і різниця між цінами не є значною. Одним з найпопулярніших варіантів є мікроконтролер STM32F411RET6 з корпусом LQFP64 (Рис.2.6) [13], який має наступні параметри:

- Максимальна тактова частота: 100 МГц
- Розмір програмної пам'яті: 512 кБ
- Розмір RAM пам'яті: 112 кБ
- Напруга живлення: 1.7...3.6 В
- Струм споживання: 240 мА
- Робоча температура: -40 ... +85 °С

					<i>ДК62.469515.001 ПЗ</i>	Арк.
Змн.	Арк.	№ докум.	Підпис	Дата		17

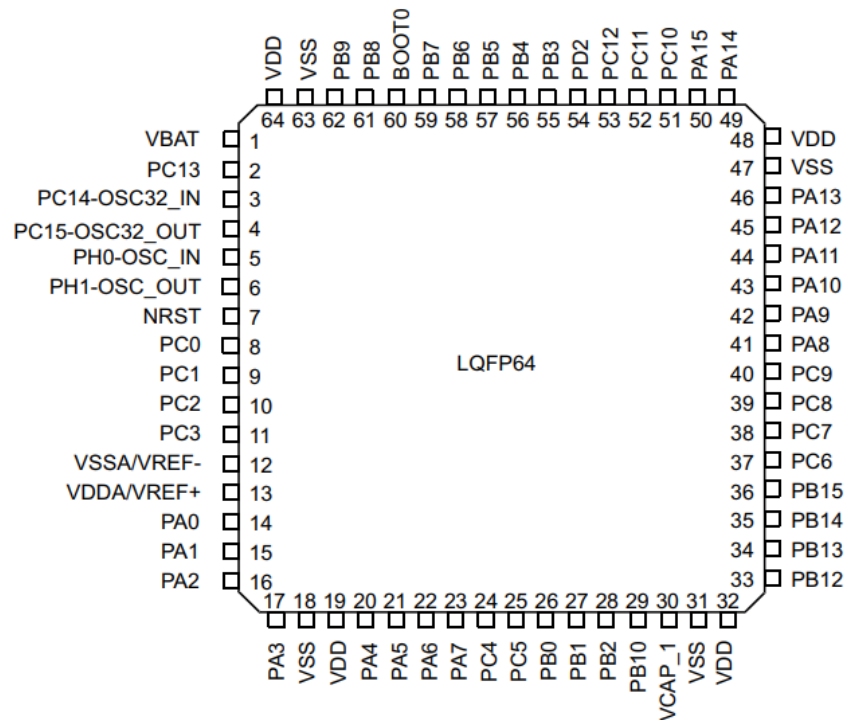


Рис.2.6 — Мікроконтролер STM32F411RE у корпусі LQFP64

Далі обираємо дисплей, за допомогою якого буде виконуватись відображення результату. Оскільки результатом відображення буде виведення тексту та самого QR-коду, то достатньо він був монохромним. З урахуванням того, що QR-код має квадратні розміри, тому сам дисплей повинен мати квадратні габарити. В результаті з такими вимогами найдешевший варіант являється OLED-дисплей на контролері SSD1327 RGS15128128WR000 [14], який зображено нижче на Рис.2.7 [15] і має наступні параметри:

- Розширення екрана: 128x128 піксель
- Розмір екрана: 1,5 дюйма
- Габаритні розміри: 33,8x43,7x1,41 мм
- Інтерфейс модуля: 4-wire SPI, I2C
- Напруга живлення: 2,6...3,5 В (3,3 В)
- Струм споживання: 40 мА
- Робоча температура: -40...+85 °С



Рис.2.7 — OLED-дисплей з SSD1327

Далі обираємо роз'єми, яких повинно бути 2. Перший з них повинен бути USB типу А. Вибір виконується в першу чергу за мінімальними габаритними розмірами, а потім обирається найдешевший варіант. З даними вимогами було обрано роз'єм UJ2-AH-1-SMT [16] з наступними параметрами:

- Стандарт: USB 2.0
- Робоча температура: -40...+105 °С

Наступним роз'ємом повинен бути USB-micro типу В і також мінімальними габаритними розмірами. Тому тут було обрано роз'єм 47346-0001 [17] з наступними параметрами:

- Стандарт: USB 2.0
- Робоча температура: -30...+85 °С

Оскільки мікроконтролеру потрібен кварцевий резонатор, тому наступним оберемо його. Він повинен мати частоту 8 МГц та ємність навантаження 12 пФ. Такі параметри мають 4 серії (12SMX-B [18], 7A [19], CX5032 [20], HC49 [21]), які мають 4 різних параметри: максимальна робоча температура, допустиме відхилення, рівень збудження та ціна. Дані параметри зображено нижче в Таблиця 2.1:

					<i>ДК62.469515.001 ПЗ</i>	Арк.
Змн.	Арк.	№ докум.	Підпис	Дата		19

Таблиця 2.1 — Основні параметри ІС, що порівнюються

Назва серії	Максимальна робоча температура, °С	Допустиме відхилення, ppm (10^{-6} Гц)	Рівень збудження, мкВт	Ціна, \$
12SMX-B	+70	30	100	1,08
7A	+85	30	100	0,539
CX5032	+125	50	500	0,939
HC49	+70	10	500	0,569
Ваговий коеф. b_j	0.4	0.2	0.2	0.2

В даному виборі найбільш важливішим параметром є максимальна робоча температура, оскільки інші параметри практично не впливають на результат. Далі складемо матрицю параметрів $|X|$ згідно Таблиця 2.1:

$$|X| = \begin{array}{|c|c|c|c|} \hline 70 & 30 & 100 & 1,08 \\ \hline 85 & 30 & 100 & 0,539 \\ \hline 125 & 50 & 500 & 0,939 \\ \hline 70 & 10 & 500 & 0,569 \\ \hline \end{array}$$

Проаналізувавши матрицю $|X|$, можна її перетворити таким чином, щоб найбільшому значенню матриці відповідала найкраща якість ІС, а найменшому — найгірша якість ІС. Для такого перетворення кожен параметр матриці необхідно визначити за наступною формулою (2.1):

$$Y_{ij} = \frac{1}{x_{ij}}, \quad (2.1)$$

де $i = \overline{1, n}$ — кількість вибраних ІС;

$j = \overline{1, m}$ — кількість параметрів ІС.

Оскільки максимальна робоча температура та рівень збудження зі збільшенням покращує якість, тому матриця приведених параметрів $|Y|$ має вигляд:

Y=	70	0,033	100	0,926
	85	0,033	100	1,855
	125	0,02	500	1,065
	70	0,1	500	1,757

Далі матрицю |Y| за допомогою нормування параметрів приведемо до матриці |A|, яке виконується за формулою (2.2). Нормування параметрів виконуємо за наступною формулою:

$$a_{ij} = \frac{\max_j y_{ij} - y_{ij}}{\max_j y_{ij}}, \quad (2.2)$$

де $\max_j y_{ij}$ — максимальний елемент в стовпчику |Y|; y_{ij} — поточне значення елемента в стовпчику |Y|.

A=	0,44	0,667	0,8	0,501
	0,32	0,667	0,8	0
	0	0,8	0	0,426
	0,44	0	0	0,053

Для узагальнення аналізу системи параметрів вводяться оціночну функцію і визначається виразом (2.3):

$$Q_i = \sum_{j=1}^m a_j \cdot b_j, \quad (2.3)$$

де b_j — ваговий коефіцієнт.

$$Q_1 = 0,44 \cdot 0,5 + 0,667 \cdot 0,2 + 0,8 \cdot 0,2 + 0,501 \cdot 0,1 = 0,570;$$

$$Q_2 = 0,32 \cdot 0,5 + 0,667 \cdot 0,2 + 0,8 \cdot 0,2 + 0 \cdot 0,1 = 0,421;$$

$$Q_3 = 0 \cdot 0,5 + 0,8 \cdot 0,2 + 0 \cdot 0,2 + 0,426 \cdot 0,1 = 0,245;$$

$$Q_4 = 0 \cdot 0,44 + 0 \cdot 0,2 + 0 \cdot 0,2 + 0,053 \cdot 0,1 = 0,187.$$

Значення оціночної функції для вибраних ІС наведені в Таблиця 2.2:

					<i>ДК62.469515.001 ПЗ</i>	Арк.
Змн.	Арк.	№ докум.	Підпис	Дата		21

Таблиця 2.2 — Значення оціночної функції

Назва серії	Q
12SMX-B	0,570
7A	0,421
CX5032	0,245
HC49	0,187

Після визначення всіх оцінюючих функцій порівнюємо результати та обираємо ту серію, яка має найменше значення Q, оскільки це відповідає вимогам для обирання кращої серії ІС. Проаналізувавши результати Таблиця 2.2 можна зробити висновок, що найкращий варіантом буде серії HC49. Тому обираємо LFX TAL031826Bulk з корпусом HC49S [21], який має наступні параметри:

- Частота: 8 МГц
- Ємність навантаження: 12 пФ
- Допустиме відхилення: 10 ppm(10^{-6} Гц)
- Рівень збудження: 500 мкВт
- Робоча температура: -20...+70 °С

Далі обираємо всі ІС в схемі. Тому спочатку обираємо ІС, яка буде перетворювати інтерфейс з USB в UART. Обирати потрібно таким чином, щоб напруга живлення була 5 В або 3,3 В. Найдешевший варіант з такими умовами являється ІС MCP2200T-I/SO з корпусом SOIC-20 [22] і має наступні параметри:

- Напруга живлення: 3...5,5 В
- Діапазон даних: 12 Мб/с
- Струм споживання: 13 мА
- Робоча температура: -40...+85 °С

Для такої мікросхеми необхідно ще один кварцевий резонатор. Його частота повинна складати 12 МГц, тому тут обираємо LFX TAL031823Bulk з корпусом HC49S [21], який має наступні параметри:

									Арк.
									22
Змн.	Арк.	№ докум.	Підпис	Дата	<i>ДК62.469515.001 ПЗ</i>				

- Частота: 12 МГц
- Ємність навантаження: 12 пФ
- Допустиме відхилення: 10 ppm(10^{-6} Гц)
- Рівень збудження: 500 мкВт
- Робоча температура: -20...+70 °С

Далі оберемо перетворювачі напруги. Він повинен перетворити напругу з 5 В у 3,3 В, маючи вихідний струм 300 мА. З даними параметрами мається перетворювач напруги LP3981ILD-3.3 з корпусом WSON-6 [23] і має наступні параметри:

- Вхідна напруга: 2,7...6 В
- Вихідна напруга: 3,3 В
- Вихідний струм: 300 мА
- Струм споживання: 0,21 мА
- Робоча температура: -40...+125 °С

Далі знову оберемо перетворювач напруги, але з іншими параметрами. Тут він повинен перетворити напругу з 5 В у 15 В і мати вихідний струм 100 мА. Обравши ІС з даними параметрами мається перетворювач напруги TLV76015DBZR з корпусом SOT-23-3 [24] і має наступні параметри:

- Вхідна напруга: 4,2...30 В
- Вихідна напруга: 15 В
- Вихідний струм: 100 мА
- Струм споживання: 2мА
- Робоча температура: -40...+125 °С

Наступний перетворювач напруги повинен перетворити вхідну напругу в діапазоні 0...15 В у напругу 5 В з вихідним струмом 950 мА. Обираємо ІС LD1117S50CTR з корпусом SOT-223-3 [25] і наступними параметрами:

- Вхідна напруга: 0...15 В
- Вихідна напруга: 5 В
- Вихідний струм: 950 мА

					<i>ДК62.469515.001 ПЗ</i>	Арк.
Змн.	Арк.	№ докум.	Підпис	Дата		23

- Струм споживання: 5 мА
- Робоча температура: -40...+150 °С

Далі обираємо кнопку. Кнопка використовується лише для скидання даних, тому достатньо обирати тип кнопки DPST. Оскільки кнопка планується мати вивід до бокової сторони корпусу, тому було обрано РВ4000АQХ [26] (рис.2.2.3) і має наступні параметри:

- Тип контакту: DPST
- Робоча температура: -5...+60 °С



Рис. 2.8 — вигляд кнопки РВ4000АQХ

При виборі резистору слід враховувати наступні параметри: номінали, номінальна напруга та потужність. Резистори мають наступні номінали: 4,7 кОм, 10 кОм, 1,2 МОм. Оскільки струми в схемі не є достатньо великими, тому достатньо мати потужності резистора 0,125 Вт. Максимальною робочою напругою в схемі маєтсья 15 В. З даних умов і необхідних номіналів обираємо резистори фірми Yageo з наступними параметрами:

- Корпус: 0805
- Номінали: 4,7 кОм, 10 кОм, 1,2 МОм
- Потужність: 0,125 Вт
- Точність: 5%
- Робоча температура: -55...+155 °С

Конденсатори можуть мати невелику точність, але оскільки вони мають невелику ціну, тому обираємо точність 10%. Робочу напругу вибираємо із запасом, яка дорівнює 25 В, температурний коефіцієнт Х5R. З даними умовами обираємо керамічні конденсатори фірми Yageo з наступними параметрами:

					<i>ДК62.469515.001 ПЗ</i>	Арк. 24
Змн.	Арк.	№ докум.	Підпис	Дата		

- Корпус: 0805
- Номінали: 15 пФ, 20 пФ, 33 нФ, 100 нФ, 2,2 мкФ, 4,7 мкФ, 10 мкФ
- Напруга: 25В
- Точність: 10%
- Робоча температура: -55...+125 °С

2.4. Розробка та опис схеми електричної принципової

Схема електрична принципова створюється відповідно до схеми електричної структурної. Дана схема на початку має 2 роз'єми:

- Роз'єм для живлення пристрою
- Роз'єм для передачі даних через USB інтерфейс

Роз'ємом для живлення пристрою має 5 сигналів:

- *USB_Vcc* та *USB_GND*: дані контакти використовуються у якості живлення всієї схеми.
- *USB_Data+* та *USB_Data-*: дані контакти зазвичай використовуються у якості передачі даних. Оскільки даний роз'єм використовується лише з ціллю живлення схеми, тому дані сигнали або не використовуються або необхідно закоротити один з одним.
- *USB_ID*: для USB мікро В даний контакт не використовується.

Підключення такого роз'єму зображено нижче на Рис.2.9:

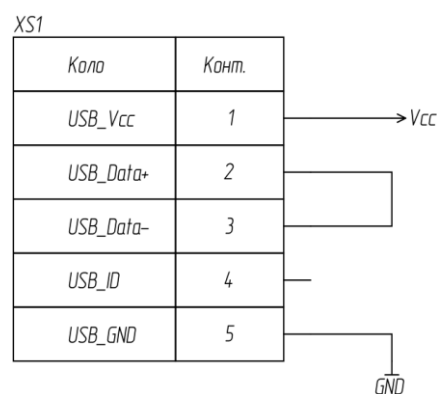


Рис.2.9 — Схема підключення роз'єму USB мікро В

Аналогічні сигнали має роз'єм для підключення клавіатури, для якого необхідно подати +5 В на контакт *USB_Vcc* та заземлити *USB_GND*. Контакти *USB_Data+* та *USB_Data-* використовуються у якості прийняття

					ДК62.469515.001 ПЗ	Арк. 25
Змн.	Арк.	№ докум.	Підпис	Дата		

даних з клавіатури за допомогою USB інтерфейсу. Підключення такого роз'єму зображено нижче на Рис.2.10:

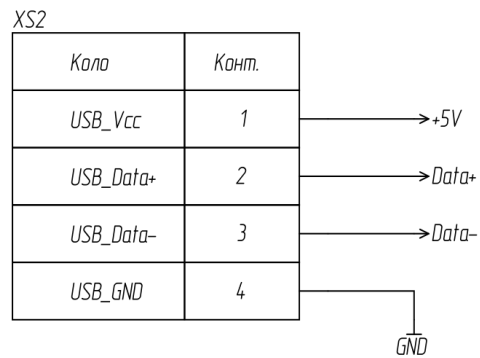


Рис.2.10 — Схема підключення роз'єму USB A

Оскільки мікроконтролер приймає дані використовуючи UART інтерфейс, тому в схемі маєтся перетворювач інтерфейсів, підключення якого зображено нижче на Рис.2.11 [22]:

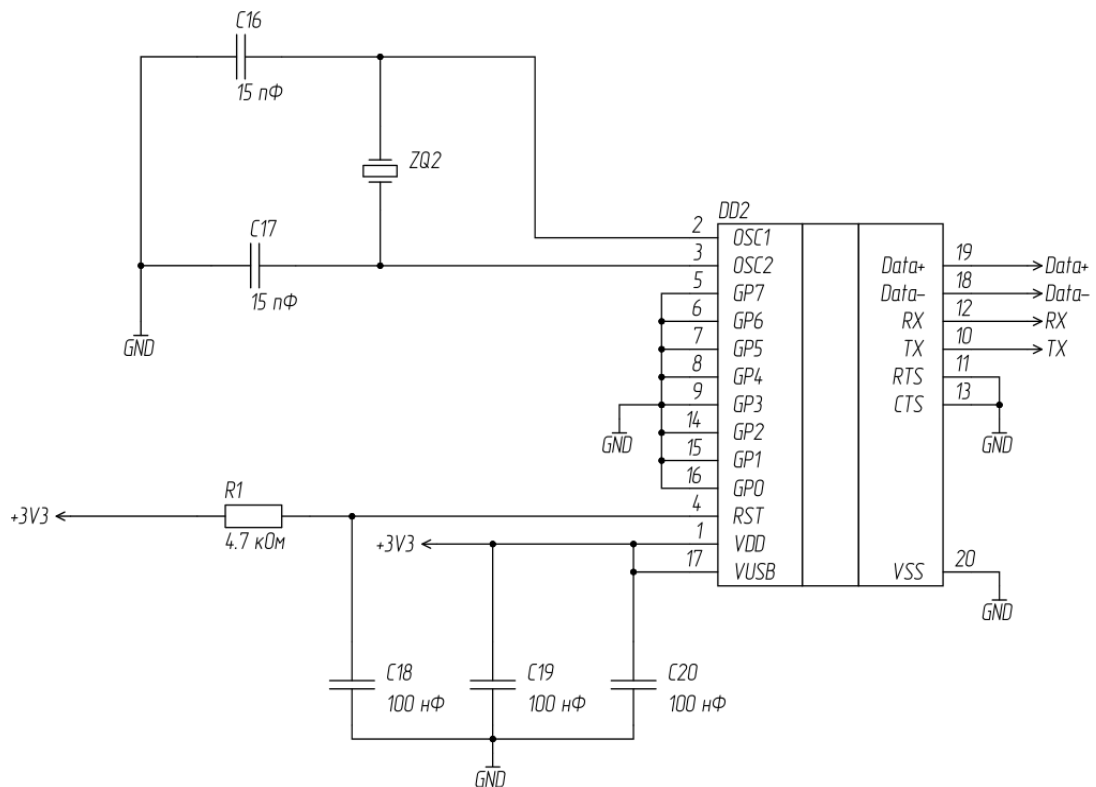


Рис.2.11 — Схема підключення перетворювача інтерфейсів

В даній IC необхідно підключити наступні контакти:

- *OSC1* та *OSC2* підключити до кварцевого резонатора з частотою 12 МГц через 2 конденсатори, номінали яких рекомендовано розробниками самого кварцевого резонатора [22].

- *Data+* та *Data-* підключити до контактів даних роз'єму USB A
- *RX* та *TX* підключити до мікроконтролера, які використовуються для роботи з інтерфейсом USB
- *VDD* та *VUSB* підключаємо до живлення 3,3 В. Дані контакти використовуються для живлення самої ІС.
- *RST* також підключаємо до живлення 3,3 В через резистор для обмеження струму. Виробниками рекомендовано поставити резистор від 1 до 10 кОм [22].

Також поруч з контактами *VDD*, *VUSB* та *RST* поставлено конденсатори, які мають ємність 100 нФ. Це необхідно для того, щоб коли відбуваються просадки напруги, ІС продовжувала працювати.

Маючи вхідну напругу, яка згідно технічного завдання може дорівнювати 3,3 або 5 В \pm 20% перетворимо її за допомогою перетворювача напруги LD1117S50CTR до 5 В, підключення якого зображено на Рис.2.12. Дана перетворена напруга використовується для живлення клавіатури через роз'єм USB A. Номінали конденсаторів були обрані 100 нФ на вході та 10 мкФ на виході [20].

