

МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ

Сумський державний університет

Факультет електроніки та інформаційних технологій

Кафедра електроніки та комп'ютерної техніки

«До захисту допущено»

Завідувач кафедри

Анатолій ОПАНАСЮК

(підпис)

(Ім'я та ПРИЗВИЩЕ)

_____ 20__ р.

КВАЛІФІКАЦІЙНА РОБОТА

на здобуття освітнього ступеня бакалавра

зі спеціальності 171 Електроніка

освітньої програми "Електронні системи та компоненти"

на тему: «Голографічний пристрій відображення інформації»

Здобувача групи ЕС-91/4-1 Подобрія Андрія Дмитровича

Кваліфікаційна робота містить результати власних досліджень. Використання ідей, результатів і текстів інших авторів мають посилання на відповідне джерело.

_____ Андрій ПОДОБРІЙ

(підпис)

(Ім'я та ПРИЗВИЩЕ здобувача)

Керівник старший викладач, к.т.н. Олексій ГОРЯЧЕВ

(посада, науковий ступінь, вчене звання, Ім'я та ПРИЗВИЩЕ)

_____ (підпис)

Суми – 2023

Календарний план

№ п/п	Найменування етапів дипломного проекту (роботи)	Термін виконання етапів проекту (роботи)	Примітка
1	Огляд літератури	10.04.2023	
2	Принцип побудови та основні функціональні блоки системи	25.04.2023	
3	Розрахунок інтенсивності навантаження на мережі	12.05.2023	
4	Розрахунок кількості обладнання системи	20.05.2023	
5	Представлення роботи керівнику	30.05.2023	

Студент-дипломник _____ Подобрій Андрій Дмитрович _____

Керівник проекту _____ Горячев Олексій Євгенійович _____

« _____ » _____ 20__ р.

СУМСЬКИЙ ДЕРЖАВНИЙ УНІВЕРСИТЕТ

Факультет. Кафедра Електроніки і комп'ютерної техніки
Спеціальність 6.05080202 "Електронні системи"

ЗАТВЕРДЖУЮ

Зав. кафедрою Опанасюк А.С.

“ _____ ” _____ 201__ р.

ЗАВДАННЯ

на кваліфікаційну роботу бакалавра студентові

Подобрій Андрій Дмитрович

(прізвище, ім'я, по батькові)

1 Тема роботи: «Голографічний пристрій відображення інформації», затверджено наказом по інституту від «30» березня 2023р. № 0310 - VI

2 Термін здачі студентом закінченого проекту - 06.06.2023 р.

3 Вихідні дані до проекту: Пристрій може відображати різноманітні типи контенту, включно зі статичними зображеннями, анімаціями. Розроблений пристрій компактний і портативний, що дає змогу його використання в різних умовах і місцях.

4 Зміст розрахунково-пояснювальної записки (питання, які належить розробити):

4.1. Огляд літератури та поставлення задачі проекту.

4.2. Вибір та обґрунтування алгоритму функціонування та схеми структурної системи.

4.3. Розробка функціональних схем блоків системи.

4.4. Вибір елементної бази та розробка схеми електричної.

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

5.1. Схема електрична структурна

5.2. Схема електрична функціональна

5.3. Схема електрична принципова

Керівник _____.

(підпис) (прізвище, ім'я, по батькові)

Завдання прийняв до виконання _____

(підпис студента)

АНОТАЦІЯ

Тема дипломного проекту: «Голографічний пристрій відображення інформації». Дипломний проект складається з 4 розділів, розділи у свою чергу розбиті на окремі підрозділи. Робота містить 67 сторінок. Робота включає у себе схему алгоритму, структурну схему системи, електричну функціональну схему, електричну принципову схему. При написанні роботи було опрацьовано 12 джерел.

У першому розділі дипломного проекту був зроблений вибір напрямку й пристрою для проектування, описана актуальність проектування, описана загальна структура і призначення пристроїв голографічного відображення інформації, наведено основні етапи розробки проектованого пристрою, поставлена задача проектування. У другому розділі зроблений вибір й обґрунтування алгоритму функціонування проектованого пристрою, розроблена структурна схема пристрою автоматичного регулювання. У третьому розділі була розроблена функціональна схема, на основі якої була побудована схема електрична принципова. Був виконаний вибір та розрахунок необхідних елементів пристрою. У четвертому розділі описується розробка програмного забезпечення пристрою, яке приведено у додатку роботи. Зроблені висновки по виконаній роботі.

Ключові слова: ГОЛОГРАФІЯ, ВІДОБРАЖЕННЯ ІНФОРМАЦІЇ, ДИСПЛЕЙ

ЗМІСТ

ВСТУП.....	4
1. ОГЛЯД ЛІТЕРАТУРИ.....	6
1.1 Відомості про пристрої голографічного відображення інформації.....	6
1.2 Етапи розробки проектного пристрою.....	8
1.3 Постановка задачі проектування.....	12
2. ВИБІР ТА ОБҐРУНТУВАННЯ СТРУКТУРНОЇ СХЕМИ ПРОЕКТОВАНОГО ПРИБОРУ.....	14
2.1 Розробка алгоритму функціонування проектного пристрою	14
2.2 Розробка структурної схеми проектного пристрою	15
3. РОЗРОБЛЕННЯ ТА РОЗРАХУНКИ ПРИНЦИПОВИХ ЕЛЕКТРИЧНИХ СХЕМ ВУЗЛІВ І БЛОКІВ ПРИБОРУ	16
3.1 Розробка функціональної схеми проектного пристрою	16
3.2 Вибір та розрахунок елементної бази	17
4. РОЗРОБКА ПРОГРАМНОГО ЗАБЕЗПЕЧЕННЯ	27
ВИСНОВКИ.....	34
СПИСОК ЛІТЕРАТУРИ.....	35
ДОДАТОК А.....	36

					ЕлІТ 6.171.00.02.267 ПЗ			
<i>Вим.</i>	<i>Арк.</i>	<i>№ докум.</i>	<i>Підпис</i>	<i>Дата</i>				
Розроб.		Подобрий А.Д.			<i>Голографічний пристрій відображення інформації</i>	<i>Лит.</i>	<i>Аркуш</i>	<i>Аркушів</i>
Перевір.		Горячев О.Є.				3	35	
Реценз.								
Н. Контр.		Горячев О.Є.				СумДУ гр. ЕС-91/4-1		
Затверж.		Опанасюк А.С.						

ВСТУП

Голографічні дисплеї - це нова технологія, що використовує принципи голографії для створення тривимірних зображень, які можна розглядати під різними кутами. Ця технологія має величезний потенціал у багатьох галузях, включно з освітою, медициною, розвагами і виробництвом. Пристрої відображення голографічної інформації також можуть допомогти людям краще зрозуміти навколишній світ і поліпшити спосіб взаємодії з інформацією.

Наприклад, голографічні пристрої можна використовувати для створення інтерактивних освітніх додатків, що допомагають учням краще розуміти наукові концепції та історичні події.

У реальний час інформаційні технології зазнають швидкого становлення, що вносить значні видозміни в багато сфер людського життя. Голографія являє собою спосіб виробництва тривимірних зображень, що володіють реалістичністю і глибиною, що робить її особливо симпатичною для різних додатків.

Метою наданої дипломної роботи є освоєння і розробка голографічного пристрою, здатного відображати інформацію із застосуванням голографічних спецтехнологій. Головним завданням роботи є розробка пристрою, який дасть змогу користувачеві взаємодіяти з тривимірними зображеннями й отримувати інформацію з високим ступенем реалістичності.

У процесі вивчення буде розглянуто основні тези голографії, включно з утворенням голографічного зображення, його записом і відтворенням. Буде вивчено наявні способи і спецтехнології, що використовуються в голографії. За допомогою отриманих умінь буде розроблено голографічний пристрій, що пропонує нові можливості у відображенні інформації.

Підсумки наданої роботи можуть бути використані в різних галузях, як-от медицина, освіта, веселощі та технічне проектування. Хоча голографічні пристрої відображення все ще перебувають на стадії розробки і не всі проблеми вирішені, цей підхід має великий потенціал і може мати важливе значення для розробки майбутніх пристроїв відображення. Однією з основних проблем голографічних

					<i>ЕЛІТ 6.171.00.02.267 ПЗ</i>	Арк
						4
<i>Вим.</i>	<i>Арк.</i>	<i>№ докум.</i>	<i>Підпис</i>	<i>Дата</i>		

пристроїв є їхня роздільна здатність і частота оновлення. Але в міру вдосконалення технологій ці проблеми можуть бути вирішені, що відкриє нові можливості для створення реалістичних та інтерактивних 3D-зображень. Голографічні пристрої можуть набагато вдосконалити здатність людей сприймати і взаємодіяти з інформацією, відкриваючи нові горизонти для інновацій і поліпшення.

					<i>ЕЛІТ 6.171.00.02.267 ПЗ</i>	Арк
<i>Вим.</i>	<i>Арк.</i>	<i>№ докум.</i>	<i>Підпис</i>	<i>Дата</i>		5

1. ОГЛЯД ЛІТЕРАТУРИ

1.1 Відомості про пристрої голографічного відображення інформації

Голографія - це метод отримання тривимірного зображення об'єкта за допомогою інтерференції хвиль. Вперше ідею голографії висловив Д. Габор (Англія, 1948 р.), але її було надзвичайно складно реалізувати технічно, і голографія не стала популярною. Поява лазера відкрила численні та різноманітні можливості для практичного застосування голографії в радіоелектроніці, оптиці, фізиці та різних технічних галузях.

Для отримання зображення об'єкта фотографічним способом зазвичай використовується камера, яка фіксує розсіяне від об'єкта випромінювання на фотопластинку. Кожна точка об'єкта є центром розсіювання падаючого світла, який посилає в простір сферичні розбіжні світлові хвилі, що фокусуються об'єктивом в маленьку точку на світлочутливій поверхні фотопластинки. Оскільки відбивна здатність об'єкта змінюється від точки до точки, інтенсивність світла, що падає на відповідну ділянку фотопаперу, також змінюється. Тому на фотопапері з'являється зображення об'єкта. Це зображення складається із зображення відповідних точок об'єкта, сформованих на кожній частині світлочутливої поверхні. Таким чином, тривимірний об'єкт записується у вигляді плоского двовимірного зображення.

У процесі фотографування на фотопапір записується лише розподіл інтенсивності або амплітуда відбитих від об'єкта електромагнітних хвиль. Однак світлові хвилі, відбиті від об'єкта, змінюються не тільки за амплітудою, але й за фазою, залежно від властивостей поверхні об'єкта в даній точці.

Голографія надає більш повну інформацію про об'єкт, записуючи на фотопластинку не тільки амплітуду, але й фазу світлових хвиль, розсіяних об'єктом. Для цього на фотопластинку одночасно з хвилею (сигналом), розсіяною об'єктом, повинна бути опромінена допоміжна хвиля постійної амплітуди і фази від того ж джерела світла.

					ЕЛІТ 6.171.00.02.267 ПЗ	Арк
						6
Вим.	Арк.	№ докум.	Підпис	Дата		

Геометрична, або променева, оптика адекватно описує багато оптичних явищ, і її використання приносить велику практичну користь, наприклад, у створенні оптичних приладів, при якісному описі багатьох особливостей оптичних систем тощо.

Однак хвильові властивості світла, електромагнітна природа світлового випромінювання відіграють надзвичайно важливу роль під час дослідження особливостей передавання оптичним випромінюванням інформації про об'єкти матеріального світу та кількісного визначення характеристик речовин і процесів, особливо в голографії.