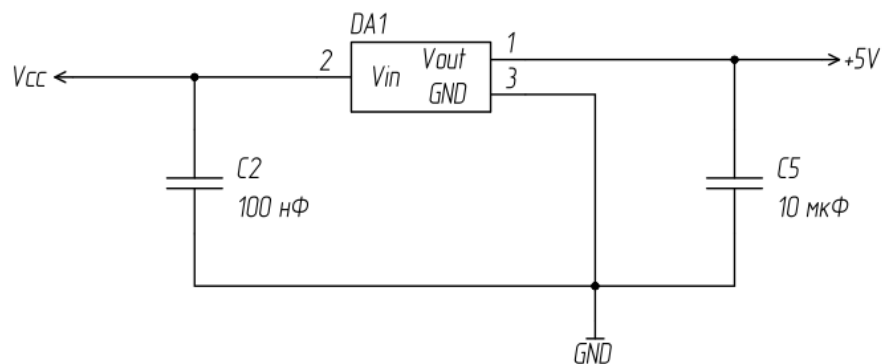


Рис.2.12 — Схема перетворення вхідної напруги у 5 В

Далі для живлення мікроконтролера необхідно мати напругу 3,3 В. Тому для цього перетворюємо напругу 5 В у 3,3 В за допомогою LP3981ILD, підключення якого зображено нижче на Рис.2.13:

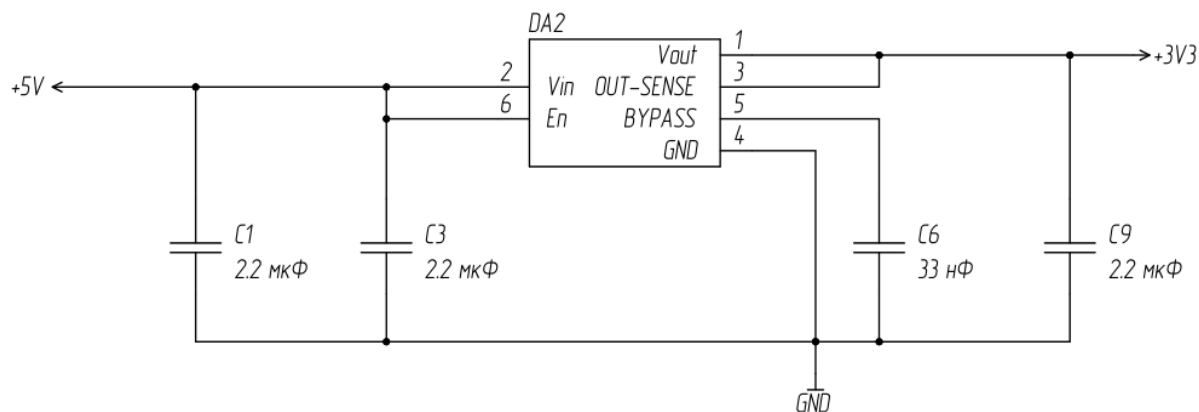


Рис.2.13 — Схема перетворення напруги 5 В у 3,3 В

Для живлення OLED-дисплея необхідно мати напругу 15 В. Тому для такої задачі знову використовуємо перетворювач напруги TL760150BZR підключення якого зображено на Рис. 2.14:

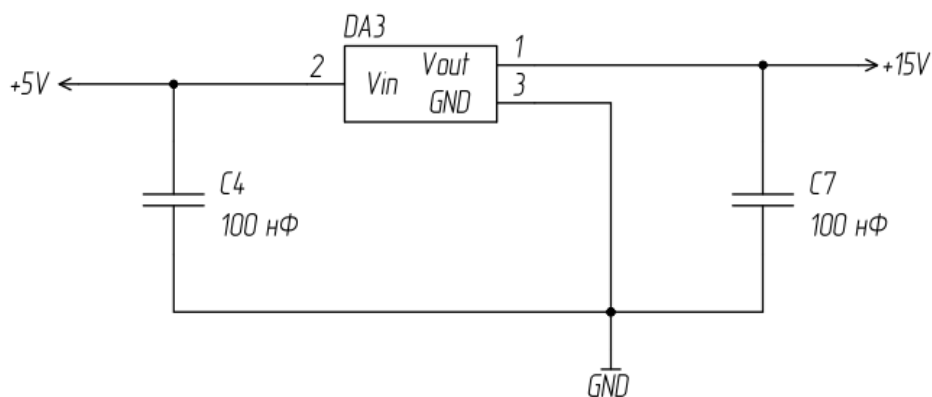


Рис. 2.14 — Схема перетворення напруги 5 В у 15 В

Кожен підключений перетворювач напруги має на кожному вході та виході свій конденсатор. Ємність цих конденсаторів визначені виробниками самих перетворювачів напруг.

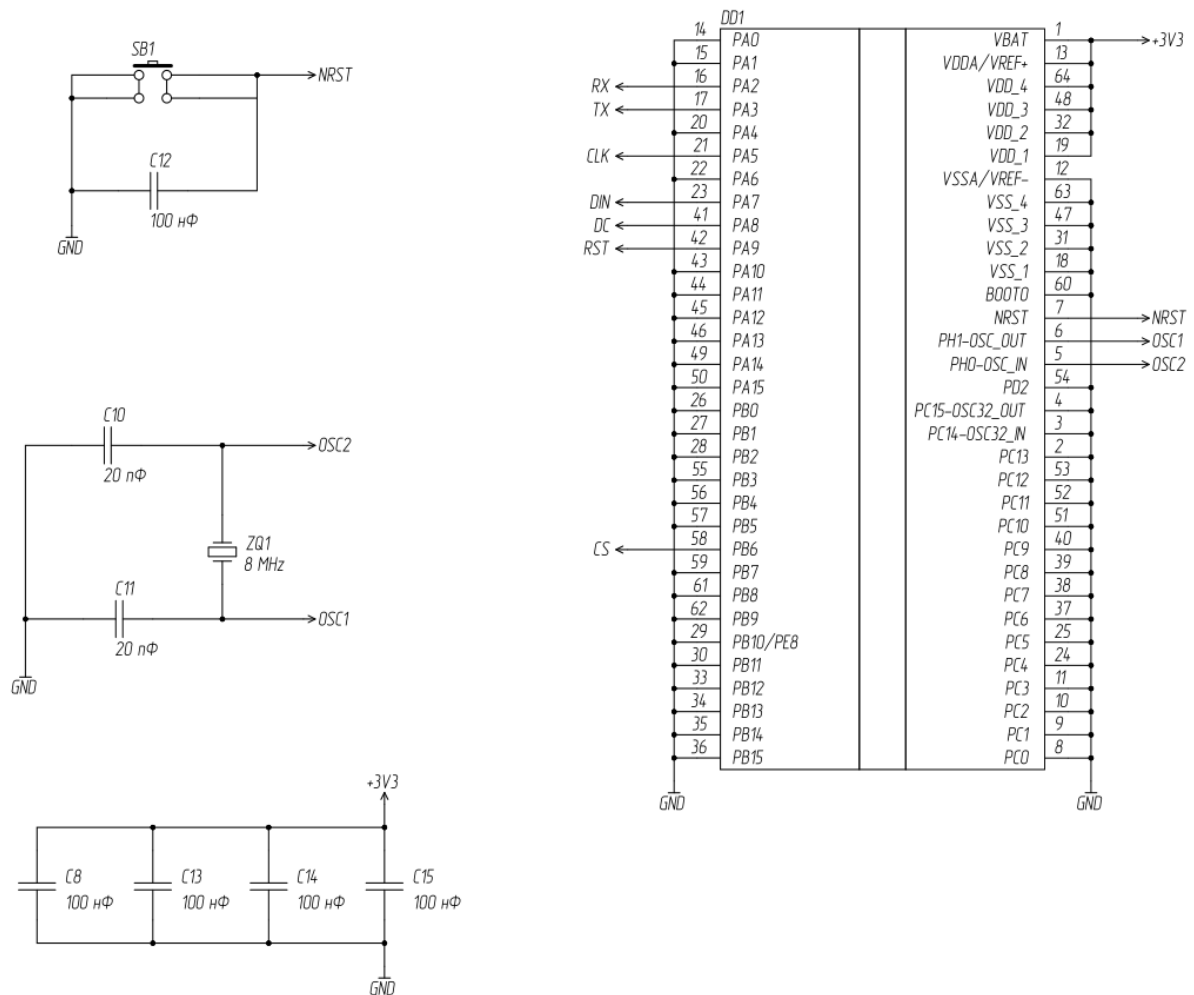


Рис.2.15 — Схема підключення мікроконтролера

Після отримання сигналів *RX* та *TX* у перетворювача інтерфейсів підключаємо їх до мікроконтролера до *PA2* та *PA3* для роботи з UART інтерфейсом. Для повного підключення мікроконтролера також необхідно підключити кварцевий резонатор з частотою 8 МГц до контактів *PH1* та *PH2* (Рис.2.15). Для перезапуску мікроконтролера використовується кнопка *SB1*, до якого підключено паралельно конденсатор. Це необхідно для того, щоб прибрати брязкіт контактів. Також мається 4 конденсатори, які розташовані біля входів живлення для захисту від просадків напруги.

Далі підключається OLED-дисплей, як показано на Рис.2.16. Контакти *VCC*, *VCOMH*, *VCI*, *VDD* проходять через 4 конденсатори, номінали яких були рекомендовані виробниками для підтримки живлення [14]. Аналогічно виконується для контакту *IREF*, до якого підключено резистор. Даний резистор розраховується наступним чином:

$$R = \frac{U_{Iref} - V_{cc}}{I_{ref}} = \frac{15 - 3}{10 \cdot 10^{-6}} = 1,2 \text{ (МОм)}$$

Контакти *CS*, *RST*, *DC*, *CLK*, та *DIN* використовуються для роботи з контролером і проходять через резистори номіналом 10 кОм [14] для обмеження струму. Інші контакти закорочені, оскільки вони не використовуються

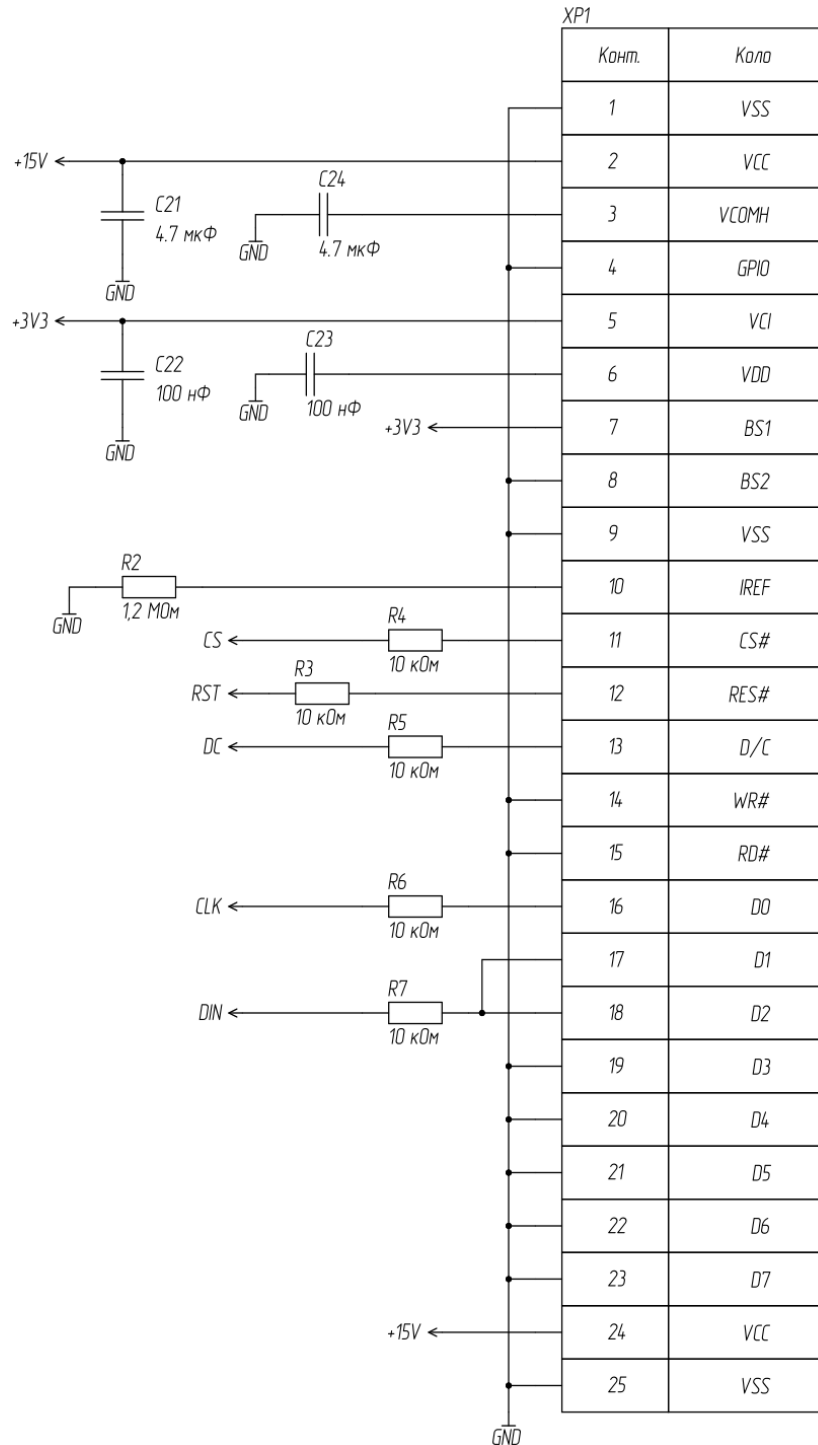


Рис.2.16 — Схема підключення OLED-дисплея

Висновок:

У даному розділі було розроблено схему електричну принципову генератора QR-кода. Дана схема містить 2 роз'єми: перший використовується для живлення схеми, другий для підключення клавіатури у якості вхідних даних. Для обробки даних було обрано мікроконтролер STM32F411RET6. Оскільки вхідні дані приймаються з клавіатури, маючий інтерфейс USB та мікроконтролер має здатність отримувати дані за допомогою UART інтерфейсу, тому було обрано перетворювач інтерфейсів MCP2200, який перетворює USB в UART інтерфейс. Для виведення результату було обрано монохромний OLED-дисплей, розмірністю в 1,5 дюймів. Оскільки схема містить 3 живлення (+3 В, +5 В, +15 В), тому до даної схеми було додано також 3 перетворювачі напруги. Для перезапуску та скидання даних також було обрано кнопку PB400. Згідно зі структурною схемою та елементною базою розроблено схему електричну принципову. Під час вибору номіналів деякі компоненти, що слугують в якості обв'язки мікросхем вибиралися відповідно до рекомендації наведених в документації на обрані компоненти. Також було проаналізовано алгоритм генерації QR-коду.

					<i>ДК62.469515.001 ПЗ</i>	<i>Арк.</i>
<i>Змн.</i>	<i>Арк.</i>	<i>№ докум.</i>	<i>Підпис</i>	<i>Дата</i>		<i>31</i>

Розділ 3. Проектування друкованого вузлу

3.1. Вибір і обґрунтування типу друкованої плати

Друкована плата (ДП) — пластина, виконана з діелектрика (склотекстоліт, текстоліт, гетинакс, ситал тощо), на якій або/і всередині якої сформований хоча б один шар з провідними доріжками. На друковану плату монтується електронні компоненти, які з'єднуються своїми виводами з елементами провідного рисунка паянням, або, значно рідше, зварюванням, у результаті чого складається електронний модуль — змонтована друкована плата [28]. ДП по конструкції розділяються на наступні класи: одношарові (ОДП), двошарові (ДДП), багатошарові (БДП), гнучкі (ГДП).

Оскільки дана схема містить 3 різних живлення (+3, +5, +15 В) і землю, тому можна обрати ДДП або БДП. Оскільки розведення схеми дозволяє верхній шар ДДП розділити на 3 частини ніяк не вплинувши на загальну довжину провідників, тому тут було обрано ДДП.

ДДП мають провідний малюнок на двох сторонах основ. Вони мають високу щільність монтажу, підвищену надійність з'єднань, високу механічну міцність їх кріплень. Також дозволяє полегшати трасування провідників та оптимізувати розміри плати завдяки щільному розміщенні елементів. Дані ДП дають можливість виконати монтаж штирьових елементів з однієї сторони ДП, SMD компонентів — із двох сторін. Тому ДДП є найбільш розповсюдженим класом конструкції.

3.2. Вибір і обґрунтування матеріалу друкованої плати

При виробництві ДП використовуються вітчизняні та імпорتنі матеріали різних виробників. Під час обирання матеріалу для виготовлення основи ДП необхідно враховувати, щоб даний матеріал мав високі електроізоляційні властивості, достатню механічну міцність, був стійким до кліматичних впливів. Цими вимогами задовольняє електротехнічні матеріали: гетинакс (ГОСТ 2718-74), текстоліт (ГОСТ 2910-74) і

					<i>ДК62.469515.001 ПЗ</i>	<i>Арк.</i>
<i>Змн.</i>	<i>Арк.</i>	<i>№ докум.</i>	<i>Підпис</i>	<i>Дата</i>		32

склотекстоліт (ГОСТ 12652-74) та інші, властивості яких занесено до

Таблиця 3.1:

Таблиця 3.1 — Основні властивості матеріалів друкованої плати

Властивості	Гетинакс	Текстоліт	Склотекстоліт
Густина, г/см ³	1,3...1,4	1,3...1,45	1,6...1,75
Теплостійкість по Мартенсу °С, не менше	150	125	180
Холодостійкість, °С	-60	-60	-60
Границя міцності, МПа: при розтягу на згин	80...100 130...150	50...65 90...120	120...180 200
Питома ударна в'язкість, кДж/м ²	15...20	20...27	75
Питомий об'ємний опір, Ом*м	10 ⁸ ...10 ¹¹	10 ⁶ ...10 ⁸	10 ¹¹
Діелектрична проникність	6...7	8	4...7
Тангенс кута діелектричних втрат при 10 ⁶ Гц	0,035...0,08	0,07	0,02
Електрична міцність (перпендикулярно шарам), МВ/м, не менше	33	4...8	18

Оскільки склотекстоліт має кращі характеристики, ніж у гетинаксу та в текстоліту, тому в якості матеріалу для основи ДП було обрано склотекстоліт. Він складається з 8 шарів склотекстоліту, які в сумі дають 1,5 мм товщини. Тому, для реалізації ДДП було обрано FR402-35-1.5, основа якого складається із склотекстоліту і має товщинну 1,5 мм, покритий з двох сторін мідною електролітичною фольгою товщиною 35 мкм.

3.3. Вибір і обґрунтування класу точності друкованої плати

Точність ДП залежить від комплексу технологічних параметрів і з практичної точки зору визначає основні параметри елементів ДП. В першу чергу це відноситься до мінімальної ширини провідників, мінімального зазору між елементами провідного малюнка і до ряду інших параметрів.

					<i>ДК62.469515.001 ПЗ</i>	Арк.
Змн.	Арк.	№ докум.	Підпис	Дата		33

Згідно ГОСТ 23751-86 передбачає п'ять класів точності ДП. Вибір класу точності завжди пов'язаний з конкретним виробництвом. Спроба вирішити цю задачу у зворотньому порядку може призвести до того, що проект не буде реалізований. Необхідні граничні значення елементів друкованого монтажу і допустимі похибки наведені в Таблиця 3.2 та Таблиця 3.3 відповідно:

Таблиця 3.2 — Граничні значення основних параметрів ПМ

Параметер	Клас точності				
	1	2	3	4	5
Ширина друкованого провідника $b_{пр}$, мм	0.75	0.45	0.25	0.15	0.10
Відстань S між краями сусідніх елементів провідного рисунку, мм	0.75	0.45	0.25	0.15	0.10
Гарантований поясок $b_{по}$, мм	0.30	0.20	0.10	0.05	0.03
Відношення $f_{дт}$ номінального діаметру найменшого з металізованих отворів до товщини друкованої плати, мм	0.40	0.40	0.33	0.25	0.20
Щільність монтажу	мала	середня	середня	висока	висока

Таблиця 3.3 — Допустимі похибки виконання елементів ПМ

Похибка	Обозначення	Максимальне значення, мм
Зміщення провідників відносно ліній КС	$\delta_{сп}$	0,05
Розташування отворів (всіх) відносно вузлу КС	δ_o	0,07
Розташування КМ відносно вузлу КС	$\delta_{км}$	0.015(0.05)
Фотокопії та фотошаблону	$\delta_{фф}$	0,06
Розташування КМ відносно вузлу КС на фотошаблоні	$\delta_{фш}$	0,05

Виготовлення ДП п'ятого класу точності вимагає застосування унікального високоточного обладнання, спеціальних (як правило, дорогих) матеріалів і навіть створення у виробничих приміщеннях "чистої зони". Таким вимогам відповідає не кожне виробництво. Однак, ДП невеликого розміру можуть виконуватися по п'ятому класу на обладнанні, що забезпечує виробництво плат четвертого класу.

ДП четвертого класу випускається на високоточному обладнанні, але вимога до матеріалів, обладнання і виробничих приміщень нижче, ніж для п'ятого класу.

ДП третього класу — найбільш поширені, оскільки, з одного боку, забезпечують досить високу щільність трасування і монтажу, а з іншого — для їх виробництва достатньо звичайного спеціалізованого обладнання.

Випуск ДП другого і третього класів здійснюється на звичайному не спеціалізованому обладнанні. Такі ДП, з невисокими конструктивними параметрами, призначені для недорогих пристроїв з малою щільністю монтажу.

Якщо обрати 2-ий клас точності, то ширина провідників не дозволить виконати з'єднання для більшості SMD компонентів. Якщо обрати 3-ій клас точності, то цієї точності достатньо, щоб виконати всі з'єднання між SMD компонентами, тому обирати 4-ий клас точності вже немає сенсу, адже даний вибір лише підвищить складність розробки ДП. Отже, для даної ДП рекомендовано обрати 3-ій клас точності.

3.4. Вибір і обґрунтування методу виготовлення друкованої плати

Класифікація методів виготовлення ДП, що використовуються в наш час на виробництві можна представити наступним чином (Рис.3.1):

					<i>ДК62.469515.001 ПЗ</i>	<i>Арк.</i>
<i>Змн.</i>	<i>Арк.</i>	<i>№ докум.</i>	<i>Підпис</i>	<i>Дата</i>		35

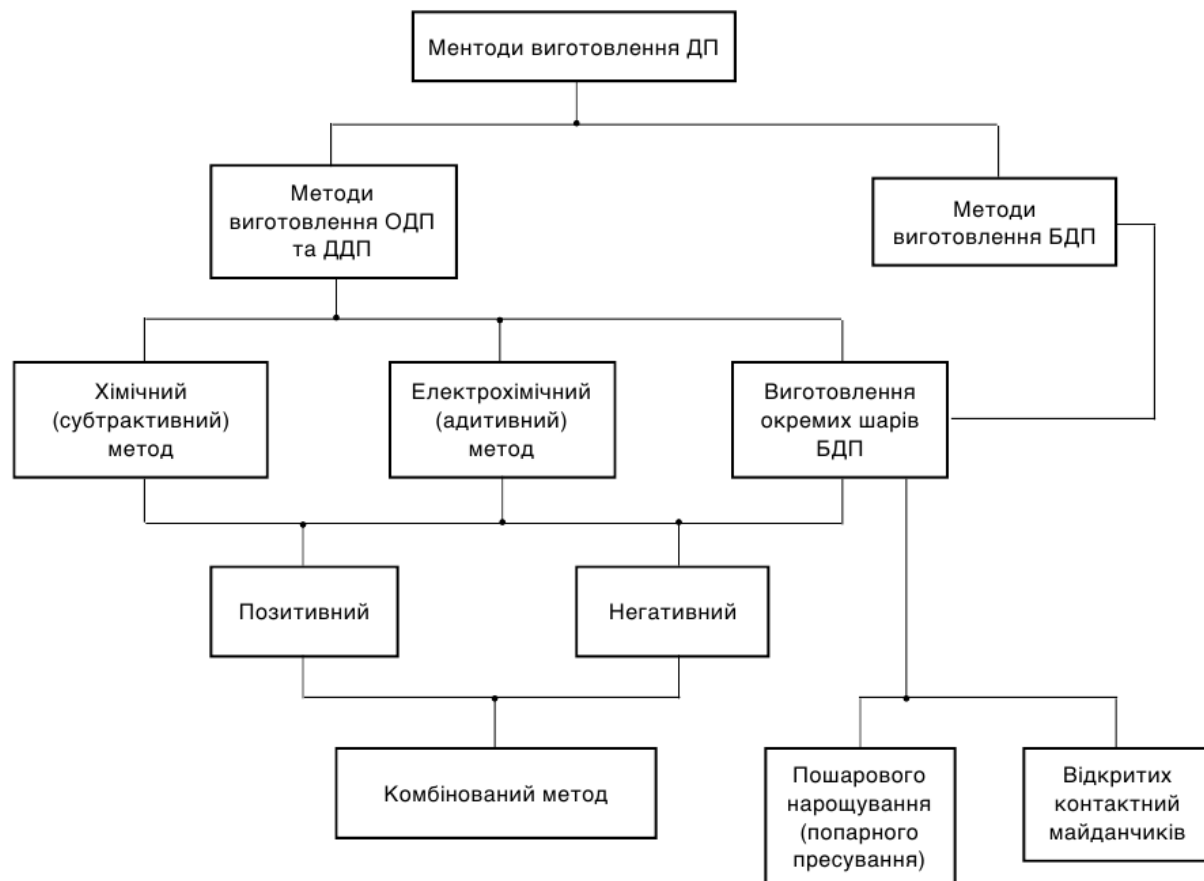


Рис.3.1 — Методи виготовлення ДП

Слід відмітити, що всі методи виготовлення ДП включають набір приблизно однотипних технологічних операцій, основними з яких є:

- Нанесення захисного покриття(резисту)
- Отримання друкованого малюнку елементів монтажу
- Виготовлення МО та ПО, КМ для поверхневого монтажу (для SMD компонентів)
- Пресування або склеювання друкованих шарів(для багатошарової структури)
- Операційний контроль

Для виготовлення друкованого вузлу був обраний комбінований позитивний метод, оскільки він має ряд переваг перед іншими методами:

- Можливість відтворювати всіх типів друкованих елементів з високим ступенем дозволу.