Для голографії та оптичного опрацювання інформації вельми важливими характеристиками є просторова і часова когерентність.

Під час поширення електромагнітної хвилі в речовині або під час відбиття і розсіювання на поверхні тіл, ті чи інші характеристики хвильового поля зазнають змін. Отже, хвильове поле, що пройшло крізь будь-яке середовище, відбите або розсіяне середовищем, стає носієм інформації про властивості та структуру тіл і речовин, з яких складаються ці тіла...

Можна визначити такі сфери застосування голографії: як образотворчий засіб; у техніці передавання й оброблення інформації; у кібернетиці, автоматичній та обчислювальній техніці; як засіб дослідження в оптичному і фізичному приладобудуванні, у технології.

Застосування голографії як образотворчого засобу зачіпає фотографію, кіно і телебачення.

Ще більш важливими видаються завдання, які можуть бути вирішені в техніці передавання й оброблення інформації та обчислювальній техніці: тут є перспективи створення цілого класу спеціалізованих аналогових оптичних обчислювальних машин.

Використання методів голографії та голографічної інтерферометрії в технології дає змогу розв'язувати задачі неруйнівного контролю якості виробів, а

					<i>ЕЛІТ 6.171.00.02.267 ПЗ</i>	Арк
						7
<i>Вим.</i>	<i>Арк.</i>	<i>№ докум.</i>	<i>Підпис</i>	<i>Дата</i>		

також здійснювати дефектоскопію виробів в ультразвуковому та рентгенівському діапазонах.

Широкі перспективи для обчислювальної техніки відкриває можливість створення голографічних систем пам'яті з великою інформаційною ємністю.

У радіодіапазоні голографія принципово дає змогу розв'язувати велику групу завдань, серед яких можна вказати на створення систем об'ємного телебачення, створення пристроїв радіолокаційного типу із синтезованою апертурою для одержання радіозображення з надвисокою роздільною здатністю тощо.

Стосовно акустики голографію можна використовувати для створення систем звуковіденання, зокрема об'ємного.

Голографія може знайти широке застосування в оптиці та оптичному приладобудуванні.

З наведеного далеко не повного переліку видно, наскільки широка сфера застосування голографії

1.2 Етапи розробки проектного пристрою

Голографічні дисплеї являють собою багате і захоплююче майбутнє в технології відображення. Ця технологія здатна змінити те, як ми взаємодіємо з інформацією та навколишнім світом, створюючи нові можливості для різних сфер діяльності.

Своїм проектом було вирішено зробити простий голографічний дисплей на основі мікроконтролера arduino nano та адресної світлодіодної стрічки. Для цього проекту було виділено такі етапи:

- Планування і концептуалізація
- Написання схеми та підключення компонентів
- Написання програмного коду
- Тестування та налагодження

					ЕЛІТ 6.171.00.02.267 ПЗ	Арк
						8
Вим.	Арк.	№ докум.	Підпис	Дата		

Перший етап розробки голографічного дисплея - це планування і концептуалізація проєкту. На цьому етапі визначаються основні вимоги та створюється загальна концепція дисплея.

1. Визначення вимог:

- Розміри: Визначення бажаного розміру голографічного дисплея. Розмір може залежати від обмежень простору і передбачуваного використання.
- Роздільна здатність: Визначення необхідної роздільної здатності дисплея. Роздільна здатність впливає на деталізацію і чіткість відображуваних голографічних зображень.
- Тип адресної стрічки: Вибір відповідного типу адресної стрічки для створення голографічних ефектів.
- Датчик Холла: Вирішення, якого типу датчик Холла буде використовуватися для визначення положення моторчика.

2. Розробка загальної концепції та дизайну:

- Вивчення наявних прикладів голографічних дисплеїв.
- Розробка загальної концепції дисплея, визначення його форми і структури.

3. Проведення дослідження:

- Вивчення доступних матеріалів та ресурсів про голографію та використання адресних стрічок на Arduino.
- Вивчення документації Arduino Nano, адресної стрічки і датчика Холла для розуміння їхніх можливостей і специфікацій.
- Ознайомлення з прикладами коду і проєктами, пов'язаними з голографічними ефектами, адресними стрічками і датчиками Холла.

У результаті першого етапу ми повинні мати чітке уявлення про вимоги до голографічного дисплея і його концепції. Це забезпечить основу для детальнішого розроблення, включно з вибором компонентів, схемами підключення та програмуванням Arduino Nano.

					<i>ЕЛІТ 6.171.00.02.267 ПЗ</i>	Арк
						9
<i>Вим.</i>	<i>Арк.</i>	<i>№ докум.</i>	<i>Підпис</i>	<i>Дата</i>		

Другий етап розробки голографічного дисплея - написання схеми та підключення компонентів. На цьому етапі буде створення схема підключення із обраних компонентів та їх підключення за створенною схемою.

1. Створення схеми підключення:

- Створення схеми підключення, яка вказує, які дроти і компоненти будуть підключені до кожного піна Arduino Nano.
- За необхідності використання додаткових компонентів, як-от резистори або конденсатори, для забезпечення правильної роботи адресної світлодіодної стрічки та датчика Холла.

2. Підключення та перевірка компонентів:

- Відповідно до створенної схеми підключаємо всі компоненти, звертаючи увагу на особливості підключення.
- Перевірка правильності під'єднання всіх компонентів, що дроти надійно закріплені та не перекривають один одного.
- Підключення Arduino Nano до комп'ютера за допомогою USB-кабелю і завантаження тестової програми для перевірки роботи адресної світлодіодної стрічки і датчика Холла.
- Переконавшись, що адресна світлодіодна стрічка працює коректно і здатна відображати різні кольори та ефекти. Також перевірка роботи датчика Холла, переконавшись, що він може виявляти зміну магнітного поля від моторчика.

У результаті другого етапу маємо готову схему підключення і перевірені компоненти, готові до програмування і подальшого налаштування.

Третій етап - написання програмного коду, є одним із ключових етапів розробки голографічного дисплея.

1. Визначення функціональності:

- Визначення основної функціональності голографічного дисплея. Які ефекти та анімації будуть відображатися на дисплеї, як користувач буде взаємодіяти з дисплеєм.

					ЕЛІТ 6.171.00.02.267 ПЗ	Арк
Вим.	Арк.	№ докум.	Підпис	Дата		10

2. Вивчення бібліотек і документації:

- Вивчення доступних бібліотек та ресурсів, які допоможуть реалізувати бажану функціональність. Наприклад, для роботи з адресною стрічкою можна використовувати бібліотеки, як-от FastLED або NeoPixel, які надають зручні функції для керування адресними світлодіодами.
- Ознайомлення з документацією Arduino Nano, щоб зрозуміти, як правильно використовувати його функції та піни введення-виведення.

3. Створення основного циклу програми:

- Написання основного циклу програми, який виконуватиметься на Arduino Nano. Цей цикл контролюватиме весь процес роботи голографічного дисплея.

4. Реалізація конкретних функцій і ефектів:

- Реалізування конкретних функцій та ефектів, які були визначені на першому кроці.
- Використання функцій бібліотеки адресної стрічки для керування відображенням світлодіодів.
- Написання коду для обробки взаємодії користувача з дисплеєм, наприклад, за допомогою кнопок або інших сенсорів.

5. Завантаження програмного коду на Arduino Nano:

- Підключення Arduino Nano до комп'ютера за допомогою USB-кабелю та використання Arduino IDE або інше середовище розробки для завантаження програмного коду на плату Arduino Nano.
- Переконавання, що обрано правильний порт і тип плати в налаштуваннях Arduino IDE.

Четвертий етап - тестування та налагодження. Необхідний для перевірки та вирішення помилок які могли виникнути в ході розробки коду.

1. Налагодження програмного коду:

					<i>ЕЛІТ 6.171.00.02.267 ПЗ</i>	Арк
<i>Вим.</i>	<i>Арк.</i>	<i>№ докум.</i>	<i>Підпис</i>	<i>Дата</i>		11

- Під час тестування можуть виникати помилки або непередбачувана поведінка. Використання монітору порту в Arduino IDE або інші засоби налагодження, щоб ідентифікувати та виправити проблеми.
- Перевірка, що всі змінні та функції в програмному коді працюють правильно і взаємодіють одна з одною так, як задумано.
- Перевірка логіки програми, що всі умови та цикли виконуються правильно.

2. Ітеративне тестування та коригування:

- Після виявлення помилок або неправильної поведінки в програмному коді, внесення необхідних змін і повторне тестування коду. Доки воне не працюватиме стабільно і без помилок

3. Валідація та порівняння з очікуваними результатами:

- Перевірка, що результати тестування відповідають вашим очікуванням і вимогам проєкту.
- Переконавання, що голографічний дисплей правильно реагує на зміни магнітного поля і відображає необхідні ефекти на адресній стрічці.

Тестування та налагодження дають змогу переконатися в коректній роботі голографічного дисплея та виявити й виправити можливі проблеми й помилки в програмному коді. Це важливий етап, який гарантує надійність і ефективність голографічного дисплея.

1.3 Постановка задачі проєктування

Основна мета дипломного проєкту полягає в дослідженні та розробленні голографічного пристрою, здатного відображати інформацію з використанням голографічної технології.

Конкретні підцілі проєкту можуть включати:

- вивчення основ голографії;
- розробка голографічного пристрою;
- створення програмного коду;

					ЕЛІТ 6.171.00.02.267 ПЗ	Арк
						12
<i>Вим.</i>	<i>Арк.</i>	<i>№ докум.</i>	<i>Підпис</i>	<i>Дата</i>		

- тестування та налагодження;
- валідація та оцінювання результатів.

Актуальність роботи з розроблення голографічного пристрою відображення інформації полягає в безлічі чинників, які є актуальними та затребуваними в сучасному суспільстві. Цей напрям має широкий потенціал застосування і може принести значні переваги в різних сферах життя людини.

					<i>ЕЛІТ 6.171.00.02.267 ПЗ</i>	<i>Арк</i>
<i>Вим.</i>	<i>Арк.</i>	<i>№ докум.</i>	<i>Підпис</i>	<i>Дата</i>		13

2. ВИБІР ТА ОБҐРУНТУВАННЯ СТРУКТУРНОЇ СХЕМИ ПРОЕКТОВАНОГО ПРОСТРОЮ

2.1 Розробка алгоритму функціонування проектного пристрою

Алгоритм роботи проектного пристрою (рисунок 2.1) лінійний. Алгоритм дії проектного пристрою – циклічний, що реалізується на рівні програми мікроконтролера.



Рисунок 2.1 - Граф-схема алгоритму роботи пристрою.

2.2 Розробка структурної схеми проектного пристрою

Узагальнена структурна схема керуючого пристрою приведена на рисунку 2.2.

Пристрій складається з основних елементів:

Мікроконтролер - пристрій, що використовується для управління електронними пристроями;

Блок живлення - забезпечує необхідні рівні напруги для мікроконтролерів та пристроїв керування;

Датчик Холла - прилад для вимірювання величини магнітного поля;

Світлодіодна стрічка - це джерело світла, що являє собою набір світлодіодів;

Підвищувальний перетворювач - електронний пристрій, що використовується для збільшення або підвищення рівня вхідної напруги до бажаного вихідного значення.

Двигун - електричне обладнання, яке перетворює електричну енергію в механічну кінетичну енергію.