- Захищеність фольгою ізоляції від технологічних розчинів — гарна надійність ізоляції.
- Гарна міцність зчеплення (адгезія) металічних елементів плати з діелектричною основою.

3.5. Розміщення КЕ в монтажному просторі

Розміщення КЕ в монтажному просторі необхідно для того, щоб зменшити довжину зв'язків. Тому, для вирішення даної задачі маємо схему електричну принципову, яка буде описана матрицею зв'язків.

Для створення матриці зв'язків спершу потрібно спростити схему прибравши компоненти та їх зв'язки. При прибиранні компонентів слід враховувати їх габаритні розміри та їх власне нагрівання, щоб компонент мав власний простір для розсіювання своєї температури і також для рівномірного виділенні температури, щоб не було прогинів у корпусі.

За рахунок того, що висота OLED-дисплея являється значно меншою за висоту деяких інших компонентів і дана ДП, тому було КЕ ХР1 було розміщено на зворотній частині ДП. Серед компонентів залишаємо всі мікросхеми, роз'єми, кварцеві резонатори та кнопку. Також прибираємо всі зв'язки живлення та землі. В результаті такого спрощення мається спрощена принципова схема, яка зображена на Рис.3.2.

					<i>ДК62.469515.001 ПЗ</i>	Арк.
Змн.	Арк.	№ докум.	Підпис	Дата		37

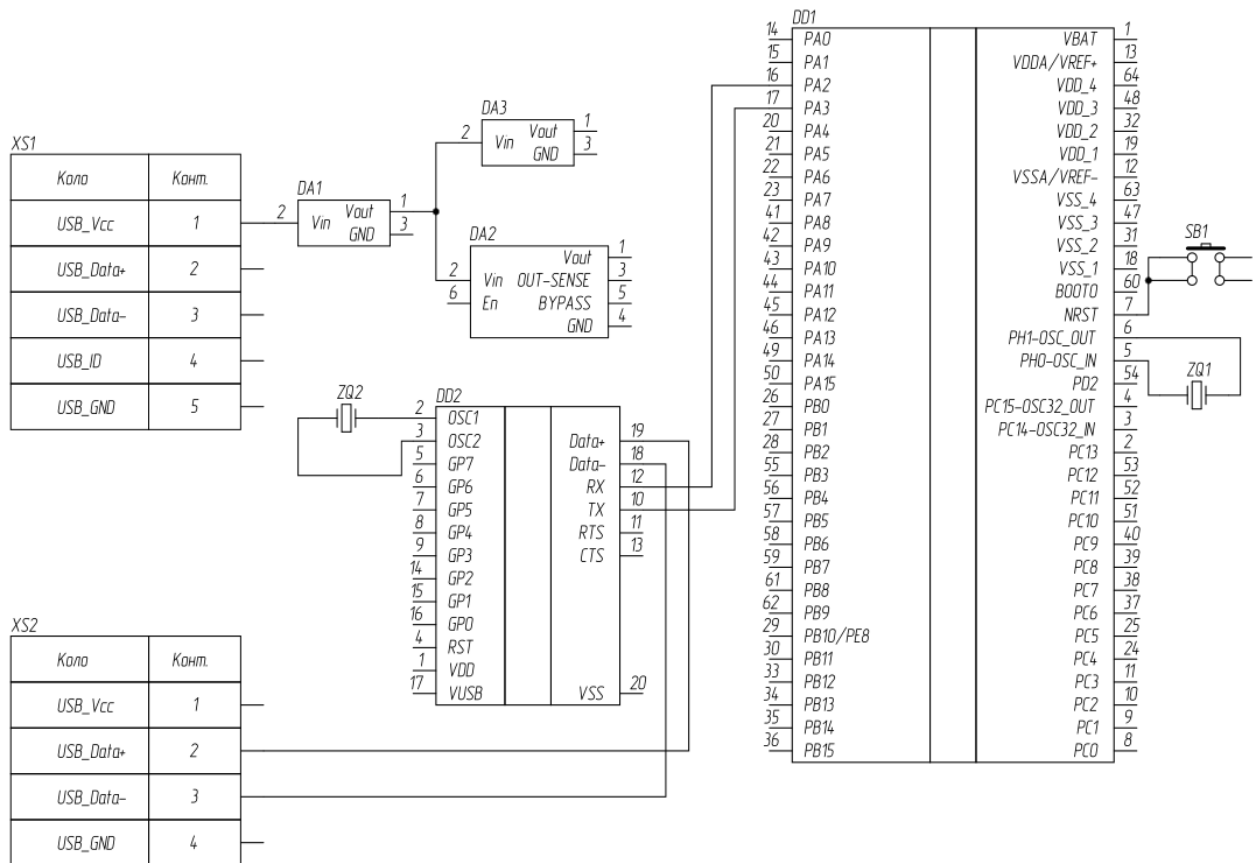


Рис.3.2 — Спрощена схема електрична принципова

Після даного спрощення можна побудувати матрицю зв'язків. Дана схема містить 10 компонентів і має наступну множину зв'язків:

$$X = \{DD1, DD2, XS1, XS2, DA1, DA2, DA3, ZQ1, ZQ2, SB1\}$$

Для розташування даних компонентів необхідно мати 10 посадкових місць, але оскільки ДП було мати вигляд більш квадратного розміру, тому для даної ДП буде виділено 12 посадкових місць, розташування яких зображено на Рис.3.3.

Під час побудови розміщення зв'язків, критеріями побудови будуть враховуватись:

- 1) Мінімальна сумарна довжина зв'язків всіх з'єднань.
- 2) Максимальне близьке розміщення зв'язків, що мають найбільше число зв'язків.
- 3) Максимальне число провідників простої конфігурації і т.д..

P1	P2	P3
P4	P5	P6
P7	P8	P9
P10	P11	P12

Рис.3.3 — Посадкові місця

Для розташування КЕ мається таблиця матриці зв'язків (Таблиця 3.4). При розташуванні необхідно врахувати, що XS1, XS2 та SB1 повинні бути розташовані на краю плати і також з конструктивної точки зору КЕ XS1, XS2 та SB1 для зручності повинні бути з однієї сторони. Також за рахунок того, що мається 3 перетворювачі напруги (DA1-DA3), їх потрібно розташувати максимально близько до роз'єму XS1.

Таблиця 3.4 — Матриця зв'язків

	DD1	DD2	XS1	XS2	DA1	DA2	DA3	ZQ1	ZQ2	SB1	p
DD1	0	2	0	0	0	0	0	2	0	1	5
DD2	2	0	0	2	0	0	0	0	2	0	6
XS1	0	0	0	0	1	0	0	0	0	0	1
XS2	0	2	0	0	0	0	0	0	0	0	2
DA1	0	0	1	0	0	1	1	0	0	0	3
DA2	0	0	0	0	1	0	1	0	0	0	2
DA3	0	0	0	0	1	1	0	0	0	0	2
ZQ1	2	0	0	0	0	0	0	0	0	0	2
ZQ2	0	2	0	0	0	0	0	0	0	0	2
SB1	1	0	0	0	0	0	0	0	0	0	1

Для початку розташуємо КЕ XS1 на краю плати по центру на ПМ P11, оскільки поруч з ним потрібно розташувати DA1-DA3. Наступними компонентами розташовуємо XS2 та SB1 на ПМ P12 та відповідно. Далі поруч розташовуємо DA1 на ПМ P8, DA2 на P7 та DA3 на P8. Найбільшу

кількість зв'язків мають компоненти DD1 та DD2, тому їх розташовуємо посередині плати на ПМ P5 та P6 відповідно. Далі розташовуємо ZQ1, оскільки він має зв'язки з DD1, тому його розміщуємо в ПМ P4. Наступним по кількості зв'язків розташовуємо KE ZQ2, який розташовуємо поруч з DD2, тобто в ПМ P3. Останній компонент SB1 розташовуємо в ПМ DD1. В результаті такого розташування маємо наступне розташування, яке зображено на рис.3.1.4:

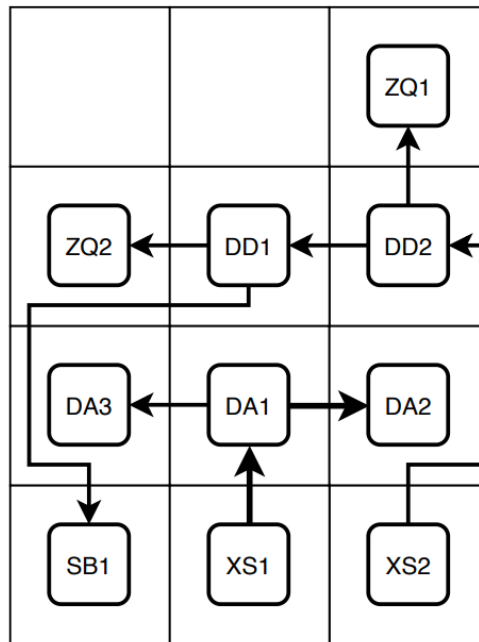


Рис.3.4 — Розміщення компонентів на ПМ

3.6. Проектування друкованої плати у середовищі Altium Designer

Altium Designer — це комплексна система автоматизованого проектування радіоелектронних засобів. Дозволяє проектувати друковані плати у тривимірному вигляді, а також обмінюватись інформацією з механічними системами автоматизованого проектування [28].

Під час виконання дипломного проекту було розглянуто наступні можливості програми:

- самостійне створення бібліотеки компонентів (додавання УГП і посадкового місця);
- створення принципової електричної схеми;
- трасування друкованої плати (ручне або автоматичне трасування).

Приклад готової друкованої плати наведено на Рис.3.5 та Рис.3.6:

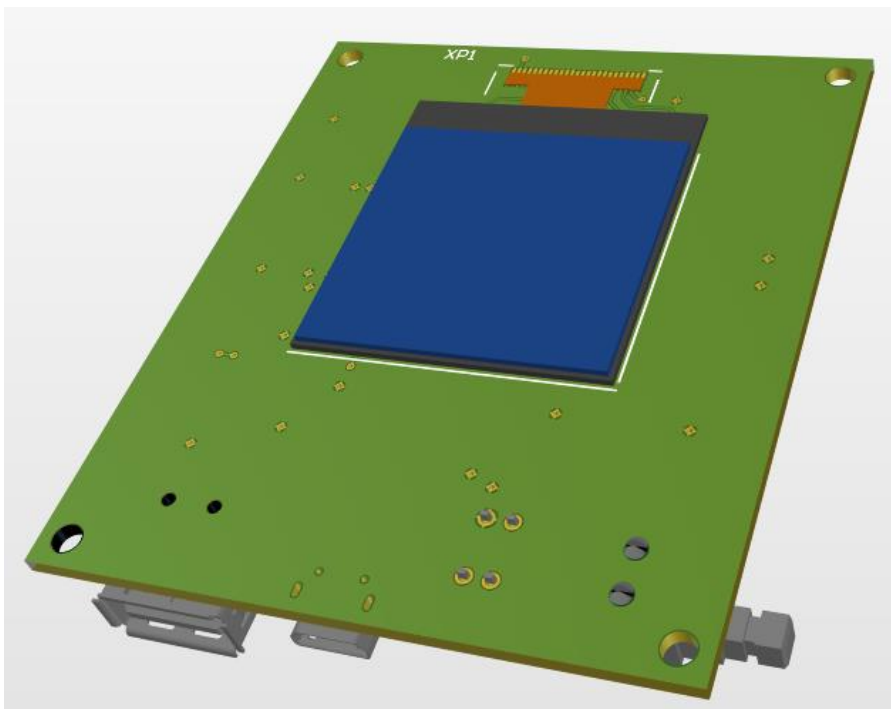


Рис.3.5 — 3-D модель ДП. Фронтальний вид

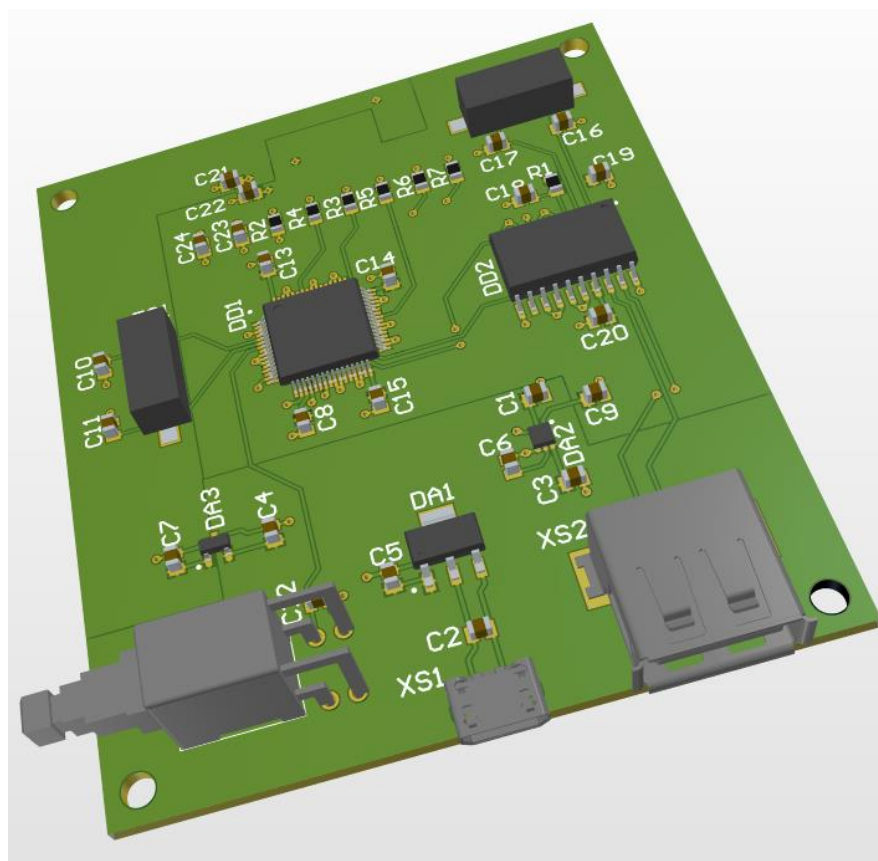


Рис.3.6 — 3-D модель ДП. Задній вид

Під час проектування ДП в середовищі AD було враховано:

- Обраний тип ДП, який вплинув на кількість шарів ДП;

					<i>ДК62.469515.001 ПЗ</i>	Арк.
Змн.	Арк.	№ докум.	Підпис	Дата		41

- Обраний матеріал, який вплинув на товщину ДП;
- Обрану точність, яка вплинула на товщину провідників;
- Розміщення КЕ в монтажному просторі.

Висновок:

У даному розділі було: розглянуто та обрано тип друкованої плати, обрано метод виготовлення ДП, обрано клас точності монтажу, розміщено КЕ в монтажному просторі та спроектовано ДП у середовищі AD. Проаналізувавши ТЗ і схему електричну принципову обрано ДДП для реалізації схеми, завдяки чому плата має невеликі габаритні розміри. Матеріалом ДП було обрано склотекстоліт FR402-35-1,5 через свої переваги над іншими матеріалами. Для конструювання ДП обрано 3-ій клас точності, так як даний клас точності може забезпечити виконати всі з'єднання між КЕ. Також було розглянуто метод виготовлення ДП, було вирішено використовувати комбінований позитивний метод, оскільки даний метод широко використовується і має ряд своїх переваг перед іншими методами. Було виконано розміщення КЕ в монтажному просторі, під час чого було обрано найбільш оптимальне розміщення. В кінці було зпроектовано ДП в середовищі AD, під час чого було створено бібліотеки компонентів, принципову схему електричну та виконано трасування ДП.

					<i>ДК62.469515.001 ПЗ</i>	Арк.
Змн.	Арк.	№ докум.	Підпис	Дата		42

Розділ 4. Розрахунки, що підтверджують правильність конструкторського рішення

4.1. Конструкторсько-технологічний розрахунок друкованої плати

4.1.1. Розрахунок мінімальної ширини провідника по постійному струмі для ланцюгів живлення та землі

Мінімальна ширина друкованого провідника по постійному струму $b_{\min I}$ (мм) для ланцюгів живлення та «землі» визначається за виразом (4.1):

$$b_{\min I} = \frac{I_{\max}}{j_{\text{доп}} \cdot t_{\text{пров}}}, \quad (4.1)$$

де I_{\max} — максимально можливий струм в ланцюгу;

$j_{\text{доп}}$ — допустима щільність струму для ДП, що виготовлені комбінованим позитивним методом, $j_{\text{доп}} = 48 \frac{\text{А}}{\text{мм}^2}$ (за ОСТ 4.010.022-85);

$t_{\text{пров}}$ — товщина друкованого провідника і визначається за виразом (4.2).

Друкований провідник виготовлюється комбінованим позитивним методом, тому:

$$t_{\text{пров}} = h_{\text{ф}} + h_{\text{ГМ}} + h_{\text{ХМ}}, \quad (4.2)$$

де $h_{\text{ф}}$ — товщина фольги, $h_{\text{ф}} = 0,035$ мм;

$h_{\text{ГМ}}$ — товщина шару гальванічно осадженої міді, $h_{\text{ГМ}} = 0,055$ мм;

$h_{\text{ХМ}}$ — товщина шару хімічно осадженої міді, $h_{\text{ХМ}} = 0,0065$ мм.

$$t_{\text{пров}} = 0,035 + 0,055 + 0,0065 = 0,097 \text{ (мм)}$$

Параметр I_{\max} у виразі (4.1) визначається як сума струмів, що споживають всі активні елементи схеми. Значення струмів, які споживають елементи схеми, наведені у Таблиця 4.1:

Таблиця 4.1 — Струми, які споживаються елементи схеми

КЕ	$I_{\text{спож}}, \text{мА}$
STM32F411RET6	240
MCP2200T-I/SO	13

Таблиця 4.2. Продовження

LP3981ILD-3.3	0,21
TLV76015DBZR	2
LD1117S50CTR	5
SSD1327	40
UJ2-AH-1-SMT	500

У результаті:

$$I_{max} = 240 + 13 + 0,21 + 2 + 5 + 40 + 500 = 800,21 \text{ (мА)}$$

$$b_{min I} = \frac{800,21 \cdot 10^{-3}}{48 \cdot 0,097} = 0,172 \text{ (мм)}$$

Отримане значення мінімальної ширини провідника $b_{min I} = 0,172$ мм входить в значення обраного 3 класу точності ($b_{пр}^r = 0,25$ мм). Таким чином, обраного класу точності провідників достатньо, щоб проводити постійний струм для ланцюгів живлення та «землі».

4.1.2. Визначення мінімальної ширини провідника з урахуванням допустимого падіння напруги на ньому

Мінімальна ширина провідника з урахуванням допустимого падіння напруги на ньому, визначається за виразом (4.3)

$$b_{min U} = \frac{\rho \cdot I_{max} \cdot l_{пр}}{U_{доп} \cdot t_{пров}}, \quad (4.3)$$

де ρ — питомий опір провідника, для комбіновано позитивного методу

$$\rho = 0,0175 \frac{\text{Ом} \cdot \text{мм}^2}{\text{м}};$$

I_{max} — максимальний струм у провіднику, $I_{max} = 800,21$ мА, визначено вище у пункті 4.1.1;

$l_{пр}$ — довжина найдовшого друкованого провідника ДП, $l_{пр} = 35,881$ мм;

$U_{доп}$ — допустиме падіння напруги на друкованому провіднику, $U_{доп} = 5 \cdot 5\% = 0,25$ В;

					ДК62.469515.001 ПЗ	Арк.
Змн.	Арк.	№ докум.	Підпис	Дата		44

$t_{\text{пров}}$ — товщина друкованого провідника, визначено вище у пункті 4.1.1.

$$b_{\min U} = \frac{0,0175 \cdot 800,21 \cdot 10^{-3} \cdot 35,881 \cdot 10^{-3}}{0,25 \cdot 0,097} = 0,021 \text{ (мм)}$$

Така мінімально допустима ширина провідника цілком задовольняє вибраній ширині провідників для живлення рівній 0,3 мм.