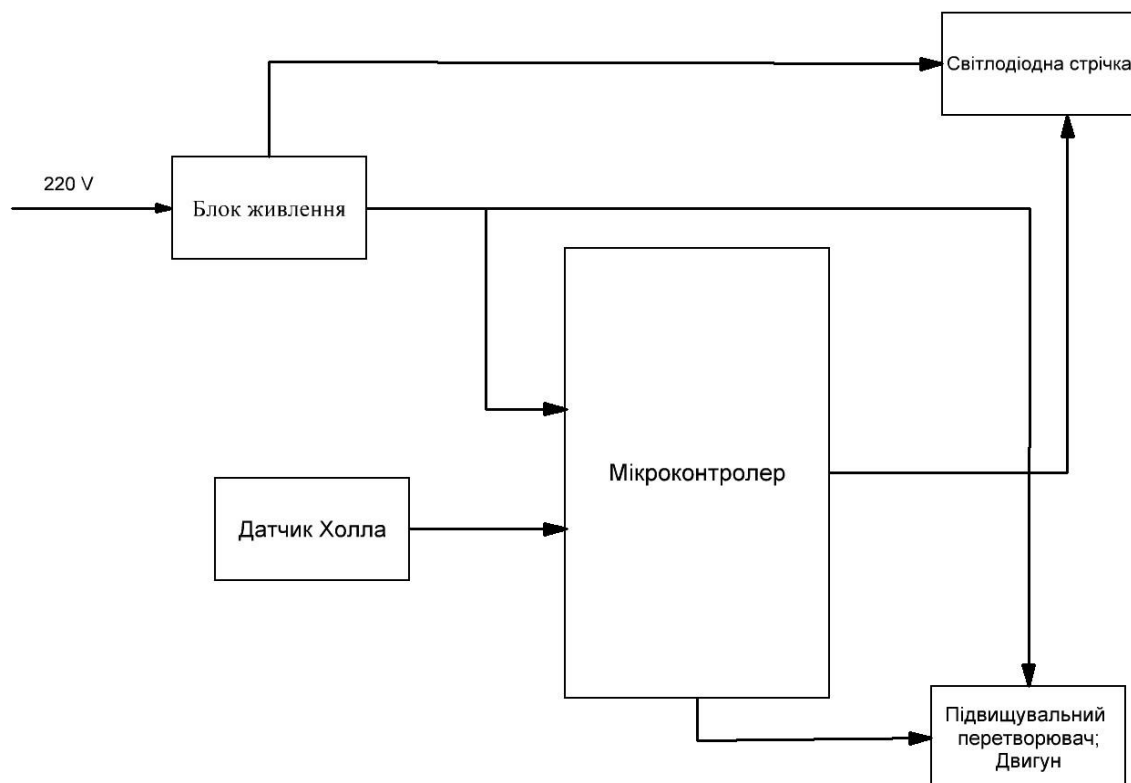


Рисунок 2.2 - Структурна схема голографічного дисплея

перетворювач (boost converter). Підвищувальний перетворювач дає змогу збільшити напругу живлення, що може бути корисно для досягнення необхідного рівня потужності для роботи моторчика. Конденсатор використовується для згладжування напруги та стабілізації живлення. Він може бути під'єднаний паралельно з моторчиком або на вході підвищувального перетворювача для фільтрації шумів і підтримки стабільного живлення. Резистори можуть бути використані в схемі для різних цілей, як-от обмеження струму або встановлення певних значень опору в ланцюзі керування. У цій схемі резистори можуть використовуватися в різних частинах пристрою залежно від конкретних вимог. Датчик Холла використовується для вимірювання обертів моторчика і визначення його швидкості обертання. Він виявляє зміни магнітного поля, створюваного моторчиком, що обертається, і генерує відповідні імпульси, які потім обробляються мікроконтролером. Моторчик являє собою пристрій, який перетворює електричну енергію в механічну енергію обертання. Він може бути постійного або змінного струму залежно від конкретних вимог і типу пристрою. Моторчик керується за допомогою транзистора і може обертатися з різними швидкостями залежно від регулювання потужності.

Усю схему керування, включно з блоком живлення, транзистором, підвищувальним перетворювачем, конденсатором, резисторами, датчиком Холла і двигуном, пов'язана за допомогою дротів і з'єднувальних елементів відповідно до вимог системи. Крім того, мікроконтролер Arduino пов'язаний із датчиком Холла для оброблення даних про швидкість обертання моторчика та керування вихідним сигналом або іншими компонентами системи.

3.2 Вибір та розрахунок елементної бази

3.2.1 Вибір мікроконтролера

Для проектування пристрою голографічного відображення інформації необхідно вибрати відносно недорогий, простий і широковживаний мікроконтролер. Всі ці особливості можна віднести до мікроконтролерів Arduino

					ЕЛІТ 6.171.00.02.267 ПЗ	Арк
Вим.	Арк.	№ докум.	Підпис	Дата		17

Nano V3.0 (ATmega328P). ATmega328P є 8-бітовим мікроконтролером, який широко використовується в різних електронних пристроях і особливо в платформі Arduino.

Докладний опис характеристик ATmega328P:

1. Архітектура і процесор: ATmega328P заснований на архітектурі AVR і оснащений одноядерним процесором. Він має піпеліновану архітектуру з процесорним ядром, що працює на тактовій частоті до 20 МГц.
2. Кількість і тип пінів: ATmega328P має 28 пінів у пластиковому корпусі Dual Inline Package (DIP) або Quad Flat Package (QFP). Ці піни можуть бути налаштовані як цифрові входи/виходи або аналогові входи. Деякі з них також підтримують функціональність ШІМ (PWM) і переривань.
3. Пам'ять:
 - Flash-пам'ять: ATmega328P має 32 Кб внутрішньої флеш-пам'яті, де зберігається програма (код).
 - Оперативна пам'ять (SRAM): Він має 2 Кб оперативної пам'яті для зберігання тимчасових даних і змінних під час виконання програми.
 - EEPROM: ATmega328P також містить 1 Кб енергонезалежної пам'яті EEPROM для зберігання постійних даних, які не губляться при вимкненні живлення.
4. Частота тактування: ATmega328P може працювати із зовнішнім кварцовим резонатором або керамічним резонатором, а також із внутрішнім RC-генератором із частотою 8 МГц. Він підтримує різні значення тактової частоти, включно з 1 МГц, 8 МГц і 16 МГц.
5. Комунікаційні інтерфейси:
 - UART: ATmega328P має вбудований UART (Universal Asynchronous Receiver-Transmitter), який дає змогу обмінюватися даними за допомогою послідовного інтерфейсу.
 - SPI: Він підтримує інтерфейс послідовного периферійного інтерфейсу (SPI) для обміну даними з іншими пристроями.

					ЕЛІТ 6.171.00.02.267 ПЗ	Арк
Вим.	Арк.	№ докум.	Підпис	Дата		18

- I2C: ATmega328P також підтримує інтерфейс I2C (Inter-Integrated Circuit) для комунікації з пристроями по шині I2C.
- 6. Аналогово-цифрові перетворювачі (ADC): ATmega328P має 10-бітний вбудований АЦП з 8 аналоговими входами, що дає змогу зчитувати аналогові значення з датчиків та інших пристроїв.
- 7. Таймери/лічильники: Мікроконтролер має кілька вбудованих 8- і 16-бітних таймерів/лічильників, які можна використати для вимірювання часу, генерації затримок, керування ШІМ-сигналами та інших завдань таймінгу.
- 8. Підтримка переривань: ATmega328P підтримує зовнішні переривання, які дають змогу пристрою реагувати на зміни станів пінів і виконувати певні дії.
- 9. Споживання енергії: Мікроконтролер має різні режими енергозбереження, які дають змогу зменшити споживання енергії під час неактивності, що корисно для батарейного живлення та портативних пристроїв.

3.2.2 Блок живлення

Блок живлення 5 В є джерелом постійної напруги для всієї схеми. Він забезпечує стабільну напругу 5 В, необхідну для живлення всіх компонентів, включно з мікроконтролером Arduino, моторчиком та іншими пристроями. Блок живлення реалізований у вигляді зовнішнього джерела живлення, такого як адаптер змінного струму, що перетворює вхідну напругу змінного струму у вихідну напругу постійного струму 5 В.

3.2.3 Транзистор

Транзистор використовується в цій схемі для керування потужністю, що надходить на моторчик. Він виконує функцію ключа, який може відкриватися і закриватися під контролем сигналу з мікроконтролера Arduino. У цьому разі транзистор працює в режимі ключа, здатного керувати високим струмом і напругою, які споживає моторчик. Підключення транзистора включає в себе базу (В), колектор (С) і емітер (Е). База транзистора під'єднується до піна мікроконтролера Arduino, який керуватиме транзистором. Колектор транзистора

					ЕЛІТ 6.171.00.02.267 ПЗ	Арк
						19
Вим.	Арк.	№ докум.	Підпис	Дата		

під'єднується до плюсового полю блока живлення 5 В, а емітер - до позитивного виводу моторчика. Коли сигнал на базі транзистора подається з мікроконтролера і його напруга перевищує певний поріг, транзистор відкривається і дає змогу протікати струму від колектора до емітера. Таким чином, потужність від блоку живлення 5 В надходить на моторчик через транзистор. Керування транзистором здійснюється шляхом зміни сигналу на базі за допомогою мікроконтролера Arduino. Коли сигнал на базі високий, транзистор відкривається повністю і максимальний струм проходить через моторчик, забезпечуючи його максимальну потужність. Коли сигнал на базі низький, транзистор закривається, і струм перестає протікати через моторчик, що призводить до його вимкнення. Таким чином, використання транзистора дає змогу мікроконтролеру Arduino контролювати потужність, що надходить на моторчик, і вмикати/вимикати його відповідно до програмних команд і вимог. Це дає змогу регулювати швидкість і напрямок обертання моторчика в проєкті.

Деякі важливі параметри IRF3205:

- Максимальний струм зливу (I_D) = 110 А
- Напруга зливу-витік (V_{DS}) = 55 В
- Коефіцієнт посилення струму (K) = 0,008 Ом

Розрахунок базового струму:

У MOSFET-транзисторах не потрібен базовий струм, тому що керування здійснюється додаванням напруги до затвора (gate). В даному випадку, для управління IRF3205, ми будемо використовувати транзисторний ключ, який буде контролювати напругу на затворі транзистора.

Перевірка потужності:

Потужність, що розсіюється транзистором IRF3205, може бути розрахована за формулою:

$$P = V_{DS} * I_D$$

У цьому випадку:

$$P = (5 \text{ В} - 3,6 \text{ В}) * 0,25 \text{ А} = 0,4 \text{ Вт}$$

					ЕЛІТ 6.171.00.02.267 ПЗ	Арк
						20
Вим.	Арк.	№ докум.	Підпис	Дата		

Потужність розраховується як різниця напруг (V_{DS}) помножена на струм зливу (I_D) під час роботи в режимі насичення ($V_{DS} > V_{GS}$).

Таким чином, транзистор IRF3205 має бути здатний впоратися із заданими параметрами.

3.2.4 Підвищувальний перетворювач

Підвищувальний перетворювач (або підвищувальний регулятор) - це електронний пристрій, який використовується для збільшення напруги постійного або змінного струму. Він зазвичай складається з індуктивності, комутаційного елемента (транзистора або ключа), діода і конденсатора.

Робота підвищувального перетворювача заснована на принципі електромагнітної індукції. Коли ключ вмикається, енергія надходить в індуктивність, створюючи магнітне поле. Потім ключ вимикається, і магнітне поле індуктивності викликає появу зворотної електрорушійної напруги, яка заряджає конденсатор і підвищує вихідну напругу.

Підвищувальні перетворювачі широко застосовують у різних пристроях, як-от блоки живлення, системи безперебійного живлення (ДБЖ), перетворювачі постійної напруги (DC-DC), а також у багатьох інших застосунках, де потрібне підвищення напруги.

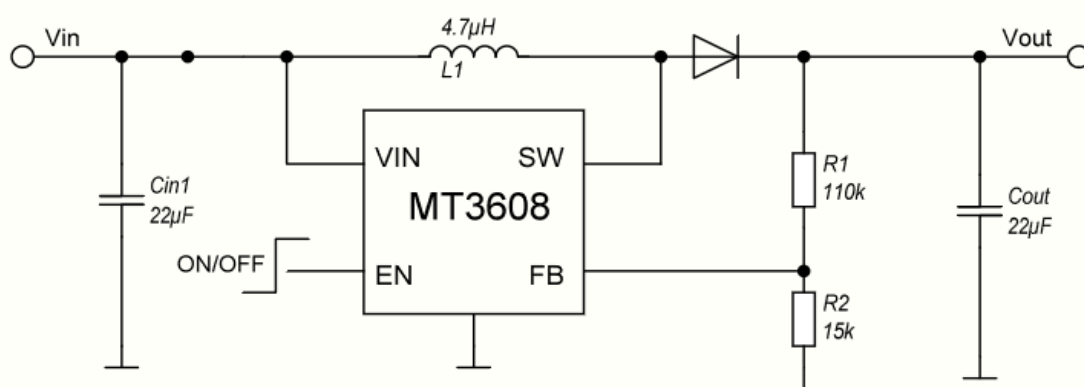


Рисунок 3.2 - Схема підвищувального перетворювача

Підвищувальний перетворювач використовується для підвищення напруги живлення від блока живлення 5 В до необхідного рівня для роботи моторчика. В даному випадку, він підвищує напругу з 5 В до рівня, необхідного для живлення моторчика 12 В. Підвищувальний перетворювач працює на основі принципу перетворення енергії. Він складається з індуктивності, діода, конденсатора і ключового елемента. Коли ключовий елемент (транзистор) відкритий, струм протікає через індуктивність, створюючи магнітне поле і енергію, що зберігається в ньому. Потім, коли ключовий елемент закритий, магнітне поле індуктивності схлопується, створюючи зворотну напругу, яка заряджає конденсатор. Діод запобігає зворотному струму від конденсатора до індуктивності. Таким чином, підвищувальний перетворювач циклічно повторює процес заряджання і розряджання індуктивності, створюючи вищу вихідну напругу, ніж вхідна. Це дає змогу досягти необхідної напруги для живлення моторчика. Важливим елементом у підвищувальному перетворювачі є конденсатор. Він служить для фільтрації вихідної напруги, згладжування пульсацій і забезпечення стабільної постійної напруги для живлення моторчика. Конденсатор зберігає енергію і забезпечує плавне живлення моторчика, усуваючи можливі пульсації та шуми. Таким чином, використання підвищувального перетворювача дає змогу отримати необхідну напругу для живлення моторчика з вихідного блока живлення 5 В. Це забезпечує ефективне та стабільне живлення моторчика, необхідне для його працездатності та керування за допомогою мікроконтролера Arduino

Деякі важливі параметри МТ3608:

- Вхідна напруга (V_{in}) = 2 В - 24 В (зазвичай 5 В)
- Вихідна напруга (V_{out}) = 5 В
- Максимальний вихідний струм (I_{out}) = 2 А
- Ефективність перетворення (η) = до 92%.

Розрахунок струмів навантаження і вхідного струму:

					ЕЛІТ 6.171.00.02.267 ПЗ	Арк
Вим.	Арк.	№ докум.	Підпис	Дата		22

Зазначено, що струм навантаження моторчика становить максимум 250 мА за напруги 3,6 В. Отже, нам необхідно переконатися, що МТ3608 здатний надати достатній струм навантаження.

Вхідний струм перетворювача може бути розрахований з використанням формули:

$$I_{in} = (I_{out} * V_{out}) / V_{in}$$

У даному випадку:

$$I_{in} = (0,25 \text{ А} * 5 \text{ В}) / 3,6 \text{ В} \approx 0,347 \text{ А (або 347 мА)}$$

Перевірка потужності:

Потужність, що розсіюється в перетворювачі МТ3608, може бути розрахована за формулою:

$$P = (V_{in} * I_{in}) - (V_{out} * I_{out})$$

У цьому випадку:

$$P = (5 \text{ В} * 0,347 \text{ А}) - (5 \text{ В} * 0,25 \text{ А}) = 0,347 \text{ Вт}$$

Таким чином, перетворювач МТ3608 має бути здатний забезпечити достатню напругу і струм для живлення моторчика із заданими характеристиками.

3.2.5 Датчик Холла

Датчики Холла - це електронні пристрої, які використовують ефект Холла для вимірювання магнітних полів або для визначення наявності та напрямку магнітного поля. Вони названі на честь американського фізика Едвіна Холла, який відкрив цей ефект.

Датчики Холла складаються зі спеціального напівпровідникового матеріалу, через який протікає електричний струм. Коли до цього матеріалу прикладається магнітне поле, створюється різниця потенціалів (напруга) у напрямку, перпендикулярному до струму. Це явище відоме як ефект Холла.

Датчики Холла можна використовувати для різних застосувань, включаючи вимірювання магнітного поля, виявлення і позиціонування магнітних матеріалів, а також вимірювання струму і швидкості обертання. Вони широко

					ЕЛІТ 6.171.00.02.267 ПЗ	Арк
Вим.	Арк.	№ докум.	Підпис	Дата		23

використовуються в різних галузях промисловості, включаючи автомобільну, медичну та електронну.

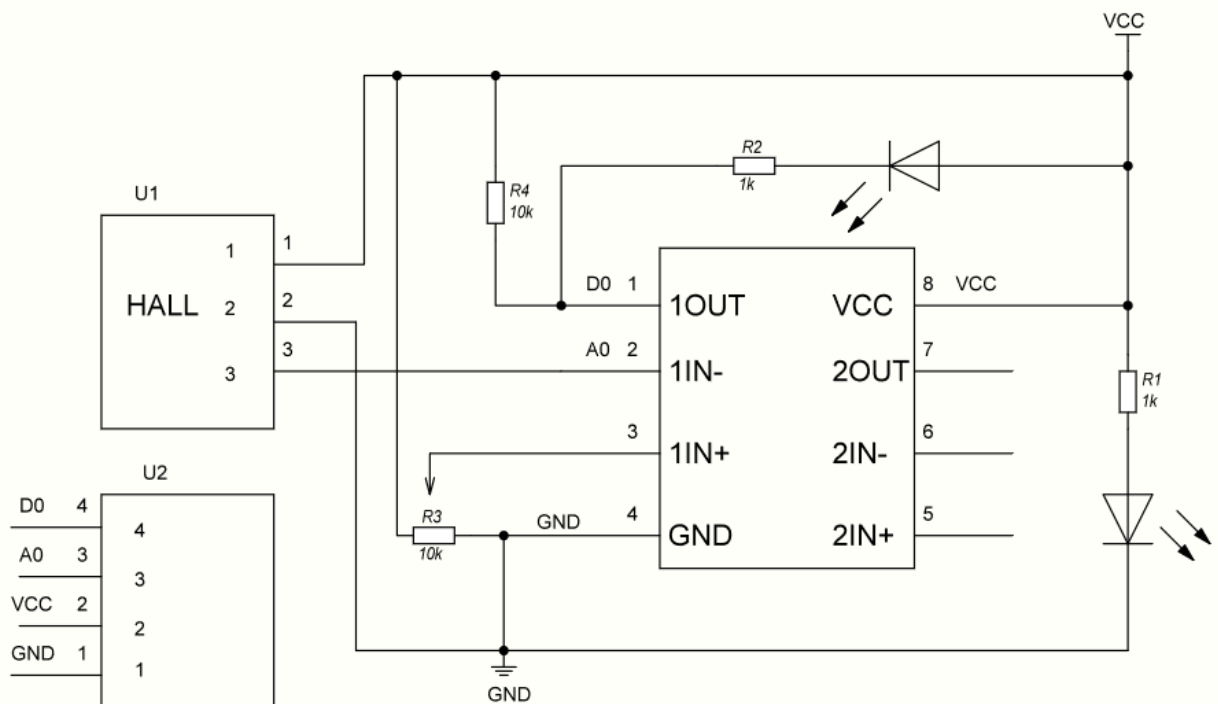


Рисунок 3.3 – Схема Датчика Холла

У цій схемі використовується датчик Холла для вимірювання обертів моторчика і визначення його швидкості обертання. Датчик Холла зазвичай складається з холлівського елемента, який реагує на зміни магнітного поля, і підсилювача, який підсилює сигнал від холлівського елемента для подальшого оброблення. Датчик Холла може бути активований зміною магнітного поля в його оточенні. У цій схемі датчик Холла використовується для вимірювання періоду обертів моторчика і визначення його швидкості обертання. Коли моторчик обертається, він створює зміни магнітного поля, які потім виявляються датчиком Холла. Коли датчик Холла виявляє зміну магнітного поля, він генерує імпульс, який використовується для визначення періоду обертів моторчика. Період обертів являє собою часовий інтервал між послідовними імпульсами і є прямим показником швидкості обертання моторчика. Що коротший період, то вища швидкість обертання моторчика. Сигнал від датчика Холла подається на мікроконтролер Arduino, який обробляє ці імпульси і використовує отриману

інформацію для управління відображенням анімації на світлодіодній стрічці. Мікроконтролер також може застосовувати фільтрацію і згладжування сигналу для більш стабільного вимірювання швидкості обертання моторчика. Таким чином, датчик Холла відіграє важливу роль у вимірюванні швидкості обертання моторчика і синхронізації відображення анімації на світлодіодній стрічці з його рухом. Це дає змогу створювати ефекти, що залежать від швидкості моторчика, і узгоджувати рух з анімацією, створюючи реалістичніші та ефективніші візуальні ефекти.

3.2.6 Двигун постійного струму

У цій схемі моторчик використовується для створення обертання, яке потім впливає на відображення анімації на світлодіодній стрічці. Моторчик може бути постійного струму (DC) або крокового (stepper) типу. У цьому випадку використовується моторчик постійного струму. Моторчик підключається до піна MOS (мосфет), який керує подачею живлення на моторчик. Керування моторчиком здійснюється за допомогою транзистора, підвищувального перетворювача і блока живлення 5 В. Транзистор відіграє роль ключа, який відкривається і закривається для управління подачею струму на моторчик. Коли транзистор відкритий, струм може протікати через моторчик, і він починає обертатися. Коли транзистор закритий, струм перестає протікати і моторчик зупиняється. Підвищувальний перетворювач використовується для підвищення напруги живлення до достатнього рівня для живлення моторчика. Оскільки блок живлення має напругу 5 В, яка може бути недостатньою для живлення моторчика, підвищувальний перетворювач збільшує цю напругу до необхідного рівня. Коли моторчик увімкнений, транзистор відкривається, і підвищувальний перетворювач подає достатню напругу на моторчик, щоб він почав обертатися. Регулювання швидкості моторчика може бути здійснене шляхом зміни ширини імпульсів (PWM - Pulse Width Modulation) сигналу, що подається на транзистор. Що ширші імпульси, то більша потужність і швидкість обертання моторчика. Таким чином, моторчик відіграє роль двигуна, який створює фізичний рух, а керування його обертанням і швидкістю здійснюється за допомогою транзистора, підвищувального перетворювача і блока

					ЕЛІТ 6.171.00.02.267 ПЗ	Арк
						25
Вим.	Арк.	№ докум.	Підпис	Дата		

живлення 5 В. Це дає змогу створювати різні ефекти та рухи в анімації на світлодіодній стрічці, пов'язані з роботою моторчика.