4.1.3. Визначення номінально діаметру монтажного отвору

Номінальний діаметр монтажного отвору визначається за виразом (4.4):

$$d \geq d_{\text{ве}} + \Delta d_{\text{мо}} + r, \quad (4.4)$$

де $d_{\text{ве}}$ — діаметр виводу елемента, для якого визначається діаметр монтажного отвору, $d_{\text{ве}} = 0,5$;

$\Delta d_{\text{мо}}$ — нижнє граничне відхилення від номінального діаметру МО, $\Delta d_{\text{мо}} = 0,1$ мм;

r — різниця між мінімальним діаметром МО та максимальним діаметром виводу елемента $r = 0,1 \dots 0,2$ мм. В цьому випадку виходить якісне заповнення МО при пайці та оптимальна вага самої пайки.

$$d \geq 0,5 + 0,1 + 0,2 = 0,8 \text{ (мм)}$$

4.1.4. Визначення діаметру контактної площадки

Діаметр контактної площадки визначається за виразом (4.5):

$$D_{\min} = D_{\min 1} + 1,5 \cdot h_{\text{ф}} + 0,03, \quad (4.5)$$

де $D_{\min 1}$ — мінімальний ефективний діаметр КМ, визначається за виразом (4.6);

$h_{\text{ф}}$ — товщина фольги, $h_{\text{ф}} = 0,035$ мм. Коефіцієнт 1,5 враховує підтравлювання фольги друкованого провідника в ширину;

0,03 — похибка для КМ виготовлених комбінованим позитивним методом.

$$D_{\min 1} = 2 \cdot \left(b_{\text{по}} + \frac{d_{\text{max}}}{2} + \delta_{\text{o}} + \delta_{\text{км}} \right), \quad (4.6)$$

де $b_{\text{по}}$ — ширина пояска КМ, $b_{\text{по}} = 0,1$ мм, визначається з Таблиця 3.2;

					ДК62.469515.001 ПЗ	Арк.
Змн.	Арк.	№ докум.	Підпис	Дата		45

d_{max} — максимальний діаметр отвору в ДП, визначається за виразом (4.7);

δ_o — похибка розташування центру отвору відносно вузла КС, $\delta_o = 0,07$ мм, визначається з Таблиця 3.2;

$\delta_{км}$ — похибка розташування центру КМ відносно вузлу КС, $\delta_{км} = 0,05$ мм, визначається з Таблиця 3.3.

Максимальний діаметр просвердленого отвору ДП визначається за виразом (4.7):

$$d_{max} = d + \Delta d + (0,1 \dots 0,15), \quad (4.7)$$

де d — номінальний діаметр МО, $d = 0,8$, визначено вище у пункті 4.1.3;

Δd — допуск на діаметр отвору, $\Delta d = 0,05$ мм.

$$d_{max} = 0,8 + 0,05 + 0,1 = 0,95 \text{ (мм)}$$

$$D_{min1} = 2 \cdot \left(0,1 + \frac{0,95}{2} + 0,07 + 0,05 \right) = 1,39 \text{ (мм)}$$

$$D_{min} = 1,39 + 1,5 \cdot 0,035 + 0,03 = 1,47 \text{ (мм)}$$

Максимальний діаметр КМ визначається за виразом (4.8):

$$D_{max} = D_{min} + 0,02 \quad (4.8)$$

$$D_{max} = 1,47 + 0,02 = 1,49 \text{ (мм)}$$

4.1.5. Визначення мінімальної та максимальної ширини провідника

Мінімальна ширина провідника визначається виразом (4.9):

$$b_{min} = b_{пр}^r + 1,5 \cdot h_{\phi} + 0,03, \quad (4.9)$$

де $b_{пр}^r$ — мінімальна ширина провідника, $b_{пр}^r = 0,25$ мм, визначається з таблиці класів точності (Таблиця 3.2);

h_{ϕ} — товщина фольги, $h_{\phi} = 0,035$ мм.

$$b_{min} = 0,25 + 1,5 \cdot 0,035 + 0,03 = 0,33 \text{ (мм)}$$

Максимальна ширина провідника визначається виразом (4.10):

$$b_{max} = b_{min} + 0,02, \quad (4.10)$$

$$b_{max} = 0,33 + 0,02 = 0,35 \text{ (мм)}$$

					ДК62.469515.001 ПЗ	Арк.
Змн.	Арк.	№ докум.	Підпис	Дата		46

4.1.6. Визначення мінімальної відстані між провідником та контактним майданчиком.

На Рис.4.1 зображено схематичне розташування провідника і КМ, зображені відстані які використовуються для розрахунку за формулою (4.11):

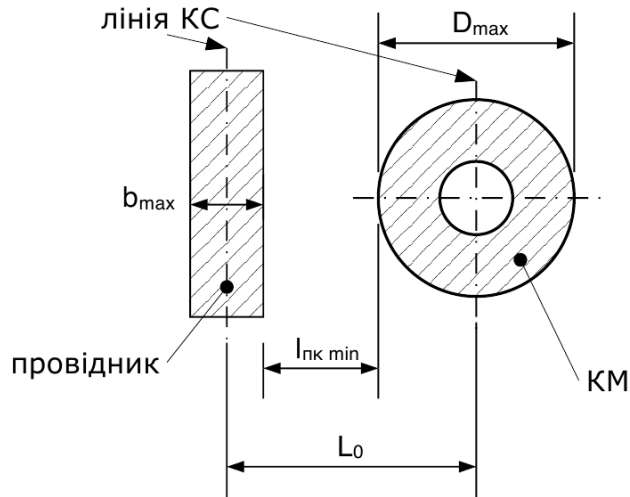


Рис.4.1 — Відстань між провідником та КМ

$$l_{\text{ПКМ min}} = L_0 - \left(\frac{D_{\text{max}}}{2} + \delta_{\text{КМ}} + \frac{b_{\text{max}}}{2} + \delta_{\text{СП}} \right), \quad (4.11)$$

де L_0 — відстань між центрами отворів та друкованим провідником, які кратні кроку КС, $L_0 = 1,25$ мм (обрано найгірший випадок);

D_{max} — максимальний діаметр КМ, $D_{\text{max}} = 1,49$, визначено вище у пункті 4.1.4;

$\delta_{\text{КМ}}$ — похибка розташування центра КМ відносно вузла КС, $\delta_{\text{КМ}} = 0,05$ мм, визначено з Таблиця 3.3;

b_{max} — максимальна ширина провідника, $b_{\text{max}} = 0,35$, визначено вище у пункті 4.1.5;

$\delta_{\text{СП}}$ — похибка, що враховує зміщення провідника, $\delta_{\text{СП}} = 0,05$ мм, визначається з Таблиця 3.3.

$$l_{\text{ПКМ min}} = 1,25 - \left(\frac{1,49}{2} + 0,05 + \frac{0,35}{2} + 0,05 \right) = 0,23 \text{ (мм)}$$

4.1.7. Визначення мінімальної відстані між сусідніми провідниками.

На Рис.4.2 зображено схематичне розташування двох провідників і зображені відстані, які використовуються для розрахунку виразу (4.12):

					ДК62.469515.001 ПЗ	Арк. 47
Змн.	Арк.	№ докум.	Підпис	Дата		

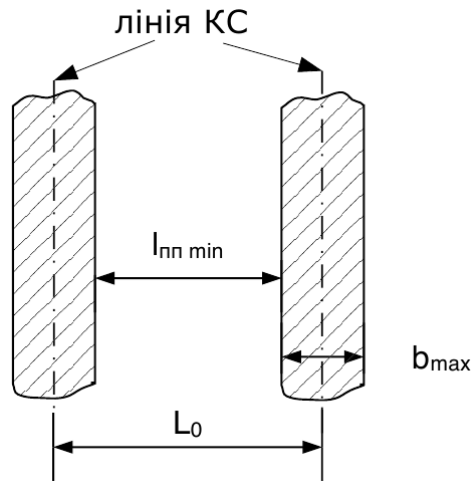


Рис.4.2 — Відстань між двома провідниками

$$l_{пп\ min} = L_0 - (b_{max} + 2 \cdot \delta_{сп}), \quad (4.12)$$

де L_0 — відстань між центрами отворів та друкованим провідником, які кратні кроку КС, $L_0 = 1,25$ мм (обрано найгірший випадок);

b_{max} — максимальна ширина провідника, $b_{max} = 0,35$, визначено вище у пункті 4.1.5;

$\delta_{сп}$ — похибка, що враховує зміщення провідника, $\delta_{сп} = 0,05$ мм, визначається з Таблиця 3.3.

$$l_{пп\ min} = 1,25 - (0,35 + 2 \cdot 0,05) = 0,8 \text{ (мм)}$$

4.1.8. Визначення мінімальної відстані між контактними площадками.

На Рис.4.3 зображено схематичне розташування провідника і КМ, де використовуються відстані для розрахунку виразу (4.13):

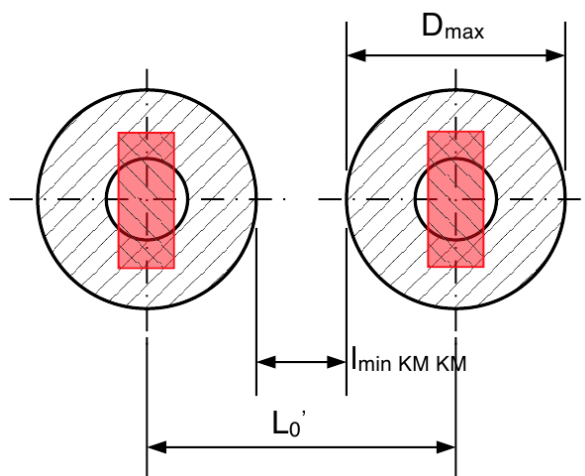


Рис.4.3 — Мінімальна відстань між двома КМ

$$l_{\min \text{ КМ КМ}} = L_{01} - (D_{\max} + 2 \cdot \delta_{\text{КМ}}), \quad (4.13)$$

де L_{01} — відстань між центрами сусідніх КМ, $L_{01} = 2,5$ мм;

D_{\max} — максимальний діаметр КМ, $D_{\max} = 1,49$, визначено вище у пункті 4.1.4;

$\delta_{\text{КМ}}$ — похибка розташування центра КМ відносно вузла КС, $\delta_{\text{КМ}} = 0,05$ мм, визначено з Таблиця 3.3.

$$l_{\min \text{ КМ КМ}} = 2,5 - (1,49 - 2 \cdot 0,05) = 1,21 \text{ (мм)}$$

Отримане значення задовольняє 3-ій клас точності.

4.2. Електричний розрахунок друкованої плати

Розрахунок виконується за умов, що плата виготовлена комбінованим методом, згідно ГОСТ Р 50621-93.

4.2.1. Визначення падіння напруги на найдовшому друкованому провіднику

Падіння напруги на найдовшому друкованому провіднику визначається за виразом (4.14):

$$U_{\text{пад}} = \frac{\rho \cdot I_{\max} \cdot l_{\text{пр}}}{b_{\text{пр}} \cdot t_{\text{пр}}}, \quad (4.14)$$

де ρ — питомий опір провідника, для комбіновано позитивного методу $\rho = 0,0175 \frac{\text{Ом} \cdot \text{мм}^2}{\text{м}}$;

I_{\max} — максимальний струм на найдовшому провіднику, $I_{\max} = 0,33$ мА;

$l_{\text{пр}}$ — довжина найдовшого друкованого провідника ДП, $l_{\text{пр}} = 0,936$ м;

$b_{\text{пр}}^r$ — мінімальна ширина провідника, $b_{\text{пр}}^r = 0,25$ мм, визначається з таблиці класів точності (Таблиця 3.2);

$t_{\text{пр}}$ — товщина друкованого провідника, $t_{\text{пр}} = 0,097$ мм, визначено вище у пункті 4.1.1.

$$U_{\text{пад}} = \frac{0,0175 \cdot 0,33 \cdot 10^{-3} \cdot 0,936}{0,15 \cdot 0,097} = 0,37 \text{ (мВ)}$$

Розраховане падіння напруги не перевищує 5% від напруги живлення.

					ДК62.469515.001 ПЗ	Арк. 49
Змн.	Арк.	№ докум.	Підпис	Дата		

4.2.2. Визначення втрати потужності двосторонньої друкованої плати

Втрати потужності визначається за виразом (4.15):

$$P_{\text{пот}} = 2 \cdot \pi \cdot f \cdot C \cdot U_{\text{жив}}^2 \cdot \text{tg}\sigma, \quad (4.15)$$

де: $f_{\text{нч}} = 1$ Гц для низьких частот та $f_{\text{вч}} = 8$ МГц для високих частот;

C — ємність друкованої плати, яка визначається за виразом (4.16);

$U_{\text{жив}}$ — напруга живлення, $U_{\text{жив}} = 5$ В;

$\text{tg}\sigma$ — тангенс кута діелектричних втрат для матеріалу друкованої плати, $\text{tg}\sigma = 0,002$ для матеріалу FR4.

$$C = \frac{0,009 \cdot \varepsilon \cdot S_m}{h}, \quad (4.16)$$

де ε — діелектрична проникність, $\varepsilon = 4,5$ для FR4;

S_m — площа металізації, $S_m = 8816$ мм²;

h — товщина ДП, $h = 1,5$ мм.

$$C = \frac{0,009 \cdot 4,5 \cdot 8816}{1,5} = 238 \text{ (пФ)}$$

Для низьких частот:

$$\begin{aligned} P_{\text{пот}} &= 2 \cdot \pi \cdot f_{\text{нч}} \cdot C \cdot E_{\text{п}}^2 \cdot \text{tg}\sigma = 2 \cdot 3,14 \cdot 1 \cdot 238 \cdot 10^{-12} \cdot 5^2 \cdot 0,002 \\ &= 74,7 \text{ (пВт)} \end{aligned}$$

Для високих частот:

$$\begin{aligned} P_{\text{пот}} &= 2 \cdot \pi \cdot f_{\text{вч}} \cdot C \cdot E_{\text{п}}^2 \cdot \text{tg}\sigma = 2 \cdot 3,14 \cdot 8 \cdot 10^6 \cdot 238 \cdot 10^{-12} \cdot 5^2 \cdot 0,002 \\ &= 597,8 \text{ (мкВт)} \end{aligned}$$

4.2.3. Визначення ємності між двома сусідніми провідниками, які розташовані на одній стороні друкованої плати та мають однакову ширину

Ємність між двома сусідніми провідниками, які розташовані на одній стороні друкованої плати та мають однакову ширину визначається за виразом (4.17):

$$C = 0,12 \cdot \varepsilon \cdot l_{\text{пр}} \cdot \left[\lg \frac{2 \cdot S}{b_{\text{пр}} + t_{\text{пр}}} \right]^{-1}, \quad (4.17)$$

					ДК62.469515.001 ПЗ	Арк. 50
Змн.	Арк.	№ докум.	Підпис	Дата		

де ε — діелектрична проникність, $\varepsilon = 4,5$;

$l_{\text{пр}}$ — довжина взаємного перекриття двох паралельних провідників,

$$l_{\text{пр}} = 10,5 \text{ мм};$$

S — відстань між двома паралельними провідниками, $S = 0,75 \text{ мм}$;

$b_{\text{пр}}$ — ширина друкованого провідника, $b_{\text{пр}} = 0,25 \text{ м}$;

$t_{\text{пр}}$ — товщина друкованого провідника, $t_{\text{пр}} = 0,097 \text{ мм}$, визначено вище у пункті 4.1.1.

$$C = 0,12 \cdot 4,5 \cdot 10,5 \cdot \left[\lg \frac{2 \cdot 0,75}{0,25 + 0,097} \right]^{-1} = 1,56 \text{ (пФ)}$$

4.2.4. Визначення взаємної індуктивності двох паралельних провідників однакової довжини

Взаємна індуктивність двох паралельних провідників однакової довжини визначається за виразом (4.18):

$$M = 0,02 \cdot \left(l_{\text{пр}} \cdot \lg \left(\frac{\sqrt{l_{\text{пр}}^2 + L_0^2} + l_{\text{пр}}}{L_0} \right) - \sqrt{l_{\text{пр}}^2 + L_0^2} + L_0 \right), \quad (4.18)$$

де $l_{\text{пр}}$ — довжина перекриття паралельних провідників, $l_{\text{пр}} = 10,5 \text{ см}$;

L_0 — відстань між осьовими лініями двох паралельних провідників, $L_0 = 0,1 \text{ см}$.

$$M = 0,02 \cdot \left(10,5 \cdot \lg \left(\frac{\sqrt{10,5^2 + 0,1^2} + 10,5}{0,1} \right) - \sqrt{10,5^2 + 0,1^2} + 0,1 \right) \\ = 0,28 \text{ (нГн)}$$

4.3. Розрахунок основних показників надійності друкованого вузлу

Надійність — це властивість об'єкту зберігати на протязі певного часу установлені значення певних параметрів, які характеризують можливість виконувати потрібні функції в заданих режимах і умовах експлуатації, технічного обслуговування, ремонтів, збереження і транспортування (ГОСТ 27.002 – 83).

					ДК62.469515.001 ПЗ	Арк. 51
Змн.	Арк.	№ докум.	Підпис	Дата		

Надійність друкованого вузлу визначається надійністю всіх елементів, що встановлюються на ДП, власне самою ДП та пайкою елементів. Відмова одного чи декількох елементів веде до відмови всієї складної ЕОА, при чому ці елементи фактично є невідновлюваними.

Основними показниками та характеристиками надійності є :

- $P(t)$ – імовірність безвідмовної роботи;
- $\lambda(t)$ – інтенсивність відмов;
- T – середній час напрацювання до першої відмови;
- $Q(t)$ – імовірність відмови.

Інтенсивність відмов ЕРЕ являється експериментальною величиною і залежить від ряду факторів: температури навколишнього середовища, теплових ударів, вологості, режиму роботи і т.д.

Важлива характеристика надійності — середній час безвідмовної роботи визначається за виразом (4.19):

$$T_{cp} = \frac{1}{\lambda}, \quad (4.19)$$

де λ — інтенсивність відмов, (год⁻¹).

Інтенсивність відмов ЕРЕ є їх вихідною характеристикою надійності, залежить від режиму роботи та ступеню тяжкості таких зовнішніх впливів, як температура, тепловий удар, вологість, вібрації і т.д. Таким чином можна записати наступний вираз (4.20):

$$\lambda_e = \lambda_{0e} \cdot K_1 \cdot K_2 \cdot \dots \cdot K_n, \quad (4.20)$$

де λ_{0e} – інтенсивність відмов за нормальних умов (температура навколишнього середовища $T = 20 \pm 5$ °С, відносна вологість $65 \pm 15\%$), параметри обираються з [29];

$K_0 \dots K_n$ – поправочні коефіцієнти, що враховують режими роботи та умови експлуатації.

Для врахування впливу режиму роботи на інтенсивність відмов ЕОА вводять коефіцієнт навантаження $K_n = \frac{H_{роб}}{H_{ном}}$, що дорівнює відношенню

					ДК62.469515.001 ПЗ	Арк.
Змн.	Арк.	№ докум.	Підпис	Дата		52

навантаженню в робочому режимі до навантаження в номінальному режимі [44].

Коефіцієнт навантаження для резисторів визначається за виразом (4.21):

$$K_{H(R)} = \frac{P_{\text{роб}}}{P_{\text{ном}}} = \frac{U_{\text{роб}}^2}{R \cdot P_{\text{ном}}} \quad (4.21)$$

Резистори мають номінальну потужність 0,125 Вт та робочу напругу 3,3 В, тому результати розрахунку представлено нижче у Таблиця 4.3:

Таблиця 4.3 — Коефіцієнти навантаження для резисторів

Номер	Опис	Коефіцієнт
R1	R-0805 0,125 Вт 4,7 кОм 1%	0,018
R3-R7	R-0805 0,125 Вт 10 кОм 1%	0,009
R2	R-0805 0,125 Вт 1,2 МОм 1%	0,002

Коефіцієнт навантаження для конденсаторів визначається за виразом (4.22):

$$K_{H(C)} = \frac{U_{\text{роб}}}{U_{\text{ном}}} \quad (4.22)$$

Конденсатори мають номінальну напругу 25 В та різну робочу напругу, тому результати розрахунку представлено нижче у Таблиця 4.4:

Таблиця 4.4 — Коефіцієнти навантаження для конденсаторів

Номер	Робоча напруга, В	Коефіцієнт
C6,C8-C20,C22,C23	3,3	0,132
C1-C5	5	0,2
C7,C21,C24	15	0,6

Для усіх інших елементів вибираємо $K_H = 1$. Інтенсивність відмов друкованої плати визначатимемо як інтенсивність відмов металізованих отворів.

Визначимо результуючу інтенсивність відмов друкованого вузлу формувача величини напруги. Друкований вузол відноситься до наземної апаратури, експлуатується при $T_p = 40^\circ\text{C}$, інші експлуатації нормальні.

Вихідні дані для розрахунку λ_p — схема принципова, перелік елементів, часова діаграма та інтенсивність відмов “компонентів надійності” від температурних впливів.

По картам робочих режимів визначаємо коефіцієнти навантаження, температурні коефіцієнти ІС та інших ЕРЕ, підраховуємо кількість всіх елементів. Вихідні дані для визначення λ_p зведені до Таблиця 4.5:

Таблиця 4.5 — Коефіцієнти надійності для елементної бази

Компонент	N	$\lambda_{0e} \cdot 10^{-7}, \text{год}^{-1}$	K_H	a_t	a_e	$N \cdot \lambda_{0e} \cdot K_H \cdot a_t \cdot a_e$
Резистор						
R1	1	0,44	0,018	0,15	15	0,0178
R3-R7	5	0,44	0,009	0,15	15	0,0446
R2	1	0,44	0,002	0,15	15	0,002
Конденсатор						
C6,C8-C20, C22,C23	16	0,22	0,132	0,4	15	2,7878
C1-C5	5	0,22	0,2	0,4	15	1,32
C7,C21,C24	3	0,22	0,6	0,4	15	2,376
Кварцевий резонатор	2	0,6	1	1	15	18
Мікросхема аналогова	3	0,28	1	1	15	12,6
Мікросхема цифрова	2	0,23	1	1	15	6,9
Перемикач кнопковий	1	0,3	1	1	15	4,5
Дисплей	1	8,8	1	1	15	132
Роз'єм	2	0,62	1	1	15	18,6
Друкована плата	2	1	1	1	15	30
Контакт роз'єму	34	0,2	1	1	15	102
Перехідні отвори	66	0,0375	1	1	15	48,375
Пайка виводів	152	0,005	1	1	15	11,4

В таблиці: a_e — поправочний коефіцієнт щодо зовнішніх впливів (для переносної апаратури $a_e = 15$), a_t — поправочний температурний коефіцієнт.