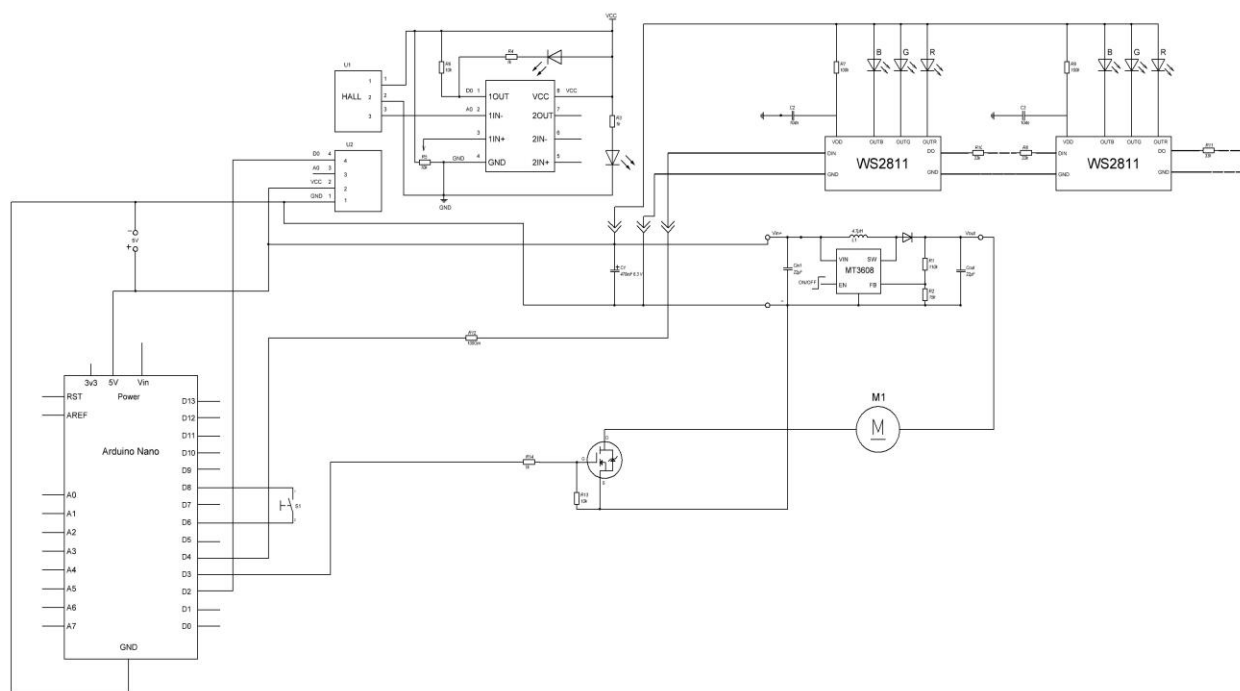


Рисунок 3.4 - Схема електрична принципова розробленого пристрою

4. РОЗРОБКА ПРОГРАМНОГО ЗАБЕЗПЕЧЕННЯ

Програма складена та налагоджена в безкоштовній середовищі розробки Arduino IDE. Код програми написаний на мові arduino, скетч займає 24,6 КБ пам'яті пристрою 81% від максимальної(30Кб). Глобальні змінні використовують 250 байт(12%) динамічної пам'яті, залишивши 1798 байт для локальних змінних, максимум - 2048 байт.

Мікроконтролер: ATmega328P - 8-бітний AVR мікроконтролер із тактовою частотою 16 МГц. Він має 32 кілобайти флеш-пам'яті для програмного коду, 2 кілобайти оперативної пам'яті (SRAM) і 1 кілобайт енергонезалежної пам'яті (EEPROM).

Arduino Nano V3.0 ATmega328P є популярною платою у світі DIY електроніки та прототипування завдяки своїй компактності, багатфункціональності та широкій підтримці спільноти Arduino.

Алгоритм управління містить наступні кроки:

1. Оголошення констант, змінних, підключення бібліотек і визначення функцій.
2. Налаштування та ініціалізація пристроїв та пінів.
3. Фільтрація помилкових спрацьовувань, запис нового періоду обороту.
4. Встановлення кольору пікселя на світлодіодній стрічці залежно від значення змінної.
5. Обробка переривань, анімації та керування мотором.

Програму можливо умовно розділити на функціональні блоки та розглянути їх функціонал, також присутні підпрограми для визначення таблиці гаммакорекції та масив значень кольорів для світлодіодів, які відправляються на стрічку. Далі розглянуті ключові моменти програмної частини проекту. Лістинг програми наведено в додатку А. Команди, що починаються з символу коментаря на початку рядка, призначені для полегшення розробки, налагодження і тестування.

Загалом, програму можна вважати оптимізованою з урахуванням своїх вимог і обмежень:

					ЕЛІТ 6.171.00.02.267 ПЗ	Арк
Вим.	Арк.	№ докум.	Підпис	Дата		27

- Код використовує константи і макроси для визначення параметрів, таких як кількість світлодіодів, яскравість, швидкість мотора та інші. Це дає змогу легко налаштувати ці значення без необхідності внесення змін до коду;
- Код використовує ключове слово `PROGMEM` для розміщення даних (кольорів пікселів) у пам'яті програми. Це допомагає економити оперативну пам'ять, звільняючи місце для інших операцій;
- Код використовує переривання для опрацювання сигналів з датчика Холла і таймера. Переривання дають змогу реагувати на події в режимі реального часу, мінімізуючи затримки та підвищуючи чуйність системи;
- Програма використовує бібліотеки `Adafruit_NeoPixel` і `TimerOne`, які надають оптимізовані функції для керування світлодіодною стрічкою і таймером;
- Код використовує цикли з поступовою зміною значення швидкості мотора для досягнення поступового прискорення і уповільнення. Це дає змогу більш плавно змінювати швидкість мотора й уникнути різких перепадів;

Програма взаємодіє з пам'яттю в кількох аспектах. Локальні змінні: У функціях `setup()`, `hall()`, `timer_interrupt()`, `setPix()`, `animation()`, `loop()` та інших використовуються локальні змінні, що зберігаються в стеку пам'яті. Це включає змінні типів `byte`, `int`, `boolean`, `uint16_t`, `uint8_t` та інші. Пам'ять для цих змінних виділяється і звільняється під час входу і виходу з відповідних функцій. Глобальні змінні: Код також використовує глобальні змінні, які зберігаються в сегменті даних програми. Сюди входять такі змінні, як `motor_state`, `timer`, `period`, `new_period`, `period_f`, `timer_isr`, `hall_isr`, `new_loop`, `counter1`, `counter2`, `counter`, `this_frame`, `count_frame` тощо. Глобальні змінні залишаються в пам'яті впродовж усього виконання програми і можуть бути доступні з різних функцій. Константи та макроси: Значення констант і макросів, як-от `NUM_LEDS`, `BRIGHTNESS`, `MOTOR_MAX`, `RES`, `COEF`, `PODGON`, `OFFSET`, `NUM_FRAMES`, `FRAME_RATE`, `PIN`, `MOS`, `BTN1`, `BTN1_G`, `BTN2`, `BTN2_G` та інші, зберігаються в сегменті констант програми (flash memory). Вони не змінюються під час виконання програми і доступні для читання. Дані з `PROGMEM`: Кольори пікселів для анімації

					ЕЛІТ 6.171.00.02.267 ПЗ	Арк
						28
Вим.	Арк.	№ докум.	Підпис	Дата		

зберігаються в спеціальному сегменті програмної пам'яті (PROGMEM). Для доступу до цих даних використовується функція `pgm_read_word()`, яка дає змогу зчитувати дані з пам'яті програми. Таким чином, програма використовує як статичну (глобальні змінні, константи), так і динамічну (локальні змінні) пам'ять під час виконання.

Першою функцією після об'явлення констант для різних параметрів та визначення пінів є допоміжна функція `expandColor()` яка використовується для перетворення кольірних значень пікселів із 16-бітного формату в 24-бітний формат. Це необхідно для правильного відображення кольорів на світлодіодній стрічці. Функція приймає 16-бітне значення `color` і повертає 24-бітне значення типу `uint32_t`, що представляє розширений колір. Перетворення відбувається в кілька кроків:

- `color >> 11` - операція зсуву бітів вправо на 11 позицій. Це використовується для отримання старших 5 бітів із `color`, які представляють червоний (Red) компонент кольору.
- `color >> 5 & 0x3F` - операція зсуву бітів вправо на 5 позицій, за якою слідує побітова маска `0x3F` (`0b00111111`). Це використовується для отримання наступних 6 бітів із `color`, які представляють зелений (Green) компонент кольору.
- `color & 0x1F` - побітова маска `0x1F` (`0b0001111111`) застосовується до `color` для отримання молодших 5 бітів, що представляють синій (Blue) компонент кольору.
- `pgm_read_byte()` - функція використовується для читання байта з пам'яті програми (PROGMEM). Вона приймає покажчик на адресу в пам'яті та повертає значення з цієї адреси.

Зібрані компоненти кольору об'єднуються за допомогою побітових операцій зсуву і побітового АБО, щоб отримати 24-бітне значення кольору. Червоний компонент займає біти 16-23, зелений компонент займає біти 8-15, а синій компонент займає біти 0-7. Розширений колір повертається як результат функції.

					<i>ЕЛІТ 6.171.00.02.267 ПЗ</i>	Арк
						29
<i>Вим.</i>	<i>Арк.</i>	<i>№ докум.</i>	<i>Підпис</i>	<i>Дата</i>		

Функція `expandColor()` використовується для перетворення колірних значень пікселів з пам'яті програми у формат, який може бути правильно відображений на світлодіодній стрічці.

Функція `setup()` - це спеціальна функція в Arduino, яка викликається один раз під час запуску програми. Вона використовується для ініціалізації початкових станів і налаштувань пристроїв і змінних перед виконанням основного циклу `loop()`. Функція `setup()` виконує наступні дії. Ініціалізує послідовний зв'язок через порт UART за допомогою функції `Serial.begin()`. Тут передається швидкість передавання даних 115200 біт на секунду, щоб установити з'єднання з монітором порту (комп'ютером) і забезпечити налагоджувальний вивід. Ініціалізує світлодіодну стрічку за допомогою функції `strip.begin()`. Це гарантує правильну роботу світлодіодної стрічки. Встановлює яскравість світлодіодної стрічки за допомогою функції `strip.setBrightness()`. Тут передається значення `BRIGHTNESS`, яке визначено в макросі та вказує бажану яскравість.

Налаштовує піни `BTN1`, `BTN2`, `BTN1_G`, `BTN2_G`, `MOS` за допомогою функції `pinMode()`. `BTN1` і `BTN2` налаштовуються як входи з підтяжкою до живлення, `BTN1_G` і `BTN2_G` налаштовуються як виходи. `MOS` налаштовується як вихід. Встановлює низький рівень сигналу на пінах `BTN1_G` і `BTN2_G` за допомогою функції `digitalWrite()`. Це може бути корисно для керування зовнішніми компонентами, пов'язаними з цими пінами. Прив'язує переривання до піна 0 (переривання 0) за допомогою функції `attachInterrupt()`. Тут вказується функція `hall()`, яка буде викликатися при зміні сигналу на `Rising Edge` (по передньому фронту сигналу). Ініціалізує таймер 1 за допомогою функції `Timer1.initialize()`. Таймер може використовуватися для виконання завдань за розкладом або з певною частотою. Зупиняє таймер 1 за допомогою функції `Timer1.stop()`. Це гарантує, що таймер не буде запущено до відповідного моменту.

Очищає світлодіодну стрічку за допомогою функції `strip.clear()`. Це встановлює всі пікселі світлодіодної стрічки в чорний колір (вимкнення світіння). Вмикає світлодіодну стрічку за допомогою функції `strip.show()`. Це застосовує всі

					<i>ЕЛІТ 6.171.00.02.267 ПЗ</i>	Арк
						30
Вим.	Арк.	№ докум.	Підпис	Дата		

зміни, зроблені на стрічці, і вмикає світіння пікселів. Функція `setup()` виконується один раз на початку роботи програми, щоб налаштувати необхідні параметри та ініціалізувати пристрої перед основним циклом `loop()`.

Функція `hall()` є перериванням, яке спрацьовує при зміні стану датчика Холла. У цій функції відбувається фільтрація помилкових спрацьовувань (якщо минуло достатньо часу з моменту попереднього спрацьовування), і потім записується новий період обороту. Після цього встановлюється прапор `hall_isr`, який вказує, що відбулося переривання. Функція `timer_interrupt()` також є перериванням і спрацьовує за таймером. У цій функції встановлюється прапор `timer_isr`, який вказує, що відбулося переривання за таймером.

Функція `setPix(byte i, int counter, int this_pix)` використовується для встановлення кольору пікселя світлодіодної стрічки на певній позиції і на основі заданих параметрів `counter` і `this_pix`. Використовуючи оператор `switch`, визначає поточний кадр анімації `this_frame`. Залежно від значення `this_frame`, виконається відповідна гілка `case`. У кожній гілці `case` використовується функція `strip.setPixelColor()` для встановлення кольору пікселя на позиції і світлодіодної стрічки. Колір пікселя визначається за допомогою функції `expandColor()`, яка перетворює 16-бітове значення кольору, зчитане з `PROGMEM`, у 24-бітове значення кольору. Значення кольору пікселя читається з відповідного масиву `frameX[counter][this_pix]`, де `X` - номер поточного кадру анімації, `counter` - значення лічильника, і `this_pix` - значення пікселя на поточній позиції. Функція `setPix()` використовується всередині функції `animation()`, щоб встановити кольори пікселів світлодіодної стрічки на основі поточного кадру анімації та значень лічильників. Це дає змогу створювати анімацію на світлодіодній стрічці, змінюючи кольори пікселів залежно від поточного стану анімації.

Функція `animation()` відповідає за оновлення анімації на світлодіодній стрічці. Скидання прапора переривання таймера `timer_isr`, щоб запобігти повторному виконанню функції в основному циклі програми. Очищення всіх пікселів на світлодіодній стрічці за допомогою функції `strip.clear()`. Це дає змогу оновити

анімацію, видаляючи попередні кольори пікселів. Змінна `this_pix` ініціалізується нульовим значенням. Потім, за допомогою циклу `for`, встановлюється колір пікселів для першої половини стрічки, починаючи з останнього пікселя і рухаючись до першого пікселя. Функція `setPix()` викликається для кожного пікселя, передаючи поточну позицію, значення лічильника `counter1` і значення `this_pix`. Змінна `this_pix` скидається в нуль. Потім, за допомогою другого циклу `for`, встановлюється колір пікселів для другої половини стрічки, починаючи з першого пікселя і рухаючись до останнього пікселя. Функція `setPix()` викликається для кожного пікселя, передаючи поточну позицію, значення лічильника `counter2` і значення `this_pix`. Значення лічильника `counter1` збільшується на одиницю, щоб перейти до наступного набору кольорів пікселів для першої половини стрічки. Значення лічильника `counter2` збільшується на одиницю, щоб перейти до наступного набору кольорів пікселів для другої половини стрічки. Якщо значення лічильника `counter2` перевищує межу `n_segm`, то лічильник скидається в нуль. Якщо значення лічильника `counter1` перевищує межу `n_segm`, то лічильник скидається в нуль. Функція `strip.show()` викликається для відображення оновленої анімації на світлодіодній стрічці. Це дає змогу побачити зміни кольорів пікселів, які були встановлені в попередніх кроках, і створює враження руху або зміни стану на стрічці. Функція `animation()` викликається в основному циклі програми та оновлює анімацію на світлодіодній стрічці відповідно до поточного стану лічильників і кадру анімації.

Функція `loop()` є головним циклом програми Arduino і виконується нескінченну кількість разів після виконання функції `setup()`. Якщо прапор `hall_isr` встановлено (відбулося переривання тахометра), то виконуються такі дії:

- Скидання прапора переривання `hall_isr`.
- Встановлення початкових значень лічильників `counter1` і `counter2` з урахуванням зсуву.
- Якщо значення `counter2` перевищує межу `n_segm`, то лічильник обнуляється.
- Збільшення значення лічильника кадрів `count_frame`.

					<i>ЕЛІТ 6.171.00.02.267 ПЗ</i>	Арк
<i>Вим.</i>	<i>Арк.</i>	<i>№ докум.</i>	<i>Підпис</i>	<i>Дата</i>		32

- Якщо значення `count_frame` перевищує або дорівнює `FRAME_RATE`, лічильник скидається в нуль, і вибирається наступний кадр анімації.
- Виклик функції `animation()` для оновлення анімації на світлодіодній стрічці.
- Обчислення нового періоду оновлення анімації на основі значення `new_period` і поточного значення `period_f` з використанням коефіцієнта `COEF`.
- Встановлення нового періоду для переривання таймера і перезапуск таймера.

Якщо прапор `timer_isr` встановлено (відбулося переривання таймера), то виконуються такі дії:

- Скидання прапора переривання `timer_isr`.
- Виклик функції `animation()` для оновлення анімації на світлодіодній стрічці.

Якщо кнопка `BTN1` натиснута, то виконуються такі дії:

- Інвертування стану `motor_state`.

Якщо `motor_state` істинний (мотор має бути ввімкнений), то виконуються такі дії:

- Плавне збільшення швидкості обертання мотора шляхом зміни значення ШІМ на виводі `MOS`.
- Підключення переривання таймера.

В іншому випадку (мотор має бути вимкнений) виконуються такі дії:

- Плавне зменшення швидкості обертання мотора.
- Зупинка мотора.
- Зупинка таймера і вимкнення переривання таймера.
- Затримка 500 мс.
- Скидання прапора переривання таймера.
- Очищення світлодіодної стрічки.
- Відображення порожньої стрічки.

Функція `loop()` перевіряє прапори переривань і стан кнопки, і залежно від цього виконує відповідні дії для поновлення анімації на світлодіодній стрічці, керування швидкістю мотора і відображення станів на стрічці.

ВИСНОВКИ

У рамках цієї дипломної роботи було розроблено голографічний пристрій відображення інформації, заснований на світлодіодній стрічці та контролері Arduino. Цей пристрій здатний створювати тривимірні ефекти, які надають зображенню глибину та об'єм.

Порівняно з іншими подібними конструкціями, розроблений голографічний пристрій має низку переваг. По-перше, він має високий ступінь гнучкості та налаштованості. За допомогою програмного коду можна легко змінювати і додавати нові анімаційні ефекти, а також налаштовувати параметри роботи пристрою під конкретні вимоги і переваги.

По-друге, розроблений пристрій має досить компактні розміри та низьку вартість виробництва порівняно з деякими іншими голографічними пристроями. Це робить його більш доступним і привабливим для широкої аудиторії користувачів.

Незважаючи на досягнуті результати, слід зазначити, що голографічні пристрої розвиваються і поліпшуються з кожним роком, і в майбутньому можлива поява нових технологій і конструкцій, що володіють ще більшою функціональністю і якістю відображення.

В процесі розробки було вивчено і опрацьовано велику кількість вітчизняних і зарубіжних джерел, в тому числі інформацію з мережі Інтернет, технічні характеристики і принцип дії пристроїв-аналогів. Результати роботи представлені у вигляді тез на конференції факультету ЕлІТ СумДУ «Фізика. Електроніка. Електротехніка 2023», електричної принципової схеми та лістингу програми керування пристроєм.

					<i>ЕлІТ 6.171.00.02.267 ПЗ</i>	Арк
						34
<i>Вим.</i>	<i>Арк.</i>	<i>№ докум.</i>	<i>Підпис</i>	<i>Дата</i>		

СПИСОК ЛІТЕРАТУРИ

1. Методичні вказівки до виконання кваліфікаційної роботи бакалавра зі спеціальності 0908 171 «ЕЛЕКТРОНІКА» освітньої програми «ЕЛЕКТРОНІ СИСТЕМИ ТА КОМПОНЕНТИ» /Укладачі: І.А. Кулик, А.І. Новгородцев, О.В. Бережная, В.В. Гриненко. – Суми: Вид-во СумДУ, 2018. – 18с.
2. Горячев О.Є. Подобрій А.Д. Голографічний пристрій відображення інформації. Матеріали конференції факультету ЕліТ СумДУ "Фізика. Електроніка. Електротехніка 2023" С. 85
3. Malacara, D. (2007). Optical Testing. Wiley.
4. Smith, J., & Johnson, A. "Holography: A Practical Approach" CRC Press, 2018.
5. Kasperczyk, P. (2016). Holography: Basic Principles and Contemporary Applications. Springer.
6. Hariharan, P. (2007). Optical Holography: Principles, Techniques, and Applications. Cambridge University Press
7. Cai, Y., & Poon, T.-C. (2013). Digital Holography and Three-Dimensional Display: Principles and Applications. Springer.
8. Smalley, D. E. (2016). The Handbook of Photonics. CRC Press.
9. Schouten, J. F., & Visser, T. D. (Eds.). (2012). Holography: Advances and Modern Trends. CRC Press.
10. Nishchal, N. K., & Joseph, J. (2019). Practical Holography XXIII: Materials and Applications. SPIE Press.
11. Yamaguchi, I., & Zhang, T. (2017). Phase-shifting Digital Holography: Practical Techniques and Applications. CRC Press.
12. Голографія - Енциклопедія Сучасної України URL: <https://esu.com.ua/article-25436> (дата звернення: 01.06.2023)

					ЕліТ 6.171.00.02.267 ПЗ	Арк
Вим.	Арк.	№ докум.	Підпис	Дата		35

ДОДАТОК А

```
#define NUM_LEDS 30 // количество светодиодов

#define BRIGHTNESS 200 // яркость

#define MOTOR_MAX 160 // скорость мотора (0-255)

#define RES 10 // разрешение (каждые n градусов)

#define COEF 0.3 // коэф. сглаживания частоты оборотов

#define PODGON 1.3 // коэф. подгона (γ, блет)

#define OFFSET 240 // сдвиг по углу (0-360)

#define NUM_FRAMES 8 // количество фреймов анимации

#define FRAME_RATE 60 // количество перерисовок между сменой кадра

#define PIN 4 // пин ленты

#define MOS 3 // пин мосфета (мотора)

#define BTN1 8 // пин кнопки 1

#define BTN1_G 6 // земля кнопки 1

#define BTN2 11 // пин кнопки 2

#define BTN2_G 9 // земля кнопки 2

#include "static.h"

#include <TimerOne.h>

#include "Adafruit_NeoPixel.h"

Adafruit_NeoPixel strip = Adafruit_NeoPixel(NUM_LEDS, PIN, NEO_GRB + NEO_KHZ800);

#include "gamma.h"

boolean motor_state = false;

volatile uint32_t timer;

volatile uint32_t period, new_period, period_f;

volatile boolean timer_isr, hall_isr, new_loop;
```