Показники інтенсивності відмов елементів, що наведені в таблиці, дещо завищені, що дозволяє виконати розрахунок для “найгіршого” випадку. Результуюча інтенсивність відмов дорівнює сумі інтенсивностей відмов компонентів виконується за виразом (4.23):

$$\lambda_p = \sum_{i=1}^n \lambda_{pi}, \quad (4.23)$$

де: $\lambda_{pi} = N \cdot \lambda_{0e} \cdot K_n \cdot a_t \cdot a_e$;

N — кількість елементів даного типу;

λ_{0e} — інтенсивність відмов в год⁻¹;

K_n — коефіцієнт навантаження;

a_t — поправочний температурний коефіцієнт;

a_e — поправочний коефіцієнт на зовнішні впливи.

Сумарна інтенсивність відмов друкованого вузла:

$$\lambda_p = 379,673 \cdot 10^{-7} \text{ (год}^{-1}\text{)}$$

Середній час напрацювання до першої відмови виконується за виразом (4.24):

$$T_{cp} = \frac{1}{\lambda_p} \quad (4.24)$$

$$T_{cp} = \frac{1}{379,673 \cdot 10^{-7}} = 26338,5 \text{ (год)}$$

Ймовірність безвідмовної роботи на протязі року виконується за виразом (4.25):

$$P = e^{-\lambda_p \cdot t} \quad (4.25)$$

$$P = e^{-379,673 \cdot 10^{-7} \cdot 8760} = 0,717$$

Ймовірність відмов на протязі року:

$$Q(t) = 1 - 0,717 = 0,283$$

					ДК62.469515.001 ПЗ	Арк. 55
Змн.	Арк.	№ докум.	Підпис	Дата		

Графік залежності безвідмовної роботи ДВ та ймовірність відмов ДВ від часу представлено нижче на Рис.4.4:

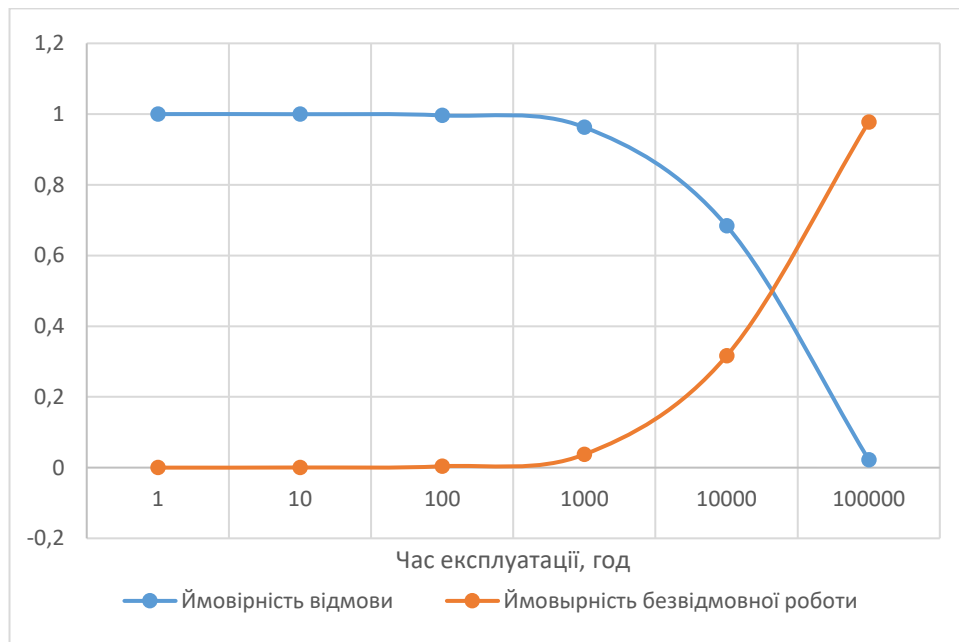


Рис.4.4 — Ймовірність відмовної і безвідмовної роботи

Даний рисунок показує, що графіки змін характеристик імовірності безвідмовної роботи $P(t)$ та відмови $Q(t)$ мають спільну точку перетину, а отже їх імовірності порівнюються. Це станеться після 16000 годин безвідмовної роботи, але даний результат було отримано без урахування таких факторів, як раптові відмови, старіння приладу, його знос і т.д..

4.4. Розрахунок віброміцності друкованого вузла

Для забезпечення віброміцності друкованої плати необхідно виконання наступних вимог [31]:

- відсутність в конструкції механічних резонансів;
- обмеження амплітуди вібропереміщення значеннями, що виключають;
- небезпечні механічні напруження.

Основні параметри склотекстоліту FR-4 [31]:

- тимчасовий опір $\sigma_T = 105$ (МПа);
- модуль Юнга $E = 3,02 \cdot 10^{10}$ (Па);
- коефіцієнт Пуассона $\mu = 0,22$;
- показник затухання $\varepsilon = 0,06$;

- питома вага $\rho_{FR-4} = 1850$ (кг/м³);
- питома щільність $\nu = 2,05 \cdot 10^4$ (Н/м³);
- коефіцієнт запасу міцності $n1 = 2$.

Плата пристрою, що проектується, має лінійні розміри $a \times b \times h = 125 \times 65 \times 1,5$ (мм). Маса друкованої плати розраховується за виразом (4.26):

$$m_{\text{ДП}} = \rho \cdot V = \rho_{FR-4} \cdot (a \cdot b \cdot h), \quad (4.26)$$

де $\rho_{FR-4} = 2,05$ г/см³ — густина склотекстоліту FR-4, враховуючи друковані провідники.

$$m_{\text{ДП}} = 1,85 \cdot 7,5 \cdot 6,5 \cdot 0,15 = 13,53$$

Розрахунок маси елементів занесено до Таблиця 4.6:

Таблиця 4.6 — Маса елементів друкованої плати

Елемент	Позначення	Кількість	Маса, г	Загальна маса, г
Конденсатор керамічний	0805-X5R	24	0,006	0,144
Резистор	SMD 0,25 Вт 0805	7	0,003	0,021
Мікросхеми	LD1117S50CTR	1	0,25	0,25
	LP3981ILD-3.3	1	0,01	0,01
	TLV76015DBZR	1	0,112	0,112
	MCP2200T-I/SO	1	0,801	0,801
	STM32F411RET6	1	0,343	0,343
Перемикач	PB4000AQX	1	8,504	8,504
Роз'єм	47346-0001	1	0,247	0,247
	UJ2-AH-1-SMT	1	1,885	1,885
Дисплей	RGS15128128YR000	1	10	10
Кварцевий резонатор	LFXTAL031823Bulk	1	1,2	1,2
	LFXTAL031826Bulk	1	1,2	1,2
Загальна маса елементів, г				24,717

Загальна маса елементів які входять до пристрою становить $m_E = 24,717$ г. Коефіцієнт перенавантаження $n = 8$. Вибраний тип закріплення друкованої плати — спірання на 3 сторони (Рис.4.5), для якого коефіцієнт α визначається за виразом (4.26)(4.27).

Спірання на 3 сторони

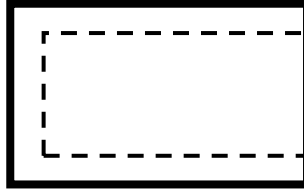


Рис.4.5 — Варіанти закріплення друкованої плати

$$\alpha = \pi^2 \cdot \sqrt{1 + 0,43 \cdot \frac{a^2}{b^2}} \quad (4.27)$$

Визначення коефіцієнту K_B здійснюється за виразом (4.28):

$$K_B = \frac{1}{\sqrt{1 + \frac{m_E}{m_{ДП}}}} \quad (4.28)$$

$$K_B = \frac{1}{\sqrt{1 + \frac{24,717}{13,53}}} = 0,595$$

Коефіцієнт α , враховуючи вибраний тип закріплення буде дорівнювати:

$$\alpha = \pi^2 \cdot \sqrt{1 + 0,43 \cdot \frac{0,075^2}{0,065^2}} = 12,364$$

Циліндрична жорсткість визначається за виразом (4.29):

$$D = \frac{E \cdot h^3}{12 \cdot (1 - \mu)} \quad (4.29)$$

$$D = \frac{3,02 \cdot 10^{10} \cdot (1,5 \cdot 10^{-3})^3}{12 \cdot (1 - 0,22^2)} = 8,926 \text{ (Н} \cdot \text{м)}$$

Власна частота коливань друкованої плати визначається за виразом (4.30):

					<i>ДК62.469515.001 ПЗ</i>	Арк. 58
Змн.	Арк.	№ докум.	Підпис	Дата		

$$f_{\text{влас}} = \frac{K_B \cdot \alpha}{2 \cdot \pi \cdot a^2} \cdot \sqrt{\frac{D \cdot g}{v \cdot h}} \quad (4.30)$$

$$f_{\text{влас}} = \frac{0,595 \cdot 12,364}{2 \cdot \pi \cdot 0,075^2} \cdot \sqrt{\frac{8,926 \cdot 9,81}{2,05 \cdot 10^4 \cdot 1,5 \cdot 10^{-3}}} = 370,326 \text{ (Гц)}$$

Оскільки $f_{\text{влас}} > 250$ Гц, то конструкція можна вважати абсолютно жорсткою [31].

Висновок:

В даному розділі розраховано:

- мінімальну ширину провідників на постійному струмі $b_{\min I} = 0,172$ мм, яка відповідає вимогам 3-го класу точності $b_{\text{пр}}^r = 0,25$ мм;
- мінімальну ширину провідників з урахуванням допустимого падіння наруги $b_{\min U} = 0,021$ мм, в той час як було обрано ширину даних провідників 0,3 мм;
- номінальний діаметр монтажного отвору $d \geq 0,8$ мм, мінімальний та максимальний діаметр контактної площадки $D_{\min} = 1,47$ мм, $D_{\max} = 1,49$ мм відповідно;
- мінімальна та максимальна ширина провідників $b_{\min} = 0,33$ мм, $b_{\max} = 0,35$ мм відповідно;
- мінімальну відстань між провідником та КМ $l_{\text{ПКМ min}} = 0,23$ мм;
- мінімальну відстань між сусідніми провідниками $l_{\text{ПП min}} = 0,8$ мм;
- мінімальну відстань між КМ $l_{\text{min КМ КМ}} = 1,21$ мм;
- допустиме падіння напруги на найдовшому ДП $U_{\text{пад}} = 0,37$ мВ, що не перевищує похибку від напруги живлення 5 В;
- Потужність втрат ДДП для високих частот $P_{\text{пот}} = 74,7$ Вт та для низьких частот $P_{\text{пот}} = 597,8$ Вт не впливають на роботу приладу;
- Паразитна ємність між сусідніми провідниками $C = 1,56$ пФ, що порівняно з конденсатором на схемі ємністю 15 пФ являється в 10 разів меншим значенням;

					ДК62.469515.001 ПЗ	Арк.
Змн.	Арк.	№ докум.	Підпис	Дата		59

- Взаємна індуктивність двох паралельних провідників однакової довжини $M = 0,28$ нГн;
- Надійність ДП, який показав, що проектована ДП буде працювати до першої відмови біля 2 років, що задовольняє умові ТЗ (не менше 10000 годин);
- Віброміцність друкованого вузлу, який показав, що конструкція являється абсолютно жорсткою.

					<i>ДК62.469515.001 ПЗ</i>	<i>Арк.</i>
<i>Змн.</i>	<i>Арк.</i>	<i>№ докум.</i>	<i>Підпис</i>	<i>Дата</i>		60

Розділ 5. РОЗРОБКА ПРОГРАМНОГО ЗАБЕЗПЕЧЕННЯ

5.1. Програмування мікроконтролера STM32F411RET6

В якості керуючого пристрою використовується мікроконтролер STM32F411RET6. Мікроконтролер забезпечує зчитування даних з клавіатури за допомогою інтерфейсу UART і виводом результату на OLED-дисплей. Створення даного проекту було створено за допомогою CubeMX. Налаштування мікроконтролера для роботи в даному середовищі зображено нижче на Рис.5.1:

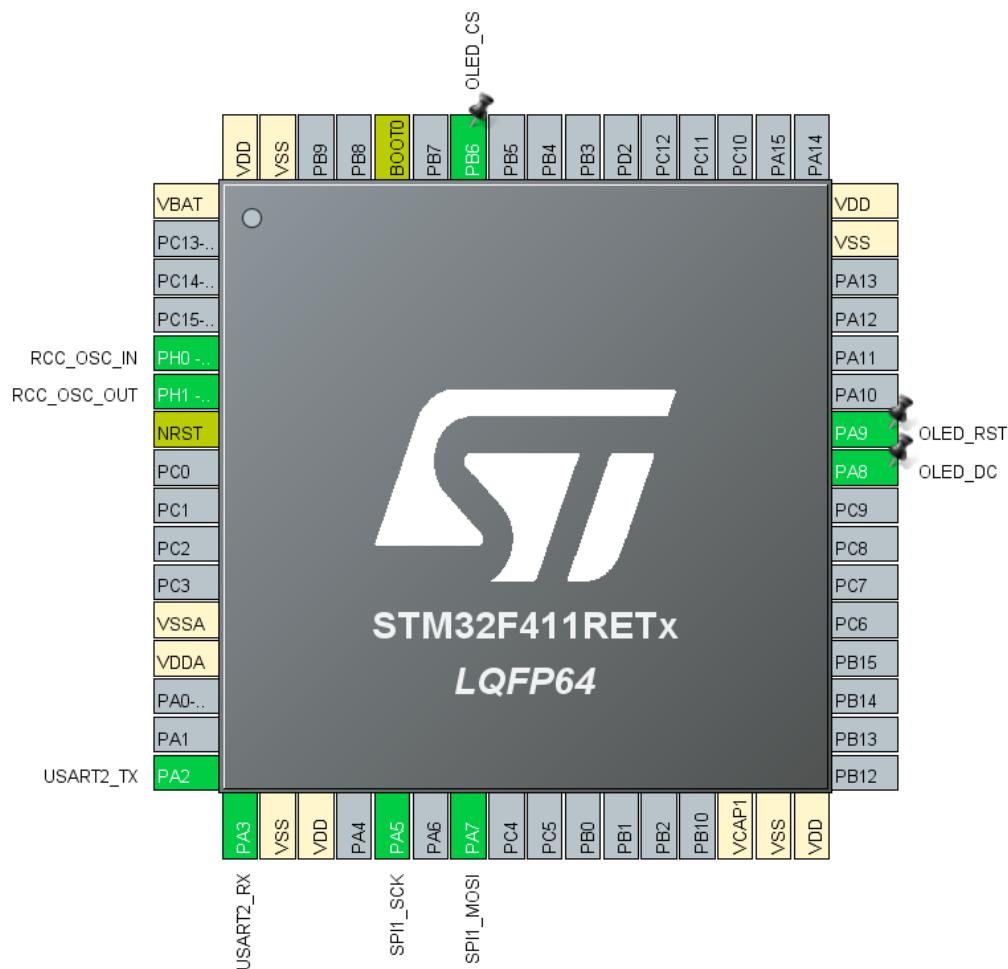


Рис.5.1 — Налаштування STM32F411RET6 в CubeMX

5.2. Структура даних

Нижче наведена таблиця, в якій пояснено які виводи плати NUCLEO використовуються і яке їх призначення (Таблиця 5.1):

					ДК62.469515.001 ПЗ	Арк. 61
Змн.	Арк.	№ докум.	Підпис	Дата		

Таблиця 5.1 — Використані виводи плати NUCLEO

Назва	Призначення
PA3	Відправлення даних через UART
PA2	Отримання даних через UART
PB6	CS для SPI (вибір мікросхеми / дозвіл роботи з мікросхемою)
PA7	MOSI для SPI (відправлення даних)
PA5	SCK для SPI (тактова частота)
PA9	Reset для OLED-дисплея
PA8	DC для OLED-дисплея (режим даних / команд)
3V3	Підключення живлення
GND	Підключення «землі»

Далі наведено таблиця, яка містить назви використаних змінних, тип та їх призначення (Таблиця 5.2):

Таблиця 5.2 — Використані змінні

Назва	Тип змінної	Призначення
str	8 розрядна змінна типу integer	Збереження останнього введеного значення ASCII
new_str	32 розрядна змінна типу char (масив з 14-ми символами)	Збереження введеного тексту, надрукованого на OLED-дисплеї
cnt_rec	32 розрядна змінна типу integer	Збереження кількості збережених символів в змінній new_str
state	32 розрядна змінна типу integer	Збереження режиму відображення (текст/QR-код)
str_size	32 розрядна змінна типу char (масив з 9-ма символами)	Збереження тексту для відображення кількості введених символів, надрукованого на OLED-дисплеї

В наступній таблиці показано які регістри мікроконтролера використовуються при його програмуванні та їх призначення (Таблиця 5.3):

Таблиця 5.3 — Регістри мікроконтролера, застосовані в програмі

Назва	Призначення
АНВ1ENR	Ввімкнення тактування портів GPIOA та GPIOB
GPIOB(MODER)	Налаштування роботи PB6 на альтернативну функцію

Таблиця 5.4. Продовження.

GPIOB(OSPEEDR)	Вибір швидкодії GPIOB
GPIOB(AFRL)	Вибір альтернативної функції PB6
GPIOC(MODER)	Налаштування роботи PC8, PC6, PC9 на вхід
GPIOC(OSPEEDR)	Вибір швидкодії GPIOC
TIM4(PSC)	Задає значення дільника частоти таймера TIM4
TIM4(CR1)	Налаштування таймера TIM4 (напрям рахунку, вирівнювання)
TIM4(ARR)	Задає значення до якого буде рахувати таймер TIM4 (завдання періоду імпульсів)
TIM4(CCMR1)	Вибір режиму PWM1 для таймера TIM4
TIM4(CCER)	Встановлення полярності вихідного ШІМ сигналу та дозвіл ШІМ на вихід
TIM4(CCR1)	Задає довжину імпульсу ШІМ сигналу (разом із ARR визначає шпаруватість (скважність) ШІМ)
TIM4(DIER)	Дозвіл переривання по переповненню таймера
TIM4(CCER)	Встановлення полярності вихідного ШІМ сигналу та дозвіл ШІМ на вихід
TIM4(CCR1)	Задає довжину імпульсу ШІМ сигналу (разом із ARR визначає шпаруватість (скважність) ШІМ)
TIM4(DIER)	Дозвіл переривання по переповненню таймера

5.3. Опис лістингу

Програма керування пристроєм написана на мові програмування C з використанням бібліотеки CMSIS та HAL. Після виконання стартового файлу

					ДК62.469515.001 ПЗ	Арк.
Змн.	Арк.	№ докум.	Підпис	Дата		63

програма переходить на функцію main(). В даній функції спочатку відбувається ініціалізація HAL:

```
HAL_Init();
```

Виконуємо налаштування PLL:

```
SystemClock_Config();
```

Далі викликаємо ініціалізацію GPIO, SPI, UART та OLED-дисплея:

```
MX_GPIO_Init();
```

```
MX_SPI1_Init();
```

```
MX_USART2_UART_Init();
```

```
OLED_Init(SCAN_DIR_DFT);
```

Далі очищуємо екран у чорний колір та виконуємо тестуючу заставку:

```
OLED_Clear(BLACK);
```

```
GUI_Show();
```

Після цього записуємо текст на екран "Pls writing text" та відображуємо його:

```
GUI_DisString_EN(0, 48, "Pls writing text", &Font12, BLACK, WHITE);
```

```
OLED_Display();
```

Далі ініціалізуємо змінні для подальшого їх використання:

```
uint8_t str[1];
```

```
char new_str[14];
```

```
int cnt_rec = 0;
```

```
int state = 0;
```

```
char str_size[9];
```

Далі виконуємо нескінченний цикл, в якому спочатку очікуємо за допомогою UART прийняття даних, після чого передаємо ці данні назад для відображення, що саме було прийнято:

```
while(1) {
```

```
HAL_UART_Receive(&huart2, str, 1, 0xFFFFFFFF);
```

```
HAL_UART_Transmit(&huart2, str, 1, 0xFFFFFFFF);
```

					ДК62.469515.001 ПЗ	Арк.
Змн.	Арк.	№ докум.	Підпис	Дата		64

Далі завжди на початку циклу необхідно очистити екран від попереднього використання та виконуємо перевірку на режим відображення:

```
OLED_Clear(BLACK);  
if(state == 0) {  
    if((char)str[0] == 13) {  
        state = 1;  
    } else if((char)str[0] == 8) {  
        if(0 < cnt_rec){  
            cnt_rec--;  
            new_str[cnt_rec] = NULL;  
        }  
    }  
}
```

Тут виконуються перевірки на кнопки «Enter» та «Backspace», якщо виконується перша кнопка, тоді змінюємо значення state на 1, а якщо натиснули «Backspace», тоді стираємо з пам'яті даний символ на зменшуємо кількість надрукованих літер на 1 (при умові якщо cnt_rec > 0).