					<i>ЕЛІТ 6.171.00.02.267 ПЗ</i>	Арк
						36
<i>Вим.</i>	<i>Арк.</i>	<i>№ докум.</i>	<i>Підпис</i>	<i>Дата</i>		

```

uint16_t counter1, counter2;

uint8_t counter;

const byte n_segm = 360 / RES;

const byte offset = (float)OFFSET / 360 * n_segm;

//const byte offset = 0;

byte this_frame = 0, count_frame;

static uint32_t expandColor(uint16_t color) {
    return ((uint32_t)pgm_read_byte(&gamma5[ color >> 11 ]) << 16) |
        ((uint32_t)pgm_read_byte(&gamma6[(color >> 5) & 0x3F]) << 8) |
        pgm_read_byte(&gamma5[ color & 0x1F]);
}

void setup() {
    Serial.begin(115200);
    strip.begin();
    strip.setBrightness(BRIGHTNESS);

    pinMode(BTN1, INPUT_PULLUP);
    pinMode(BTN2, INPUT_PULLUP);
    pinMode(BTN1_G, OUTPUT);
    pinMode(BTN2_G, OUTPUT);
    digitalWrite(BTN1_G, LOW);
    digitalWrite(BTN2_G, LOW);
    pinMode(MOS, OUTPUT);

    attachInterrupt(0, hall, RISING);
    Timer1.initialize();
    Timer1.stop();

    strip.clear();
    strip.show();
}

```

					<i>ЕЛІТ 6.171.00.02.267 ПЗ</i>	Арк
<i>Вим.</i>	<i>Арк.</i>	<i>№ докум.</i>	<i>Підпис</i>	<i>Дата</i>		37

```

}

void hall() {
  if (micros() - timer > 30000) { // фильтруем ложные
    new_period = micros() - timer; // вот он вот он период оборота
    timer = micros();
    hall_isr = true;
  }
}

// прерывания таймера
void timer_interrupt() {
  timer_isr = true;
}

// вот тут у нас просиходит целая куча всего
// из PROGMEM получаем цвета пикселей с глубиной 16 бит, переводим в 24 бита (спасибо adafruit) и
включаем ленту
void setPix(byte i, int counter, int this_pix) {
  switch (this_frame) {
    case 0: strip.setPixelColor(i, expandColor(pgm_read_word(&(frame1[counter][this_pix]))));
      break;
    case 1: strip.setPixelColor(i, expandColor(pgm_read_word(&(frame1[counter][this_pix]))));
      break;
    case 2: strip.setPixelColor(i, expandColor(pgm_read_word(&(frame2[counter][this_pix]))));
      break;
    case 3: strip.setPixelColor(i, expandColor(pgm_read_word(&(frame3[counter][this_pix]))));
      break;
    case 4: strip.setPixelColor(i, expandColor(pgm_read_word(&(frame4[counter][this_pix]))));
      break;
    case 5: strip.setPixelColor(i, expandColor(pgm_read_word(&(frame5[counter][this_pix]))));
      break;
  }
}

```

					ЕЛІТ 6.171.00.02.267 ПЗ	Арк
Вим.	Арк.	№ докум.	Підпис	Дата		38


```

case 6: strip.setPixelColor(i, expandColor(pgm_read_word(&(frame6[counter][this_pix]))));
    break;
case 7: strip.setPixelColor(i, expandColor(pgm_read_word(&(frame7[counter][this_pix]))));
    break;
}
}

void animation() {
    timer_isr = false;
    strip.clear();

    byte this_pix = 0;
    for (int i = NUM_LEDS / 2 - 1; i >= 0; i--) {
        setPix(i, counter1, this_pix);
        this_pix++;
    }
    this_pix = 0;
    for (int i = NUM_LEDS / 2 ; i < NUM_LEDS; i++) {
        setPix(i, counter2, this_pix);
        this_pix++;
    }

    counter1++; // счётчик первой половины ленты
    counter2++; // счётчик второй половины
    if (counter2 > n_segm) counter2 = 0;
    if (counter1 > n_segm) counter1 = 0;

    strip.show();
}

void loop() {
    if (hall_isr) {

```

					ЕЛІТ 6.171.00.02.267 ПЗ	Арк
Вим.	Арк.	№ докум.	Підпис	Дата		39

```

hall_isr = false;
counter1 = 0 + offset;
counter2 = n_segm / 2 + offset;
if (counter2 > n_segm) counter2 = counter2 - n_segm;

// выбор кадра и скорости отображения
count_frame++;
if (count_frame >= FRAME_RATE) {
    count_frame = 0;
    this_frame = ++this_frame % NUM_FRAMES;
}

animation();

period_f = (float)new_period * COEF + (float)period_f * ((float)1 - COEF);
Timer1.setPeriod(period_f / n_segm * PODGON);
Timer1.restart();
//Serial.println(isr_counter2);
}

if (timer_isr) {
    timer_isr = false;
    animation();
}

if (!digitalRead(BTN1)) {
    motor_state = !motor_state;
    if (motor_state) {
        for (int i = 0; i < MOTOR_MAX; i++) {
            analogWrite(MOS, i);
            delay(20);
        }
        Timer1.attachInterrupt(timer_interrupt);
    }
}

```

					ЕЛІТ 6.171.00.02.267 ПЗ	Арк
Вим.	Арк.	№ докум.	Підпис	Дата		40

```

} else {
    for (int i = MOTOR_MAX; i > 0; i--) {
        analogWrite(MOS, i);
        delay(20);
    }
    analogWrite(MOS, 0);
    Timer1.stop();
    Timer1.detachInterrupt();
    delay(500);
    timer_isr = false;
    strip.clear();
    strip.show();
}
}
}

```

gamma.h

```

#ifndef _GAMMA_H_
#define _GAMMA_H_

```

```

#ifdef __AVR

```

```

#include <avr/pgmspace.h>

```

```

#elif defined(ESP8266)

```

```

#include <pgmspace.h>

```

```

#else

```

```

#ifndef PROGMEM

```

```

    #define PROGMEM

```

```

#endif

```

```

#endif

```

```

static const uint8_t PROGMEM

```

```

    gamma5[] = {

```

					ЕЛІТ 6.171.00.02.267 ПЗ	Арк
						41
Вим.	Арк.	№ докум.	Підпис	Дата		

```
0x00,0x01,0x02,0x03,0x05,0x07,0x09,0x0b,  
0x0e,0x11,0x14,0x18,0x1d,0x22,0x28,0x2e,  
0x36,0x3d,0x46,0x4f,0x59,0x64,0x6f,0x7c,  
0x89,0x97,0xa6,0xb6,0xc7,0xd9,0xeb,0xff },
```

```
gamma6[] = {
```

```
0x00,0x01,0x02,0x03,0x04,0x05,0x06,0x08,  
0x09,0x0a,0x0b,0x0d,0x0e,0x10,0x12,0x13,  
0x15,0x17,0x19,0x1b,0x1d,0x20,0x22,0x25,  
0x27,0x2a,0x2d,0x30,0x33,0x37,0x3a,0x3e,  
0x41,0x45,0x49,0x4d,0x52,0x56,0x5b,0x5f,  
0x64,0x69,0x6e,0x74,0x79,0x7f,0x85,0x8b,  
0x91,0x97,0x9d,0xa4,0xab,0xb2,0xb9,0xc0,  
0xc7,0xcf,0xd6,0xde,0xe6,0xee,0xf7,0xff };
```

```
#endif // _GAMMA_H_
```

satic.h

```
const uint16_t frame0[][36]PROGMEM = {
```

```
{0x4437, 0x4436, 0x4416, 0x4416, 0x3C16, 0x3BF6, 0x3BF6, 0x33D5, 0x33D5, 0x33B5, 0x2B95, 0x7D18,  
0xE71C, 0xE71C, 0xFE60, 0xFE81, 0xFE81, 0xFE81, 0xFE81, 0xFE81, 0xFE81, 0xFE81, 0xFE81, 0xFE81, 0xFE81,  
0xFE81, 0xFE81, 0xFE81, 0xFE61, 0xFE42, 0xF602, 0xEDE2, 0xE5A3, 0x7B01, 0x0000, 0x0000 }, //10 + 0  
градуса
```

```
{0x4437, 0x4416, 0x4416, 0x3BF6, 0x3BF6, 0x3BD5, 0x33D5, 0x33B5, 0x2BB5, 0x2B94, 0x2374, 0x7D17,  
0xE71C, 0xE71C, 0xFE60, 0xFE81, 0xFE81, 0xFE81, 0xFE81, 0xFE81, 0xFE81, 0xFE81, 0xFE81, 0xFE81, 0xFE81,  
0xFE81, 0xFE81, 0xFE61, 0xFE42, 0xFE22, 0xF602, 0xEDC3, 0xE583, 0x2920, 0x0000, 0x0000 }, //20 + 0  
градуса
```

```
{0x4436, 0x4416, 0x3BF6, 0x3BF6, 0x3BD5, 0x33D5, 0x33B5, 0x2B95, 0x2B94, 0x2374, 0x2354, 0x5C76,  
0xE71C, 0xE71B, 0xFE60, 0xFE81, 0xFE81, 0xFE81, 0xFE81, 0xFE81, 0xFE81, 0xFE81, 0xFE81, 0xFE81, 0xFE81,  
0xFE81, 0xFE81, 0xFE61, 0xFE42, 0xF622, 0xF5E2, 0xEDC3, 0xE583, 0xAC22, 0x0000, 0x0000 }, //30 + 0  
градуса
```

```
{0x4436, 0x4416, 0x3BF6, 0x3BD5, 0x33D5, 0x33B5, 0x2B95, 0x2B94, 0x2374, 0x2354, 0x1B33, 0x8D78,  
0xE71C, 0xE6FB, 0xFEA0, 0xFE81, 0xFE81, 0xFE81, 0xFE81, 0xFE81, 0xFE81, 0xFE81, 0xFE81, 0xFE81, 0xFE81,  
0xFE81, 0xFE61, 0xFE62, 0xFE42, 0xF602, 0xF5E2, 0xEDA3, 0xE583, 0x0000, 0x0000, 0x0000 }, //40 + 0  
градуса
```

					ЕЛІТ 6.171.00.02.267 ПЗ	Арк
						42
Вим.	Арк.	№ докум.	Підпис	Дата		

0x4DA9, 0x4DA9, 0x4D89, 0x4D89, 0x4568, 0x4548, 0x4528, 0x4508, 0x44E8, 0x44E8, 0x0000, 0x0000 },
//150 + 0 градуса