Далі, якщо кнопки перерахування вище не натиснуті, тоді виконуємо перевірку на те, що записано одну з літер ASCII та записуємо в масив дану літеру, після чого збільшуємо лічильник символів на 1.

```
else {  
    if(32 <= (char)str[0] && (char)str[0] <= 126) {  
        new_str[cnt_rec] = str[0];  
        cnt_rec++;  
    }  
}
```

Далі, якщо лічильник переповнився (більше за 14), тоді очищуємо рядок:

```
if(cnt_rec > 14) {  
    cnt_rec = 0;  
    for(int i = 0; i < 14; i++) {
```

					ДК62.469515.001 ПЗ	Арк. 65
Змн.	Арк.	№ докум.	Підпис	Дата		

```

        new_str[i] = NULL;
    }
}

```

Далі, при state = 1 знову виконуємо перевірку на “Enter”, для того щоб перейти в попередній режим.

```

} else if(state == 1) {
    if((char)str[0] == 13) {
        state = 0;
    }
}

```

Останнім кроком буде відображення тексту при state = 0, або генерація та відображення QR-коду при state = 1:

```

if(state == 0) {
    GUI_DisString_EN(0, 0, new_str, &Font12, BLACK, WHITE);
    sprintf(str_size, "size = %d", cnt_rec);
    GUI_DisString_EN(0, 20, str_size, &Font12, BLACK, WHITE);
} else if(state == 1) {
    gen_QR(new_str, cnt_rec);
    draw_QR(10,10);
}
OLED_Display();

```

5.3.1. Опис функції Oled_Init():

Головною метою даної функції виконати ініціалізацію екрана. Для цього виконується передача на вхід даних стандартні команди (DC встановлюється в 0), деякі описано нижче:

```

OLED_WriteReg(0xae);/-- вимкнути oled панель
OLED_WriteReg(0x15); // встановити адресу стовця
OLED_WriteReg(0x00); // стартовий стовчик 0
OLED_WriteReg(0x7f); // кінцева колонка 127
OLED_WriteReg(0x75); // Встановити адресу рядка

```

					ДК62.469515.001 ПЗ	Арк. 66
Змн.	Арк.	№ докум.	Підпис	Дата		

```

OLED_WriteReg(0x00); // початковий рядок 0
OLED_WriteReg(0x7f); // кінцевий рядок 127
OLED_WriteReg(0x81); // встановити контроль контрасту
OLED_WriteReg(0x80);
OLED_WriteReg(0xa1); // Початковий рядок
OLED_WriteReg(0x00);
OLED_WriteReg(0xa2); // зміщення дисплея
OLED_WriteReg(0x00);
OLED_WriteReg(0xa8); // встановити співвідношення мультимплексу
OLED_WriteReg(0xb3); // встановити dclk
OLED_WriteReg(0x00); //80Hz:0xc1 90Hz:0xe1 100Hz:0x00 110Hz:0x30
120Hz:0x50 130Hz:0x70 01

```

4.3.2 Опис функції OLED_WriteReg/ OLED_WriteData:

За допомогою даних функцій виконується передача даних/команд:

```
void OLED_WriteReg(uint8_t Reg)
```

```
{
    OLED_DC_0;
    OLED_CS_0;
    SPI4W_Write_Byte(Reg);
    OLED_CS_1;
}
```

```
void OLED_WriteData(uint8_t Data)
```

```
{
    OLED_DC_1;
    OLED_CS_0;
    SPI4W_Write_Byte(Data);
    OLED_CS_1;
}
```

В результаті сигнали мають наступний вигляд (Рис. 5.2):

					ДК62.469515.001 ПЗ	Арк.
Змн.	Арк.	№ докум.	Підпис	Дата		67

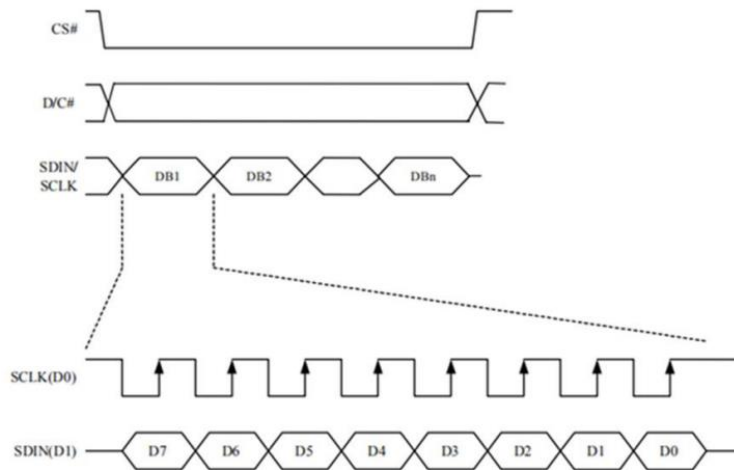


Рис. 5.2 — вигляд сигналів під час передачі даних по SPI

5.3.2. Опис функції `GUI_DisString_EN()`:

Дана функція має наступну структуру:

```
void GUI_DisString_EN(POINT Xstart, POINT Ystart, const char * pString,
sFONT* Font, COLOR Color_Background, COLOR Color_Foreground );
```

Ця функція використовується для відображення тексту. Для цього подаємо початкові координат по X та Y, сам текст у форматі `char*`, задаємо його колір (чорний або білий), на колір фону.

5.3.3. Опис функції `GUI_DrawLine ()`:

Дана функція має наступну структуру:

```
void GUI_DrawLine(POINT Xstart, POINT Ystart, POINT Xend, POINT Yend,
COLOR Color, LINE_STYLE Line_Style, DOT_PIXEL Dot_Pixel);
```

Ця функція використовується для намалювання лінії. Для цього функції передаємо початкові та кінцеві координати, колір, стиль лінії(штих/суцільна), та товщина в пікселях.

5.3.4. Опис функції `draw_QR()`:

Дана функції відповідає за відображення QR-коду.

```
void draw_QR(int x_start, int y_start)
```

```
{
```

```
    GUI_DrawLine(x_start, y_start, x_start + 22 * 5, y_start, WHITE,
LINE_SOLID, DOT_PIXEL_3X3);
```

					<i>ДК62.469515.001 ПЗ</i>	<i>Арк.</i>
<i>Змн.</i>	<i>Арк.</i>	<i>№ докум.</i>	<i>Підпис</i>	<i>Дата</i>		<i>68</i>

```

    GUI_DrawLine(x_start, y_start, x_start, y_start + 22 * 5, WHITE,
LINE_SOLID, DOT_PIXEL_3X3);

    GUI_DrawLine(x_start + 22 * 5, y_start, x_start + 22 * 5, y_start + 22 * 5,
WHITE, LINE_SOLID, DOT_PIXEL_3X3);

    GUI_DrawLine(x_start, y_start + 22 * 5, x_start + 22 * 5, y_start + 22 * 5,
WHITE, LINE_SOLID, DOT_PIXEL_3X3);

    for(int y = 0; y < 21; y++){
        for(int x = 0; x < 21; x++){
            if(matr[y][x] == 0) {
                GUI_DrawPoint(x_start + 5 * (x + 1), y_start + 5 * (y +
1), WHITE, DOT_PIXEL_3X3, DOT_FILL_AROUND);
            }
        }
    }
}

```

Тут спочатку по краям малюються лінії, щоб камера могла зчитати даний код, після чого малюються квадрати по чергово, взявши дані з масиву розміром 21x21, зберігаючи основний код QR-коду.

5.4. Тестування роботи програмного забезпечення

Для тестування програмного забезпечення використовувалась налагоджувальна плата STMicroelectronics NUCLEO-F411RET6 та OLED-дисплей «1.5inch OLED Module». Тому для цього необхідно виконати наступне:

1. Зібрати макет згідно схеми зображеної нижче (Рис. 5.3):
2. Підключити модуль USB -> miniUSB кабель до ПК.
3. В IDE ARM Keil зібрати та скомпілювати проект
4. Загрузити скомпільований код на плату NUCLEO F411.

					<i>ДК62.469515.001 ПЗ</i>	Арк.
Змн.	Арк.	№ докум.	Підпис	Дата		69

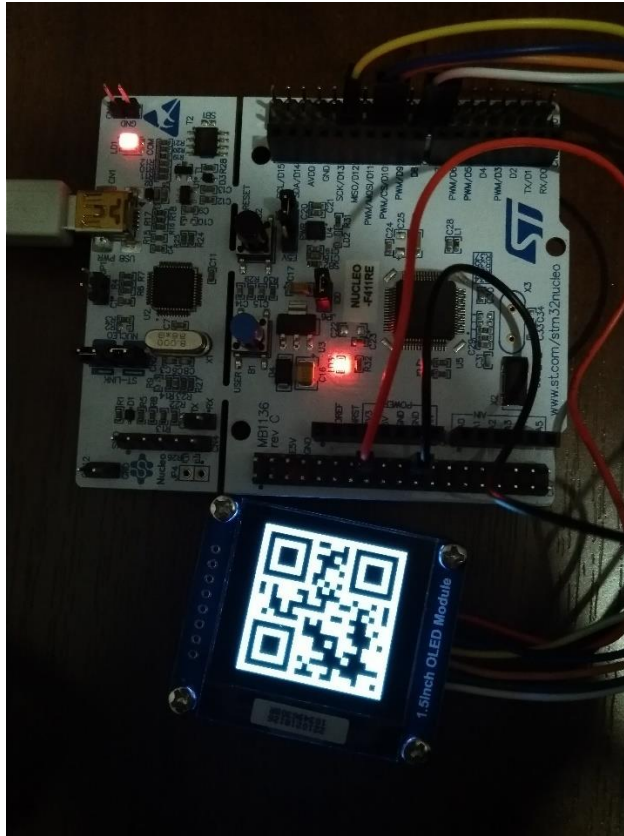


Рис. 5.5 — Результат генерування QR-коду

Висновок:

У даному розділі було розроблено програмне забезпечення у середовищі IDE ARM Keil. Написання програмного коду складалось з двох частин:

- Розробка алгоритму генерування QR-коду
- Обробка вхідних та вихідних даних.

Було протестовано та розглянуто результати роботи розробленого програмного забезпечення, в результаті чого всі згенеровані QR-коди були відскановані мобільним телефоном і результатами сканування виявились вірними.

					<i>ДК62.469515.001 ПЗ</i>	Арк.
Змн.	Арк.	№ докум.	Підпис	Дата		71

Висновки

В рамках дипломного проекту розроблено пристрій для генерування QR-коду. Під час написання першого розділу проведено аналіз існуючих пристроїв та виконано патентний пошук, в результаті чого вирішено спроектувати пристрій, який призначений для ідентифікації товарів у торговельному підприємстві. Також розроблено та протестовано генератор QR-коду за допомогою технології ПЛІС.

У другому розділі розроблено схему електричну принципову генератора QR-кода, яка містить 2 роз'єми: перший використовується для живлення схеми, другий для підключення клавіатури у якості вхідних даних. Для обробки даних обрано мікроконтролер STM32F411RET6. Також обрано перетворювач інтерфейсів MCP2200, який перетворює USB в UART інтерфейс. Оскільки схема містить 3 живлення (+3 В, +5 В, +15 В), тому до даної схеми додано також 3 перетворювачі напруги. Для перезапуску та скидання даних також обрано кнопку PB400. Також проаналізовано алгоритм генерації QR-коду.

У третьому розділі: розглянуто та обрано тип друкованої плати, обрано метод виготовлення ДП, обрано клас точності монтажу, розміщено KE в монтажному просторі та спроектовано ДП у середовищі AD. Виходячи з вимог ДП обрано ДДП, яка виготовлена зі склотекстоліту FR402-35-1,5 з 3-ім класом точності. Для ДП вирішено використовувати комбінований позитивний метод, оскільки даний метод широко використовується і має ряд своїх переваг перед іншими методами. В кінці розділу виконано розміщення KE в монтажному просторі та зпроектовано ДП в середовищі AD.

У четвертому розділі проведено конструкторсько-технологічні і електричні розрахунки, що підтверджують правильність прийнятих конструкторських рішень. Дані розрахунки підтвердили правильність обирання: мінімальної ширини провідників, номінальний діаметр монтажного отвору, мінімальні відстані між провідниками та КМ, допустиме падіння напруги на найдовшому ДП, і т.д.. Розраховано потужність втрат

									Арк.
									72
Змн.	Арк.	№ докум.	Підпис	Дата	ДК62.469515.001 ПЗ				

друкованої плати рівна для високих частот $P_{\text{пот}} = 74,7$ Вт та для низьких частот $P_{\text{пот}} = 597,8$ Вт, що є достатньо малою величиною, і можна стверджувати, що це ніяк не вплине на функціонування пристрою. Отримані значення паразитної ємності ($C = 1,56$ пФ) та взаємної індуктивності ($M = 0,28$ нГн), значення яких не впливають на працездатність схеми. Також розраховано надійність ДП, який показав, що проєктована ДП може працювати до першої відмови біля 2 років, що задовольняє умові ТЗ (не менше 10000 годин). Розраховано віброміцність друкованого вузлу, який показав, що конструкція являється абсолютно жорсткою.

У останньому розділі розроблено програмне забезпечення у середовищі IDE ARM Keil. Протестовано та розглянуто результати роботи розробленого програмного забезпечення, в результаті чого всі згенеровані QR-коди були відскановані мобільним телефоном і результатами сканування виявились вірними. Створене програмне забезпечення підтримує лише QR-коди першої версії та наразі може кодувати до 14 символів даних, застосовуючи побайтове кодування та рівень корекції типу М (дозволяє відновлювати інформацію із зображення, яке пошкоджено на 15%).

Враховуючи наявність обмежень маємо новий пристрій. За рахунок даних обмежень пристрій хоч може зберігати меншу кількість даних, але має більшу швидкість генерування QR-коду.

Враховуючи наявні обмеження у поточній реалізації пристрою, основою для подальшого його покращення можуть стати:

- збільшення довжини кодованих даних;
- підтримка інших методів кодування (цифрового та побайтового);
- додати наступні рівні корекції: L для 7%, Q для 15% та H для 30% пошкоджень;
- можливість використання інших масок.

Матеріали представлені в роботі пройшли апробацію на конференції «Електроніка-2020» [32].

					<i>ДК62.469515.001 ПЗ</i>	Арк.
Змн.	Арк.	№ докум.	Підпис	Дата		73

Література

1. Генератор QR кодов [Електронний ресурс] — Режим доступу: <http://qrcoder.ru/>.
2. Почему нам стоит присмотреться к QR-кодам в 2019 году, [Електронний ресурс] — Режим доступу: <https://cutt.ly/iyMjV8l>.
3. Wireless-handheld [Електронний ресурс] — Режим доступу: <https://cutt.ly/yyFAefX>.
4. POS-terminal [Електронний ресурс] — Режим доступу: <https://cutt.ly/MyFP70V>.
5. Liguori D., “Hand-held qr code label printer using a smartphone”, US patent 20160350624, Dec 1, 2016.
6. Moore R., “Visual biometric authentication supplemented with a time-based secondary authentication factor”, US patent 20170264608, Sep 14, 2017.
7. Verilog Language [Електронний ресурс] — Режим доступу: <https://cutt.ly/byMjNKX>.
8. Terasic DE1-SoC User Manual, 2014, p. pp. 30–33.
9. Алгоритм генерации QR-кода [Електронний ресурс] — Режим доступу: <https://cutt.ly/VyFF4ZZ>.
10. К. Касперски, Полиномиальная арифметика и поля Галуа, или информация, воскрешая из пепла, p. 84–90, 2003.
11. Wounded QR codes [Електронний ресурс] — Режим доступу: <https://cutt.ly/wyFGyDO>.
12. Алгоритм распознавания графического маркера [Електронний ресурс] — Режим доступу: <https://cutt.ly/fyFGh4r>.
13. STM32F411RET6. Datasheet [Електронний ресурс] — Режим доступу: <https://cutt.ly/tyFGx1Z>.
14. RiTdisplay RGS15128128WR000 General Features. Datasheet [Електронний ресурс] — Режим доступу: <https://cutt.ly/XyFGFaO>.

					<i>ДК62.469515.001 ПЗ</i>	<i>Арк.</i>
<i>Змн.</i>	<i>Арк.</i>	<i>№ докум.</i>	<i>Підпис</i>	<i>Дата</i>		74

15. OLED display with SSD1327 Datasheet [Электронный ресурс] — Режим доступа: <https://cutt.ly/MyFHvTw>.
16. Connector USB - A. Datasheet [Электронный ресурс] — Режим доступа: <https://cutt.ly/jyFGCTt>.
17. Connector USB – micro B. Datasheet [Электронный ресурс] — Режим доступа: <https://cutt.ly/pyFG0ID>.
18. Resonator 12SMX-B. Datasheet [Электронный ресурс] — Режим доступа: <https://cutt.ly/CyFH34K>.
19. Resonator 7A. Datasheet [Электронный ресурс] — Режим доступа: <https://cutt.ly/GyFJi4a>.
20. Resonator CX5032. Datasheet [Электронный ресурс] — Режим доступа: <https://cutt.ly/eyFJgmO>.
21. Resonator HC49. Datasheet [Электронный ресурс] — Режим доступа: <https://cutt.ly/pyFJISS>.
22. Interface – Controller MCP2200. Datasheet [Электронный ресурс] — Режим доступа: <https://cutt.ly/myFJB7W>.
23. Voltage regulator LP3981. Datasheet [Электронный ресурс] — Режим доступа: <https://cutt.ly/GyFJ56w>.
24. Voltage regulator TLV760. Datasheet [Электронный ресурс] — Режим доступа: <https://cutt.ly/vyGpk3P>.
25. Voltage regulator LD1117. Datasheet [Электронный ресурс] — Режим доступа: <https://cutt.ly/eyGpnj8>.
26. Button PB400. Datasheet [Электронный ресурс] — Режим доступа: <https://cutt.ly/QyGpWWt>.
27. Друкована плата [Электронный ресурс] — Режим доступа: <https://cutt.ly/JupqMpJ>.
28. Altium Designer [Электронный ресурс] — Режим доступа: <https://cutt.ly/yyJUopg>.
29. С. М. Боровиков. Расчёт показателей надёжности радиоэлектронных средств: учеб.-метод. пособие. – Минск: БГУИР, 2010. – 68 с.

					<i>ДК62.469515.001 ПЗ</i>	<i>Арк.</i>
<i>Эмн.</i>	<i>Арк.</i>	<i>№ докум.</i>	<i>Підпис</i>	<i>Дата</i>		<i>75</i>

30. Лескин В.Ф., Методические указания по конструкторско-технологическому и электрическому расчету печатных плат // В.Ф. Лескин, А.Н. Лысенко, Т.Л. Захарченко – Киев, НТУУ «КПИ», 2010.

31. Стеклотекстолит FR-4 [Электронный ресурс] — Режим доступа: <https://cutt.ly/vyMhwaz>.

32. Програма «Електроніка-2020» [Электронный ресурс] — Режим доступа: <https://cutt.ly/oy1GpUq>

ГОСТ 23751-86 ПЛАТЫ ПЕЧАТНЫЕ Основные параметры конструкции.

ГОСТ 23665-79. Платы печатные. Обработка контура. Требования к типовым технологическим процессам.

ГОСТ 23770-79. Платы печатные. Типовые технологические процессы химической и гальванической металлизации.

ГОСТ 2.417-91. Платы печатные. Правила выполнения чертежей.

ГОСТ 2.701-84 – ЕСКД. Правила выполнения схем.

ГОСТ 2.702-75 – ЕСКД. Правила выполнения электрических схем.

ГОСТ 2.708-81 – ЕСКД. Правила оформления схем.

ГОСТ 2.709-89 – ЕСКД. Обозначения условные проводов и контактных соединений электрических элементов, оборудования и участков цепей в электрических схемах.

ГОСТ 2.710-81 – ЕСКД. Обозначения буквенно-цифровые в электрических схемах.

ГОСТ 2.730-73 – УГО. Приборы полупроводниковые.

ГОСТ 2.743-91 – УГО. Элементы цифровой техники.

ГОСТ 2.759-82 – УГО. Элементы аналоговой техники.

ГОСТ 27.002-2015 Надежность в технике (ССНТ). Термины и определения.

					<i>ДК62.469515.001 ПЗ</i>	Арк.
Змн.	Арк.	№ докум.	Підпис	Дата		76

ГОСТ 26246.5-89 Материал электроизоляционный фольгированный нормированной горючести для печатных плат на основе стеклоткани, пропитанной эпоксидным связующим (FR-4).

ГОСТ 15150-69 Машины, приборы и другие технические изделия.

ГОСТ Р 50621-93 платы печатные одно-- и двусторонние с неметаллизированными отверстиями. Общие технические требования.

ГОСТ 2718-74 Гетинакс электротехнический листовой.

ГОСТ 2910-74 Текстолит электротехнический листовой.

ГОСТ 12652-74 Стеклотекстолит электротехнический листовой.

ОСТ 4.010.022-85 Платы печатные. Методы конструирования и расчета.

					<i>ДК62.469515.001 ПЗ</i>	<i>Арк.</i>
<i>Эмн.</i>	<i>Арк.</i>	<i>№ докум.</i>	<i>Підпис</i>	<i>Дата</i>		77

Технічне завдання на проектування

1. Найменування та галузь використання

Генератор QR-коду. Використовується з метою шифрування певної інформації для різних цілей.

2. Підстава для розробки

Підставою для розробки є завдання на дипломний проект згідно наказу по НТУУ «КПІ» № 1196-с від 25.05.2020р.

3. Мета і призначення розробки

Метою є створенням пристрою, який буде генерувати QR-код. Дана розробка призначена для ідентифікації товарів у торгівельному підприємстві.

4. Джерело розробки

Завдання на дипломний проект згідно наказу по НТУУ «КПІ ім. І. Сікорського»

5. Технічні вимоги

5.1. Склад виробу й вимоги до пристрою, що розробляється.

- Пристрій повинен забезпечувати:
- Виведення результату генерування QR-коду (дисплей);
- Можливість підключення джерела живлення;
- Можливість підключення клавіатури.

5.2. Вимоги до надійності.

Середній час напрацювання на відмову повинен бути не менше 10000 годин.

5.3. Вимоги до технологічності.

Орієнтовані на передові прийоми розробки виготовлення на підприємствах України.

5.4. Вимоги до рівня уніфікації й стандартизації.

Для виготовлення пристрою застосувати стандартні, уніфіковані деталі та вироби.

5.5. Вимоги безпеки обслуговування.

					<i>ДК62.469515.001 ПЗ</i>	Арк. 78
Змн.	Арк.	№ докум.	Підпис	Дата		

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

5.6. Вимоги до складових частин виробу, сировини, вихідних й експлуатаційних матеріалів.

Для виробництва пристрою повинні використовуватися матеріали імпортного виробництва, які можливо купити та замінити в Україні, або власна елементна база, яка розробляється в Україні.

5.7. Умови експлуатації.

Кліматичне виконання УХЛ.5.1 згідно ГОСТ 15150-69. Для експлуатації в лабораторних, капітальних житлових та інших подібного типу приміщеннях.

5.8. Вимоги до транспортування і зберігання.

Група умов зберігання Л1 по ГОСТ 15150-69. Зберігати в зачинених, опалювальних та вентиляованих приміщеннях, в яких забезпечуються наступні умови: температура повітря +5...+40°C, відносна вологість повітря 90% при 15°C (середньорічне значення), атмосферний тиск 84...106кПа. Транспортувати автомобільним, залізничним або авіаційним видами транспорту в спеціальній транспортній тарі.

					<i>ДК62.469515.001 ПЗ</i>	Арк.
Змн.	Арк.	№ докум.	Підпис	Дата		79

5.9. Додаткові технічні вимоги.

Технічні характеристики:

Габаритні розміри	80x80x20 мм.
Маса	Не більше 200 г.
Напруга живлення	3,3 ± 5% В 5 ± 5% В

6. Результати роботи

6.1. Результати даної роботи можуть бути використані як вихідна документація по створенню прототипу пристрою, його програмування, налагодження;

6.2. Дана робота (звітна документація) після виконання надається на кафедру КЕОА для подальшого захисту й зберігання як навчальної документації.

7. Робота повинна містити в собі документи

- Пояснювальну записку (формату А4, до 80 аркушів)
- Схеми електричні принципові та переліки елементів (формату А1 та А4 відповідно)
- Складальні креслення та специфікації (формату А1 та А4 відповідно)
- Креслення друкованих плат (формату А1)
- Додатки (формату А4)

8. Порядок розгляду й приймання роботи

Порядок розгляду й приймання роботи на загальних умовах, прийнятих на кафедрі КЕОА. Рецензування й прийняття роботи комісією на загальних умовах. У процесі виконання роботи проміжні звіти надаються комісії не рідше 1 раз у тиждень на загальних умовах.

9. Економічні показники

В умовах даного проекту не розглядаються.

					ДК62.469515.001 ПЗ	Арк.
Змн.	Арк.	№ докум.	Підпис	Дата		80

Додаток Б

Лістинг програми

main.c

```
#include <stdio.h>
#include <stdbool.h>
#include "main.h"
#include "DEV_Config.h"
#include "OLED_Driver.h"
#include "OLED_GUI.h"
#include "QR_prog.h"
SPI_HandleTypeDef hspi1;
DMA_HandleTypeDef hdma_spi1_tx;
UART_HandleTypeDef huart2;
void SystemClock_Config(void);
static void MX_GPIO_Init(void);
static void MX_SPI1_Init(void);
static void MX_DMA_Init(void);
static void MX_USART2_UART_Init(void);
int main(void)
{
    HAL_Init();
    SystemClock_Config();
    MX_GPIO_Init();
    MX_SPI1_Init();
    MX_DMA_Init();
    MX_USART2_UART_Init();
    OLED_Init(SCAN_DIR_DFT); //SCAN_DIR_DFT = D2U_L2R
    GUI_Show();
    OLED_Clear(BLACK);
    GUI_DisString_EN(0, 48, "Pls writing code", &Font12, BLACK, WHITE);
    OLED_Display();
    uint8_t str[1];
    char new_str[14];
    int cnt_rec = 0;
    int state = 0;
    char str_size[9];
    while(1) {
        HAL_UART_Receive(&huart2, str, 1, 0xFFFFFFFF);
        HAL_UART_Transmit(&huart2, str, 1, 0xFFFFFFFF);
        OLED_Clear(BLACK);
        if(state == 0) {
            if((char)str[0] == 13) {
                state = 1;
            } else if((char)str[0] == 8) {
                if(0 < cnt_rec){
                    cnt_rec--;
                    new_str[cnt_rec] = NULL;
                }
            } else {
                if(32 <= (char)str[0] && (char)str[0] <= 126) {
                    new_str[cnt_rec] = str[0];
                    cnt_rec++;
                }
                if(cnt_rec > 14) {
                    cnt_rec = 0;
                    for(int i = 0; i < 14; i++) {
                        new_str[i] = NULL;
                    }
                }
            }
        }
    }
}
```

										Арк.
										81
Змн.	Арк.	№ докум.	Підпис	Дата	ДК62.469515.001 ПЗ					

```

    }
    } else if(state == 1) {
        if((char)str[0] == 13) {
            state = 0;
        }
    }
    if(state == 0) {
        GUI_DisString_EN(0 , 0, new_str, &Font12, BLACK, WHITE);
        sprintf(str_size, "size = %d", cnt_rec);
        GUI_DisString_EN(0 , 20, str_size, &Font12, BLACK, WHITE);
    } else if(state == 1) {
        gen_QR(new_str, cnt_rec);
        draw_QR(10,10);
    }
    OLED_Display();
}
}
void SystemClock_Config(void)
{
    RCC_OscInitTypeDef RCC_OscInitStruct = {0};
    RCC_ClkInitTypeDef RCC_ClkInitStruct = {0};
    // Configure the main internal regulator output voltage
    __HAL_RCC_PWR_CLK_ENABLE();
    __HAL_PWR_VOLTAGESCALING_CONFIG(PWR_REGULATOR_VOLTAGE_SCALE1);

    // Initializes the CPU, AHB and APB busses clocks
    RCC_OscInitStruct.OscillatorType = RCC_OSCILLATORTYPE_HSE;
    RCC_OscInitStruct.HSEState = RCC_HSE_ON;
    RCC_OscInitStruct.PLL.PLLState = RCC_PLL_ON;
    RCC_OscInitStruct.PLL.PLLSource = RCC_PLLSOURCE_HSE;
    RCC_OscInitStruct.PLL.PLLM = 4;
    RCC_OscInitStruct.PLL.PLLN = 72;
    RCC_OscInitStruct.PLL.PLLP = RCC_PLLP_DIV2;
    RCC_OscInitStruct.PLL.PLLQ = 4;
    HAL_RCC_OscConfig(&RCC_OscInitStruct);
    // Initializes the CPU, AHB and APB busses clocks
    RCC_ClkInitStruct.ClockType = RCC_CLOCKTYPE_HCLK|RCC_CLOCKTYPE_SYCLK
|RCC_CLOCKTYPE_PCLK1|RCC_CLOCKTYPE_PCLK2;
    RCC_ClkInitStruct.SYSCLKSource = RCC_SYSCLKSOURCE_PLLCLK;
    RCC_ClkInitStruct.AHBCLKDivider = RCC_SYSCLK_DIV1;
    RCC_ClkInitStruct.APB1CLKDivider = RCC_HCLK_DIV2;
    RCC_ClkInitStruct.APB2CLKDivider = RCC_HCLK_DIV1;
    HAL_RCC_ClockConfig(&RCC_ClkInitStruct, FLASH_LATENCY_2);
}
static void MX_SPI1_Init(void)
{
    hspi1.Instance = SPI1;
    hspi1.Init.Mode = SPI_MODE_MASTER;
    hspi1.Init.Direction = SPI_DIRECTION_2LINES;
    hspi1.Init.DataSize = SPI_DATASIZE_8BIT;
    hspi1.Init.CLKPolarity = SPI_POLARITY_LOW;
    hspi1.Init.CLKPhase = SPI_PHASE_1EDGE;
    hspi1.Init.NSS = SPI_NSS_SOFT;
    hspi1.Init.BaudRatePrescaler = SPI_BAUDRATEPRESCALER_2;
    hspi1.Init.FirstBit = SPI_FIRSTBIT_MSB;
    hspi1.Init.TIMode = SPI_TIMODE_DISABLE;
    hspi1.Init.CRCCalculation = SPI_CRCCALCULATION_DISABLE;
    hspi1.Init.CRCPolynomial = 10;
    HAL_SPI_Init(&hspi1);
}

```

					<i>ДК62.469515.001 ПЗ</i>	<i>Арк.</i>
<i>Эмн.</i>	<i>Арк.</i>	<i>№ докум.</i>	<i>Подпис</i>	<i>Дата</i>		<i>82</i>

```

}
static void MX_USART2_UART_Init(void)
{
    huart2.Instance = USART2;
    huart2.Init.BaudRate = 115200;
    huart2.Init.WordLength = UART_WORDLENGTH_8B;
    huart2.Init.StopBits = UART_STOPBITS_1;
    huart2.Init.Parity = UART_PARITY_NONE;
    huart2.Init.Mode = UART_MODE_TX_RX;
    huart2.Init.HwFlowCtl = UART_HWCONTROL_NONE;
    huart2.Init.OverSampling = UART_OVERSAMPLING_16;
    HAL_UART_Init(&huart2);
}
static void MX_DMA_Init(void)
{
    //DMA controller clock enable
    __HAL_RCC_DMA2_CLK_ENABLE();
    /* DMA interrupt init DMA2_Stream2_IRQn interrupt configuration */
    HAL_NVIC_SetPriority(DMA2_Stream2_IRQn, 0, 0);
    HAL_NVIC_EnableIRQ(DMA2_Stream2_IRQn);
}
static void MX_GPIO_Init(void)
{
    GPIO_InitTypeDef GPIO_InitStruct = {0};
    /* GPIO Ports Clock Enable */
    __HAL_RCC_GPIOH_CLK_ENABLE();
    __HAL_RCC_GPIOA_CLK_ENABLE();
    __HAL_RCC_GPIOB_CLK_ENABLE();
    /*Configure GPIO pin Output Level */
    HAL_GPIO_WritePin(GPIOA, OLED_DC_Pin|OLED_RST_Pin, GPIO_PIN_RESET);
    /*Configure GPIO pin Output Level */
    HAL_GPIO_WritePin(OLED_CS_GPIO_Port, OLED_CS_Pin, GPIO_PIN_RESET);
    /*Configure GPIO pins : OLED_DC_Pin OLED_RST_Pin */
    GPIO_InitStruct.Pin = OLED_DC_Pin|OLED_RST_Pin;
    GPIO_InitStruct.Mode = GPIO_MODE_OUTPUT_PP;
    GPIO_InitStruct.Pull = GPIO_NOPULL;
    GPIO_InitStruct.Speed = GPIO_SPEED_FREQ_LOW;
    HAL_GPIO_Init(GPIOA, &GPIO_InitStruct);
    /*Configure GPIO pin : OLED_CS_Pin */
    GPIO_InitStruct.Pin = OLED_CS_Pin;
    GPIO_InitStruct.Mode = GPIO_MODE_OUTPUT_PP;
    GPIO_InitStruct.Pull = GPIO_NOPULL;
    GPIO_InitStruct.Speed = GPIO_SPEED_FREQ_LOW;
    HAL_GPIO_Init(OLED_CS_GPIO_Port, &GPIO_InitStruct);
}
void Error_Handler(void)
{
    /* USER CODE BEGIN Error_Handler_Debug */
    /* User can add his own implementation to report the HAL error return state */
    /* USER CODE END Error_Handler_Debug */
}

```

QR_prog.c

```

#include <stdio.h>
#include <stdbool.h>
#include <string.h>
#include <math.h>
#include "OLED_GUI.h"
#include "DEV_Config.h"

```

					<i>ДК62.469515.001 ПЗ</i>	<i>Арк.</i>
<i>Эмн.</i>	<i>Арк.</i>	<i>№ докум.</i>	<i>Підпис</i>	<i>Дата</i>		<i>83</i>

```

static int matr[21][21] = {
    1,1,1,1,1,1,1,0,1,0,0,0,0,0,1,1,1,1,1,1,1,
    1,0,0,0,0,0,0,1,0,0,0,0,0,0,0,1,0,0,0,0,0,1,
    1,0,1,1,1,0,1,0,1,0,0,0,0,0,0,1,0,1,1,1,0,1,
    1,0,1,1,1,0,1,0,0,0,0,0,0,0,0,1,0,1,1,1,0,1,
    1,0,1,1,1,0,1,0,0,0,0,0,0,0,0,1,0,1,1,1,0,1,
    1,0,0,0,0,0,1,0,1,0,0,0,0,0,0,1,0,0,0,0,0,1,
    1,1,1,1,1,1,1,0,1,0,1,0,1,0,1,1,1,1,1,1,1,
    0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,
    1,0,1,0,0,0,1,1,0,0,0,0,0,0,0,0,1,0,0,1,0,1,
    0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,
    0,0,0,0,0,0,1,0,0,0,0,0,0,0,0,0,0,0,0,0,0,
    0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,
    0,0,1,1,1,0,1,0,0,0,0,0,0,0,0,0,0,0,0,0,0,
    0,0,0,0,0,0,1,0,0,0,1,0,0,0,0,0,0,0,0,0,0,
    1,1,1,1,1,1,1,0,1,0,0,0,0,0,0,0,0,0,0,0,0,
    1,0,0,0,0,0,0,1,0,0,0,0,0,0,0,0,0,0,0,0,0,
    1,0,1,1,1,0,1,0,0,0,0,0,0,0,0,0,0,0,0,0,0,
    1,0,1,1,1,0,1,0,0,0,0,0,0,0,0,0,0,0,0,0,0,
    1,0,1,1,1,0,1,0,0,0,0,0,0,0,0,0,0,0,0,0,0,
    1,0,0,0,0,0,1,0,0,0,0,0,0,0,0,0,0,0,0,0,0,
    1,1,1,1,1,1,1,0,1,0,0,0,0,0,0,0,0,0,0,0,0};
int coord_x[208] = {
    0x14,0x13,0x14,0x13,0x14,0x13,0x14,0x13,0x14,0x13,0x14,0x13,0x14,0x13,0x14,0x1
3,
    0x14,0x13,0x14,0x13,0x14,0x13,0x14,0x13,0x12,0x11,0x12,0x11,0x12,0x11,0x12,0x1
1,
    0x12,0x11,0x12,0x11,0x12,0x11,0x12,0x11,0x12,0x11,0x12,0x11,0x12,0x11,0x12,0x1
1,
    0x10,0x0f,0x10,0x0f,0x10,0x0f,0x10,0x0f,0x10,0x0f,0x10,0x0f,0x10,0x0f,0x10,0x0
f,
    0x10,0x0f,0x10,0x0f,0x10,0x0f,0x10,0x0f,0x0e,0x0d,0x0e,0x0d,0x0e,0x0d,0x0e,0x0
d,
    0x0e,0x0d,0x0e,0x0d,0x0e,0x0d,0x0e,0x0d,0x0e,0x0d,0x0e,0x0d,0x0e,0x0d,0x0e,0x0
d,
    0x0c,0x0b,0x0c,0x0b,0x0c,0x0b,0x0c,0x0b,0x0c,0x0b,0x0c,0x0b,0x0c,0x0b,0x0c,0x0
b,
    0x0c,0x0b,0x0c,0x0b,0x0c,0x0b,0x0c,0x0b,0x0c,0x0b,0x0c,0x0b,0x0c,0x0b,0x0c,0x0
b,
    0x0c,0x0b,0x0c,0x0b,0x0c,0x0b,0x0c,0x0b,0x0a,0x09,0x0a,0x09,0x0a,0x09,0x0a,0x0
9,
    0x0a,0x09,0x0a,0x09,0x0a,0x09,0x0a,0x09,0x0a,0x09,0x0a,0x09,0x0a,0x09,0x0a,0x0
9,
    0x0a,0x09,0x0a,0x09,0x0a,0x09,0x0a,0x09,0x0a,0x09,0x0a,0x09,0x0a,0x09,0x0a,0x0
9,
    0x08,0x07,0x08,0x07,0x08,0x07,0x08,0x07,0x05,0x04,0x05,0x04,0x05,0x04,0x05,0x0
4,
    0x03,0x02,0x03,0x02,0x03,0x02,0x03,0x02,0x01,0x00,0x01,0x00,0x01,0x00,0x01,0x0
0};
int coord_y[208] = {
    0x14,0x14,0x13,0x13,0x12,0x12,0x11,0x11,0x10,0x10,0x0f,0x0f,0x0e,0x0e,0x0d,0x0
d,
    0x0c,0x0c,0x0b,0x0b,0x0a,0x0a,0x09,0x09,0x09,0x09,0x0a,0x0a,0x0b,0x0b,0x0c,0x0
c,
    0x0d,0x0d,0x0e,0x0e,0x0f,0x0f,0x10,0x10,0x11,0x11,0x12,0x12,0x13,0x13,0x14,0x1
4,
    0x14,0x14,0x13,0x13,0x12,0x12,0x11,0x11,0x10,0x10,0x0f,0x0f,0x0e,0x0e,0x0d,0x0
d,
    0x0c,0x0c,0x0b,0x0b,0x0a,0x0a,0x09,0x09,0x09,0x09,0x0a,0x0a,0x0b,0x0b,0x0c,0x0
c,

```

```

0xd,0xd,0xe,0xe,0xf,0xf,0x10,0x10,0x11,0x11,0x12,0x12,0x13,0x13,0x14,0x1
4,
0x14,0x14,0x13,0x13,0x12,0x12,0x11,0x11,0x10,0x10,0xf,0xf,0xe,0xe,0xd,0x0
d,
0xc,0xc,0xb,0xb,0xa,0xa,0x9,0x9,0x8,0x8,0x7,0x7,0x5,0x5,0x4,0x0
4,
0x3,0x3,0x2,0x2,0x1,0x1,0x0,0x0,0x0,0x0,0x0,0x0,0x1,0x1,0x2,0x2,0x3,0x0
3,
0x4,0x4,0x5,0x5,0x7,0x7,0x8,0x8,0x9,0x9,0xa,0xa,0xb,0xb,0xc,0x0
c,
0xd,0xd,0xe,0xe,0xf,0xf,0x10,0x10,0x11,0x11,0x12,0x12,0x13,0x13,0x14,0x1
4,
0xc,0xc,0xb,0xb,0xa,0xa,0x9,0x9,0x9,0x9,0xa,0xa,0xb,0xb,0xc,0x0
c,
0xc,0xc,0xb,0xb,0xa,0xa,0x9,0x9,0x9,0x9,0xa,0xa,0xb,0xb,0xc,0x0
c};
int mem_galua[256];
int mem_inv_galua[256];
bool main_code[16 * 8];
bool correct_code[10 * 8];
void generate_gf()
{
    int p = 29;
    int msk = 1;
    for (int i = 0; i < 8; i = i + 1){
        mem_galua[i] = msk;
        mem_inv_galua[mem_galua[i]] = i;
        msk = msk << 1;
    }
    mem_galua[8] = p;
    mem_inv_galua[mem_galua[8]] = 8;
    msk = msk >> 1;
    for (int i = 9; i < 255; i = i + 1){
        if (mem_galua[i - 1] >= msk){
            mem_galua[i] = mem_galua[8] ^ ((mem_galua[i-1] ^ msk) << 1);
        } else {
            mem_galua[i] = mem_galua[i-1] << 1;
        }
        mem_inv_galua[mem_galua[i]] = i;
    }
    mem_inv_galua[0] = -1;
    mem_galua[255] = 0;
}
bool DecToBin(int data, int num)
{
    bool bin_code[32] = {0, 0, 0, 0};
    for(int i = 31; i >= 0; i--){
        if((data & (1 << i)) != 0){
            bin_code[i] = 1;
        }else{
            bin_code[i] = 0;
        }
    }
    return bin_code[num];
}
void gen_main(char data[], int size)
{
    for(int i = 0; i < 16 * 8; i++){
        main_code[i] = 0;
    }
    main_code[1] = 1;
}

```

					<i>ДК62.469515.001 ПЗ</i>	Арк.
Эмн.	Арк.	№ докум.	Підпис	Дата		85

```

for(int i = 7; i >= 0; i--){
    main_code[11 - i] = DecToBin(size, i);
}
for(int n = 0; n < size; n++){
    for(int i = 7; i >= 0; i--){
        main_code[11 + (n + 1) * 8 - i] = DecToBin((int)data[n], i);
    }
}
bool num = 0;
for(int n = size + 2; n < 16; n++, num = !num){
    for(int i = 0; i < 8; i++){
        if(num) {
            main_code[(n + 1) * 8 - 1 - i] = DecToBin(17, i);
        } else {
            main_code[(n + 1) * 8 - 1 - i] = DecToBin(236, i);
        }
    }
}
}
void gen_correct(int size)
{
    generate_gf();
    int m_code[16];
    int g_code[10]; // = {251, 67, 46, 61, 118, 70, 64, 94, 32, 45};
    int gen_code[10] = {251, 67, 46, 61, 118, 70, 64, 94, 32, 45};
    for(int n = 0; n < 16; n++){
        m_code[n] = 0;
        for(int i = 7; i >= 0; i--){
            m_code[n] += main_code[(n + 1) * 8 - 1 - i] * pow(2, i);
        }
    }
    for(int n = 0; n < 10; n++){
        g_code[n] = m_code[n];
    }
    int array_A, array_B, tmp;
    for(int n = 0; n < 16; n++){
        array_A = mem_inv_galua[g_code[0]];
        tmp = 0;
        m_code[0] = 0;
        for(int i = 0; i < 15; i++){
            tmp = m_code[i];
            m_code[i] = m_code[i + 1];
            m_code[i + 1] = tmp;
        }
        for(int i = 0; i < 10; i++){
            array_B = (gen_code[i] + array_A) % 255;
            g_code[i] = mem_galua[array_B] ^ m_code[i];
        }
        for(int n = 0; n < 10; n++){
            m_code[n] = g_code[n];
        }
    }
    for(int i = 0; i < 10; i++){
        for(int j = 7; j >= 0; j--){
            correct_code[(i + 1) * 8 - 1 - j] = DecToBin(g_code[i], j);
        }
    }
}
bool mask(int x, int y, bool data)
{
    if(y % 2 == 0) {

```

					<i>ДК62.469515.001 ПЗ</i>	<i>Арк.</i>
<i>Эмн.</i>	<i>Арк.</i>	<i>№ докум.</i>	<i>Підпис</i>	<i>Дата</i>		<i>86</i>

```

        return !data;
    } else {
        return data;
    }
}
void gen_QR(char data[], int size)
{
    gen_main(data, size);
    gen_correct(size);
    for(int i = 0; i < 208; i++){
        if(i < 16 * 8) {
            matr[coord_y[i]][coord_x[i]] = mask(coord_x[i], coord_y[i],
main_code[i]);
        } else {
            matr[coord_y[i]][coord_x[i]] = mask(coord_x[i], coord_y[i],
correct_code[i - 16 * 8]);
        }
    }
}
void draw_QR(int x_start, int y_start)
{
    GUI_DrawLine(x_start, y_start, x_start + 22 * 5, y_start, WHITE, LINE_SOLID,
DOT_PIXEL_3X3);
    GUI_DrawLine(x_start, y_start, x_start, y_start + 22 * 5, WHITE, LINE_SOLID,
DOT_PIXEL_3X3);
    GUI_DrawLine(x_start + 22 * 5, y_start, x_start + 22 * 5, y_start + 22 * 5,
WHITE, LINE_SOLID, DOT_PIXEL_3X3);
    GUI_DrawLine(x_start, y_start + 22 * 5, x_start + 22 * 5, y_start + 22 * 5,
WHITE, LINE_SOLID, DOT_PIXEL_3X3);
    for(int y = 0; y < 21; y++){
        for(int x = 0; x < 21; x++){
            if(matr[y][x] == 0) {
                GUI_DrawPoint(x_start + 5 * (x + 1), y_start + 5 * (y + 1),
WHITE, DOT_PIXEL_3X3, DOT_FILL_AROUND);
            }
        }
    }
}
}

```

					<i>ДК62.469515.001 ПЗ</i>	<i>Арк.</i>
<i>Эмн.</i>	<i>Арк.</i>	<i>№ докум.</i>	<i>Підпис</i>	<i>Дата</i>		87

Додаток В — Матеріали конференції



УКРАЇНА

МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ

**НАЦІОНАЛЬНИЙ ТЕХНІЧНИЙ УНІВЕРСИТЕТ УКРАЇНИ
“КИЇВСЬКИЙ ПОЛІТЕХНІЧНИЙ ІНСТИТУТ
імені ІГОРЯ СІКОРСЬКОГО”**

Факультет електроніки
Комунальний позашкільний навчальний заклад
«Київська Мала академія наук учнівської молоді»
Громадська організація «Пані Наука»

ПРОГРАМА

**XIII МІЖНАРОДНА НАУКОВО-ТЕХНІЧНА
КОНФЕРЕНЦІЯ МОЛОДИХ ВЧЕНИХ
«ЕЛЕКТРОНІКА-2020»**

15-16 квітня 2020 року
Україна, м. Київ

					<i>ДК62.469515.001 ПЗ</i>	Арк.
<i>Змн.</i>	<i>Арк.</i>	<i>№ докум.</i>	<i>Підпис</i>	<i>Дата</i>		88

Генератор QR-коду на основі FPGA

Сокол[†] Я. В., ORCID [0000-0002-9104-6690](https://orcid.org/0000-0002-9104-6690)

Варфоломеев[§] А. Ю., к.т.н., ORCID [0000-0002-6990-7140](https://orcid.org/0000-0002-6990-7140)

Національний технічний університет України
«Київський політехнічний інститут імені Ігоря Сікорського»
Київ, Україна

DOI: [10.20535/2523-4445.v.3.n1.111](https://doi.org/10.20535/2523-4445.v.3.n1.111)

Анотація—Розроблено апаратне IP-ядро для генерування QR-кодів на ПЛІС, що підтримує першу версію кодів з корекцією помилок M типу. На основі розробленого IP-ядра, на платі налагодження DE1-SoC створено макетний зразок пристрою для генерування QR-кодів. Значення пристрій дозволяє за допомогою клавіатури з інтерфейсом PS/2 вводити текстові дані, які мають бути зашифровані у QR-коді та виводити результат (згенерований код) на VGA монітор у вигляді зображення. Для створення такого пристрою використовується модуль керування, IP-блок для генерування QR-кодів та модуль введення-виведення.

Ключові слова — QR-код; FPGA; VGA; клавіатура.

I. Вступ

QR-код (Quick Response) є двомірним матричним штрих-кодом, який призначений для зберігання невеликих обсягів інформації, головним чином на друкованих носіях, яку можна швидко та зручно зчитати і розпізнати як за допомогою спеціальних сканерів, так і камерою мобільного телефону. В загальному випадку у QR-коді можна закодувати будь-яку інформацію, проте найбільш часто у ньому зберігають текстову інформацію, таку як номери телефонів, посилання на сайти, геометки, номери криптовалютних гаманців тощо [1, 2].

Не дивлячись на існування альтернативних версій двомірних штрих-кодів, таких як Aztec code, Data Matrix, саме QR-коди наразі набули найбільшого розповсюдження, тому в цій роботі мова йдеться саме про них.

Створення QR-коду може здійснюватись програмно — із використанням мікропроцесора чи мікроконтролера або апаратно — за допомогою спеціалізованого пристрою. Кожен з цих підходів має свої переваги та недоліки. Так, програмні реалізації є більш гнучкими і простими, водночас, будучи розгорнутими на неспецифічних контролерах чи процесорах, вони потребують більшого енергоспоживання [3]. Апаратні реалізації навпаки є більш складними, не володіють значною гнучкістю, проте можуть бути більш швидкодіючими та споживати значно менше енергії, що робить їх використання більш доцільним у різноманітних портативних пристроях з батарейним живленням. При цьому, щоб спростити створення апаратного модуля для генерування QR-коду доцільно скористатися технологією ПЛІС.

Таким чином, мета даної роботи полягає в розробці вбудованого рішення для генерування QR-кодів на ПЛІС. Для цього було розроблено

спеціальний IP-блок на базі плати налагодження DE1-SoC.

II. ОСОБЛИВОСТІ РЕАЛІЗАЦІЇ

IP-блок для генерування QR-кодів пропонується реалізувати на доступній та поширеній платі налагодження DE1-SoC, що містить FPGA мікросхему Cyclone V. Дана плата має інтерфейси для підключення клавіатури та VGA — монітору, тому введення та виведення інформації про QR-код доцільно реалізувати через зазначені периферійні пристрої. Пропонується наступний принцип роботи розроблюваного пристрою:

- користувач вводить необхідний текст з клавіатури; даний текст паралельно відображається на моніторі;
- при натисканні на кнопку “Enter” виконується генерування QR-коду та його виведення у вигляді зображення на монітор;
- за необхідності повторне введення коду здійснюється при натисканні на клавішу “Esc”.

Загальна структурна схема пристрою та структурна схема IP-блоку генерування QR-кодів представлені на Рис. 1 а) та б) відповідно.

IP-блок для генерування QR-кодів реалізовано на мові Verilog [4] і складається з наступних модулів: `control`, `keyboard`, `mem_gen_code`, `mem_main_code`, `mem_mat_code`, `VGA`. Для спрощення алгоритму використовується обмеження, які полягають в наступному:

- Генерування здійснюється лише для першої версії коду, яка має розмір матриці 21x21 піксель.



Copyright (c) 2020 Сокол Я.В.

									Арк.
									89
Змн.	Арк.	№ докум.	Підпис	Дата	ДК62.469515.001 ПЗ				

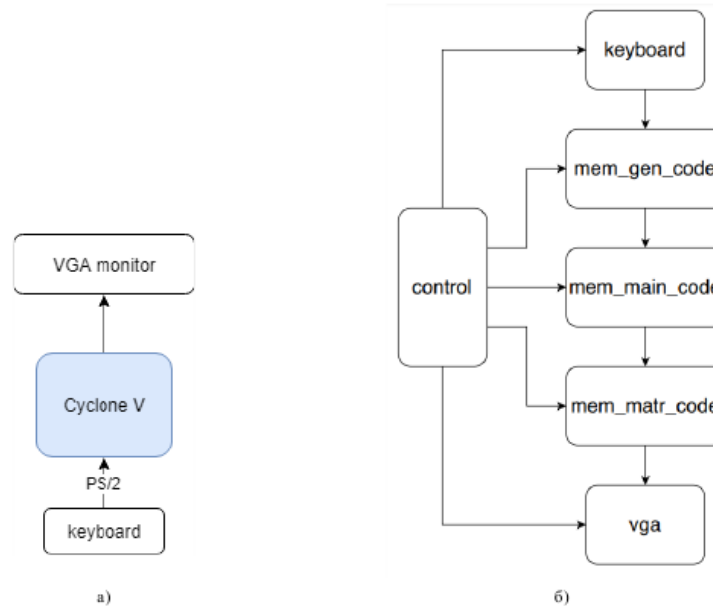


Рис. 1 Структурна схема пристрою та IP-блоку генерування QR-кодів: а) загальна структурна схема пристрою; б) Структурна схема IP-блоку генерування QR-кодів

- Застосовується рівень корекції помилок М типу. Даний тип корекції передбачає використання у кодї надлишкової інформації, що дозволяє відновити вміст закодованих даних у випадку пошкодження до 15% зображення коду (такий тип є найбільш розповсюдженим).

Виходячи з даних обмежень, код може шифрувати не більше ніж 14 символів.

III. ОПИС РЕАЛІЗАЦІ АПАРАТНИХ МОДУЛІВ ПРИСТРОЮ ГЕНЕРУВАННЯ QR-КОДІВ

Апаратна реалізація пристрою для генерування QR-код на платі налагодження DE1-SoC включає модуль керування, IP-блок для генерування коду та модулі введення-виведення. Нижче наведемо їх опис на високому рівні.

A. Модуль керування

Керування пристроєм здійснюється за допомогою модуля **control**, який є скінченним автоматом зі станами, які відповідають наступним операціям:

- скидання, очищення пам'яті;
- запис коду з клавіатури;
- створення основної частини QR-коду;
- створення байтів корекції використовуючи на основі даних основної частини QR-коду;
- об'єднання байтів корекції з основною частиною QR-коду та заповнення їх у матрицю 21x21 піксель;

- виведення коду на екран;
- очікування кнопки Esc

B. IP-блок для генерування QR-кодів

Блок складається з трьох модулів **mem_main_code**, **mem_gen_code** та **mem_matr_code**. Наведемо більш детальний їх опис.

Модуль **mem_main_code** приймає на вхід байти тексту у форматі ASCII [5] і генерує первинну послідовність даних, що мають бути зашифровані у кодї. Ця послідовність містить наступні поля:

- 1) Перші 4 біти зберігають тип кодування. В даній роботі використовується лише побайтове кодування, тому дане поле містить послідовність 0100. Загалом існують ще цифрове та літерно-числове кодування, для яких перше поле має заповнюватись бітами 0001 та 0010 відповідно.
- 2) Наступні 2 байти зберігають обсяг інформації, що зашифрована в кодї — це звичайне двійкове число.
- 3) Наступні N байт (кількість залежить від типу корекції та номеру версії) зберігають корисну інформацію (власне дані). В даній роботі використовується до 14 байтів даних. Невикористані байти заповнюються наступними константами по черзі: 11101100 та 00010001.



Copyright (c) 2020 Прізвище І. П., Прізвище І. П.

						Арк. 90
Змн.	Арк.	№ докум.	Підпис	Дата	ДК62.469515.001 ПЗ	

- 4) Останні 4 біти заповнюються нулями для того, щоб послідовність містила цілу кількість байт.

Модуль `mem_gen_code` виконується для генерування байтів корекції. Для цього виконується алгоритм Ріда-Соломона [6].

Для застосування алгоритму Ріда-Соломона необхідно для початку підготувати масив байтів корекції. Даний масив обирається за рівнем корекції та номером версії коду. Оскільки використовується рівень корекції M та перший номер версії, то масив байтів корекції містить наступні 10 елементів: {251, 67, 46, 61, 118, 70, 64, 94, 32, 45}. Для даного алгоритму також необхідні таблиці прямого та зворотного полів Галуа, які зберігають значення від 0 до 255 [7]. Сам алгоритм передбачає циклічне виконання наступні дії (кількість ітерацій циклу визначається кількістю байт даних, що мають бути закодовані) [6]:

- 5) перший елемент масиву даних зберігається у змінній a і видаляється з масиву (всі наступні значення зсуваються на одну клітинку вліво, останній елемент заповнюється нулем);
- 6) якщо a дорівнює нулю, то пропустити наступні дії і перейти до наступної ітерації циклу;
- 7) у таблиці зворотного поля Галуа визначається значення комірки, що відповідає числу a і її значення заноситься у змінну b : $b = GF^{-1}[a]$;
- 8) далі для N перших елементів, де N — кількість байтів корекції, i — лічильник циклу;
- 9) до i -го значення генеруючого многочлена додається значення b і зберігається у змінній c ;
- 10) якщо b більше 254, використовується залишок від ділення b на 255;
- 11) знайти відповідне c значення в таблиці поля Галуа і виконати побітову операцію сума по модулю 2 (XOR) з i -м значенням підготованого масиву і записати отримане значення в i -у комірку підготовленого масиву.

Модуль `mem_matr_code` виконує генерування самої матриці, розміщуючи узори для вирівнювання, код маски та рівень корекції та смуги синхронізації відповідно з Рис. 2 [8].

Дана інформація містить наступне [6]:

- Узори для вирівнювання завжди розташовуються в 3 краях (верхній лівий кут, верхній правий кут та нижній лівий кут). Матриця коду також може містити додаткові

узори для вирівнювання, що розміщуються через кожну 21 комірку матриці по висоті та ширині. Додаткові узори наявні у версіях QR-кодів вище 2-ї, тому в цій роботі не застосовуються.

- Тип маски та рівень корекції задаються 15-бітним числом. Оскільки в даній роботі використовується рівень корекції типу M та тип маски під номером 1, то код типу маски та рівня корекції містить бітову послідовність наступного вигляду 101000100100101.
- Полоси синхронізації розміщуються між узорами для вирівнювання і почергово заповнюються нулями та одиницями.
- Номер версії коду складається з 15 бітів, обирається згідно номеру версії QR-коду і використовується в 7 версії та більше.

Далі після цього виконується заповнення байтів з певною послідовністю, які були згенеровані за допомогою блоку `mem_main_code`, відповідно до Рис. 3 [6]. Після такого розташування для цих даних накладається маска, відповідно до Рис. 4 [9] (використовується маска 1-го типу, її код 001).



Рис. 2 Розміщення службової інформації на QR-коді

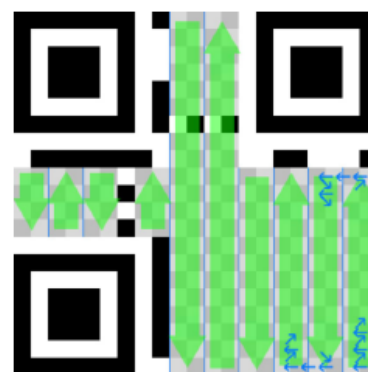


Рис. 3 Розміщення даних на QR-коді



BY

Copyright (c) 2020 Сокол Я.В.

Змн.	Арк.	№ докум.	Підпис	Дата

ДК62.469515.001 ПЗ

Арк.

91

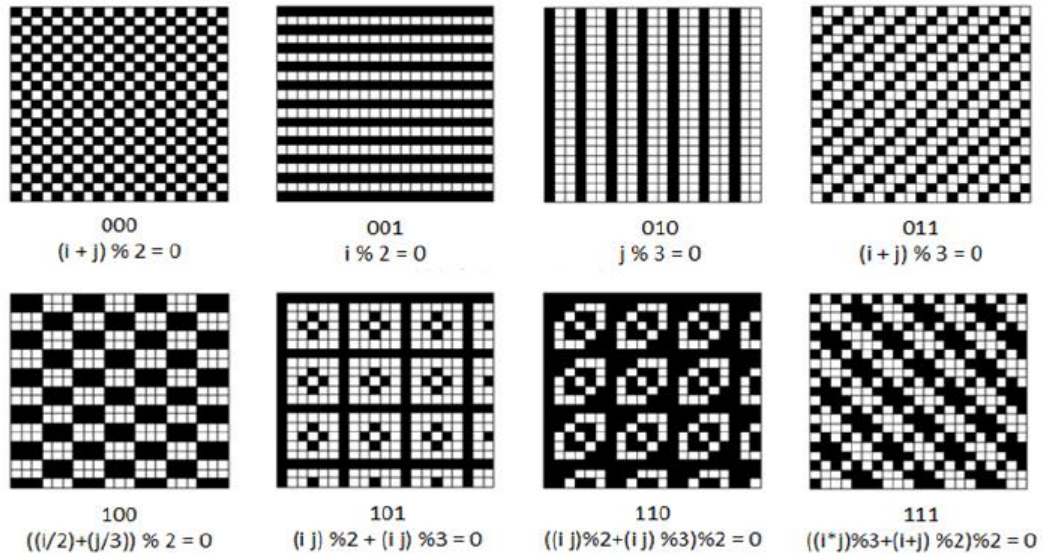


Рис. 4 Типи масок: операція «%» позначає остачу від ділення

С. Модулі введення-виведення

Модуль keyboard виконує зчитування сигналів клавіатури (сигналу синхронізації та сигналу даних) і перетворення сигналу даних на ASCII-коди. Для цього по кожному задньому фронту сигналу синхронізації виконується реєстрація значень сигналу даних, які далі записуються у 8-бітну шину. Оскільки клавіатура може мати «брязкіт» контактів на клавішах, в даному модулі додатково виконується фільтрація прийнятих даних. Фільтрація «брязкоту» контактів виконується за допомогою затримки: як тільки отримується перше прийняття даних з клавіатури створюється затримка, під час якої ігнорується повторне прийняття одних і тих же даних.

Модуль VGA формує необхідні синхросигнали для правильної роботи VGA монітору [10]. Після генерації QR-коду виконується його збільшення масштабу (дана збільшена матриця передається на один з вихідних сигналів). Також, паралельно з друком тексту виконується створення матриці пікселів, які відображають текст і передається на вихідний сигнал.

Результат роботи створеного генератора QR-коду показано нижче на Рис. 5 **Ошибка! — Источник ссылки не найден.**



Рис. 5 Результат генерування QR-коду

IV. ВИСНОВКИ

Розроблено апаратний IP-блок для генерування QR-кодів на ПЛІС. На основі даного блоку на платі налагоджування DE1-SoC реалізовано апаратний пристрій, що дозволяє вводити текстову інформацію з клавіатури, генерувати з неї QR-код та виводити його на VGA монітор. Створений пристрій підтримує лише QR-коди першої версії та наразі може кодувати до 14 символів даних, застосовуючи побайтове кодування та рівень корекції типу М (дозволяє відновлювати інформацію із зображення, яке пошкоджено на 15%).

Враховуючи наявність обмежень маємо новий пристрій. За рахунок даних обмежень пристрій хоч може зберігати меншу кількість даних, але має більшу швидкість генерування QR-коду.



Copyright (c) 2020 Прізвище І. П., Прізвище І. П.

						Арк. 92
Змн.	Арк.	№ докум.	Підпис	Дата	ДК62.469515.001 ПЗ	

Враховуючи наявні обмеження у поточній реалізації пристрою, основою для подальшого його покращення можуть стати:

- збільшення довжини кодованих даних;
- підтримка інших методів кодування (цифрового та побайтового);
- додати наступні рівні корекції: L для 7%, Q для 15% та H для 30% пошкоджень;
- можливість використання інших масок.

ПЕРЕЛІК ПОСИЛАНЬ

- [1] "Generator QR codov [Генератор QR кодов]," [Online]. Available: <http://qr-coder.ru/>. [Accessed 15 03 2020].
- [2] "Pochemu nam stoit prismetret'sya k QR-kodam v 2019 godu [Почему нам стоит присмотреться к QR-кодам в 2019 году]," [Online]. Available: <https://lifehacker.ru/qr-kody-v-2019-godu/>. [Accessed 15 03 2020].
- [3] "PLIS (FPGA) i mikrokontroler. V chem raznitsa? [ПЛИС (FPGA) и микроконтроллер. В чем разница?]," [Online]. Available: <http://micro-proger.ru/2016/03/17/plis-fpga-i-mikrokontroler-v-chem-raznica/>. [Accessed 15 03 2020].
- [4] "Verilog Language .," [Online]. Available: <https://en.wikipedia.org/wiki/Verilog>. [Accessed 15 03 2020].
- [5] "ASCII," [Online]. Available: <https://en.wikipedia.org/wiki/ASCII>. [Accessed 15 03 2020].
- [6] "Algoritm generatsii QR-koda [Алгоритм генерации QR-кода]," [Online]. Available: <https://habr.com/ru/post/172525/>. [Accessed 15 03 2020].
- [7] K. Kasperski, "Polinomial'naya arifmetika i polya Galua, ili Informatsiya, voskresshaya iz pepela [Полиномиальная арифметика и поля Галуа, или информация, воскрешая из пепла]," *Sistemnyy administrator*, p. 84–90, 2003.
- [8] "Wounded QR codes," [Online]. Available: <http://datagenetics.com/blog/november12013/index.html>. [Accessed 15 03 2020].
- [9] "Algoritmy raspoznavaniya graficheskogo markera [Алгоритмы распознавания графического маркера]," [Online]. Available: https://studbooks.net/2238811/informatika/algoritmy_raspoznaniya_graficheskogo_markera. [Accessed 15 03 2020].
- [10] *Terasic DE1-SoC User Manual*, 2014, p. pp. 30–33.

UDC 621.389

QR-code generator based on FPGA

Y. V. Sokol^f, ORCID [0000-0002-9104-6690](https://orcid.org/0000-0002-9104-6690)

A. Y. Varfolomiciev^s, PhD, ORCID [0000-0002-6990-7140](https://orcid.org/0000-0002-6990-7140)

Department of Design of Electronic Computing Equipment, Faculty of Electronics
National technical university of Ukraine
"Igor Sikorsky Kyiv polytechnic institute"
Kyiv, Ukraine

Abstract—The IP-core for QR-code generation, which supports codes of the first version and the M-type level of correction (allows to recover information from damaged up to 15% code image) is developed. Based on this IP-core and the DE1-SoC development board, containing the Cyclone V FPGA chip, the real QR-code generation device was implemented. The device uses the PS/2 interface to enter text data that should be encoded in the QR-code and output the result (generated code) on a VGA monitor as an image. The suggested device has the following structure: the control module, the IP-core for QR-code generation and I/O modules. The control module performs the general device management and implemented as the state machine, which has the following states: reset and memory cleaning, read the text that should be encoded from the keyboard, create the main part of the QR-code, create correction data, combine the correction data with the main part of the QR-code, transform the result into a 21x21 pixel matrix and displaying the code on the screen. The input module reads the keyboard signals (synchronization and data signals) and converts them into the ASCII codes. To do this, on each falling edge of the synchronization signal the values of the data signal are registered and propagated to the 8-bit bus. Since the keyboard may have a contact bounce, this module additionally filters the received data using the delay filtering. The output module is the VGA controller that simply generates the necessary clock signals for the proper VGA monitor functioning. After the QR-code is generated, its matrix is enlarged and passed to the VGA controller. Also, in parallel with that, the text, which was encoded in the code, is printed on the display. The IP block for QR-code generation consists of three submodules. The first one receives a text in ASCII format (now only 14 characters are supported) and generates the primary sequence of data to be encrypted in the code. The second submodule calculates the correction data using the Reed-Solomon algorithm. To use this algorithm, the additional array is reserved to store the correction data. This array is filled with special information, depending on the level of correction and the code version number. The algorithm also requires tables of the forward and reverse Galois fields, which are precalculated and stored in the submodule's ROM. The third module generates the matrix itself: it places the alignment patterns, the information of



Copyright (c) 2020 Сокол Я.В.

DOI: 10.20535/2523-4445.у.в.1.п

						Арк.
					ДК62.469515.001 ПЗ	93
Змн.	Арк.	№ докум.	Підпис	Дата		

the used type of mask, level of correction, and adds timing patterns. After that, the mask superimposed on the data generated by the previous module, and the result is copied to the matrix code in the required order.

Keywords — QR-code; FPGA; VGA; keyboard.

DOI: [10.20535/2523-4455.y.v.i.n](https://doi.org/10.20535/2523-4455.y.v.i.n)



Copyright (c) 2020 Прізвище І. П., Прізвище І. П.

					<i>ДК62.469515.001 ПЗ</i>	Арк.
Змн.	Арк.	№ докум.	Підпис	Дата		94