{0x4437, 0x4436, 0x4416, 0x3BF6, 0x3BF6, 0x3BF6, 0x33D5, 0x33D5, 0x33B5, 0x2B94, 0x2B74, 0x2374,
0xFFDD, 0xE71C, 0xEF3D, 0x4DA9, 0x4DA9, 0x4DA9, 0x4DA9, 0x4DA9, 0x4DA9, 0x4DA9, 0x4DA9, 0x4DA9,
0x4DA9, 0x4DA9, 0x4DA9, 0x4D89, 0x4D68, 0x4548, 0x4528, 0x4508, 0x4508, 0x44C8, 0x0000, 0x0000 },
//160 + 0 градуса

{0x4437, 0x4436, 0x4416, 0x4416, 0x3C16, 0x3BF6, 0x3BF6, 0x33D5, 0x33D5, 0x33B5, 0x33B5, 0x2B95,
0xF77D, 0xE71C, 0xEF7D, 0xF801, 0x4DA9, 0x4DA9, 0x4DA9, 0x4DA9, 0x4DA9, 0x4DA9, 0x4DA9, 0x4DA9,
0x4DA9, 0x4DA9, 0x4DA9, 0x4D89, 0x4D69, 0x4548, 0x4548, 0x4528, 0x4508, 0x44E8, 0x0000, 0x0000 },
//170 + 0 градуса

{0x4437, 0x4437, 0x4437, 0x4436, 0x4416, 0x4416, 0x3C16, 0x3C16, 0x3BF6, 0x3BF6, 0x33D5, 0x33D5,
0xFF9D, 0xE71C, 0xEF7D, 0xD862, 0xC904, 0x4DA9, 0x4DA9, 0x4DA9, 0x4DA9, 0x4DA9, 0x4DA9, 0x4DA9,
0x4DA9, 0x4DA9, 0x4DA9, 0x4DA9, 0x4D89, 0x4D68, 0x4548, 0x4528, 0x4508, 0x44E8, 0x0000, 0x0000 },
//180 + 0 градуса

{0x4437, 0x4437, 0x4437, 0x4437, 0x4437, 0x4437, 0x4437, 0x4436, 0x4416, 0x4416, 0x3BF6, 0x3BF6,
0xC67B, 0xE73C, 0xE73C, 0xB800, 0xC8E4, 0xC8E4, 0xC904, 0x36AA, 0x4DA9, 0x4DA9, 0x4DA9, 0x4DA9,
0x4DA9, 0x4DA9, 0x4DA9, 0x4DA9, 0x4D89, 0x4D69, 0x4568, 0x4548, 0x4528, 0x4508, 0x0000, 0x0000 },
//190 + 0 градуса

{0x4437, 0x4C57, 0x4C57, 0x4C57, 0x4C57, 0x4C57, 0x4C57, 0x4C57, 0x4C57, 0x4C57, 0x4437, 0x4437,
0xA5FA, 0xEF3C, 0xEF3C, 0xD000, 0xC8E4, 0xC8E4, 0xC8E4, 0xC8E4, 0xC8E4, 0xC8E4, 0xD0A4, 0x3689, 0x4DA9,
0x4DA9, 0x4DA9, 0x4DA9, 0x4DA9, 0x4D89, 0x4D69, 0x4548, 0x4528, 0x4528, 0x0000, 0x0000 }, //200 + 0
градуса

{0x4C57, 0x4C57, 0x4C57, 0x4C77, 0x4C77, 0x4C77, 0x5478, 0x5478, 0x5477, 0x5477, 0x4C77, 0x4C77,
0xD6FC, 0xEF3C, 0xEF3C, 0xE021, 0xD904, 0xD904, 0xD904, 0xD904, 0xE104, 0xD904, 0xC8E4, 0xC8E4, 0xC8E4,
0xC8E4, 0xC8E4, 0xC8E4, 0xC904, 0xC904, 0x4D89, 0x4D68, 0x4548, 0x4528, 0x0080, 0x0000 }, //210 + 0
градуса

{0x4C57, 0x4C57, 0x4C77, 0x5478, 0x5498, 0x5498, 0x5498, 0x5498, 0x5CB8, 0x5CB8, 0x5CB8, 0x5498,
0xEF5D, 0xEF5C, 0xEF3C, 0xD800, 0xD904, 0xD904, 0xD904, 0xD904, 0xD904, 0xD904, 0xD904, 0xD904,
0xD904, 0xD904, 0xD904, 0xD904, 0xD904, 0xD904, 0xD904, 0xD904, 0x0000, 0x0000 },
//220 + 0 градуса

{0x4C57, 0x4C57, 0x5478, 0x5498, 0x5498, 0x5CB8, 0x5CB8, 0x5CD9, 0x5CD9, 0x5CD9, 0x64D9, 0x64D9,
0xF77D, 0xEF5D, 0xEF3C, 0xD800, 0xD904, 0xD904, 0xD904, 0xD904, 0xD904, 0xD904, 0xD904, 0xD904,
0xE125, 0xE125, 0xE125, 0xE125, 0xE145, 0xE145, 0xE145, 0xE145, 0xE145, 0xE145, 0x0000, 0x0000 }, //230
+ 0 градуса

{0x4C57, 0x4C77, 0x5478, 0x5498, 0x5CB8, 0x5CD9, 0x5CD9, 0x64F9, 0x64F9, 0x6519, 0x6D1A, 0x6D1A,
0xDF1C, 0xF75D, 0xF75D, 0xD842, 0xD904, 0xD904, 0xD904, 0xE125, 0xE125, 0xE145, 0xE165, 0xE165, 0xE186,
0xE186, 0xE1A6, 0xE1A6, 0xE1A6, 0xE1C6, 0xE1C6, 0xE1C7, 0xE1C7, 0xE1C7, 0x2041, 0x0000 }, //240 + 0
градуса

{0x4C57, 0x4C77, 0x5498, 0x54B8, 0x5CB8, 0x5CD9, 0x64F9, 0x64F9, 0x6D1A, 0x6D3A, 0x753A, 0x755A,
0xCEBC, 0xF77D, 0xF77D, 0xD821, 0xD905, 0xE125, 0xE145, 0xE165, 0xE186, 0xE186, 0xE1A6, 0xE1C6, 0xE1C7,
0xE1E7, 0xE207, 0xE207, 0xE227, 0xE228, 0xEA48, 0xEA48, 0xEA48, 0xEA48, 0x0000, 0x0000 }, //250 + 0
градуса

					ЕЛІТ 6.171.00.02.267 ПЗ	Арк
Вим.	Арк.	№ докум.	Підпис	Дата		44

0xFE81, 0xFE81, 0xFE81, 0xFE61, 0xFE42, 0xF622, 0xF5E2, 0xEDC3, 0x7AE1, 0x0000, 0x0000 }, //360 + 0
градуса

{0x4437, 0x4436, 0x4416, 0x4416, 0x3C16, 0x3BF6, 0x3BF6, 0x33D5, 0x33D5, 0x33B5, 0x2B95, 0x7D18,
0xE71C, 0xE71C, 0xFE60, 0xFE81, 0xFE81, 0xFE81, 0xFE81, 0xFE81, 0xFE81, 0xFE81, 0xFE81, 0xFE81,
0xFE81, 0xFE81, 0xFE81, 0xFE61, 0xFE42, 0xF602, 0xEDE2, 0xE5A3, 0x7B01, 0x0000, 0x0000 }, //370 + 0
градуса

};

const uint16_t frame1[][36]PROGMEM = {

{0xCEDB, 0xCEDB, 0xCEDB, 0xCEDB, 0x855A, 0x1338, 0x1338, 0x1338, 0x1338, 0x1338, 0x1338, 0x1338,
0x1338, 0xD964, 0xD165, 0xD165, 0xD165, 0xD165, 0xD165, 0xD104, 0xCEFC, 0xCEDB, 0xCEDB, 0xCEDB,
0xCEDB, 0xCEDB, 0xCF5D, 0xD145, 0xD165, 0xD165, 0xD165, 0xD165, 0xD165, 0xD165, 0x0000, 0x0000 },
//10 + 0 градуса

{0xCEDB, 0xCEDB, 0xCEDB, 0xCEDB, 0xEF5C, 0x0AF7, 0x1338, 0x1338, 0x1338, 0x1338, 0x1338, 0x1338,
0x1338, 0xE943, 0xD165, 0xD165, 0xD165, 0xD165, 0xD165, 0xD165, 0xCEDB, 0xCEDB, 0xCEDB, 0xCEDB,
0xCEDB, 0xCEDB, 0xCF5D, 0xD165, 0xD165, 0xD165, 0xD165, 0xD165, 0xD165, 0xD165, 0x0000, 0x0000 },
//20 + 0 градуса

{0xCEDB, 0xCEDB, 0xCEDB, 0xCEDB, 0xCEDB, 0xAE1A, 0x0AF8, 0x2398, 0x1337, 0x1338, 0x1B58, 0x1317,
0x1317, 0xE144, 0xD165, 0xD165, 0xD165, 0xD165, 0xD165, 0xD104, 0xCF7E, 0xCE9B, 0xCE9A, 0xCEBB,
0xCEBB, 0xCE9B, 0xCF5D, 0xD186, 0xD165, 0xD165, 0xD165, 0xD165, 0xD165, 0xD165, 0x0000, 0x0000 }, //30
+ 0 градуса

{0xCEDB, 0xCEDB, 0xCEDB, 0xCEDB, 0xCEDB, 0xCEDB, 0xD6DB, 0x33D8, 0x2B98, 0x2BB8, 0x2BB8, 0x2BB8,
0x33B8, 0xF984, 0xE1C6, 0xE1C6, 0xE1C6, 0xE1C6, 0xE1C6, 0xE1C6, 0xFFFF, 0xFFDF, 0xFFDF, 0xFFDF,
0xFFDF, 0xFFDF, 0xE1C6, 0xE1C6, 0xE1C6, 0xE1C6, 0xE1C6, 0xE1C6, 0xE1C6, 0x0000, 0x0000 }, //40 + 0
градуса

{0xCEDB, 0xCEDB, 0xCEDB, 0xCEDB, 0xCEDB, 0xCEDB, 0xCEDB, 0xCEDB, 0xCEDB, 0xE73B, 0x64B9, 0x1B58,
0x33B8, 0xF963, 0xE1C6, 0xE1C6, 0xE1C6, 0xE1C6, 0xE1C6, 0xE1C6, 0xFFFF, 0xFFDF, 0xFFDF, 0xFFDF, 0xFFDF,
0xFFDF, 0xFFDF, 0xE1C6, 0xE1C6, 0xE1C6, 0xE1C6, 0xE1C6, 0xE1C6, 0xE9C6, 0x0000, 0x0000 }, //50 + 0
градуса

{0xCEDB, 0xCEDB, 0xCEDB, 0xCEDB, 0xCEDB, 0xCEDB, 0xCEDB, 0xCEDB, 0xE73B, 0x1318, 0x2B98, 0x1338,
0x1B38, 0xD965, 0xD165, 0xD165, 0xD165, 0xD165, 0xD165, 0xD104, 0xCF7E, 0xCE9A, 0xCE9B, 0xCEBB,
0xCEBB, 0xCE9B, 0xCF5D, 0xD186, 0xD165, 0xD165, 0xD165, 0xD165, 0xD165, 0xD165, 0x0000, 0x0000 }, //60
+ 0 градуса

{0xCEDB, 0xCEDB, 0xCEDB, 0xCEDB, 0xCEDB, 0xCEDB, 0xCEDB, 0xCEDB, 0xCEDB, 0xE73B, 0x1338, 0x1338,
0x1338, 0x1338, 0x1338, 0xE943, 0xD165, 0xD165, 0xD165, 0xD165, 0xD165, 0xD165, 0xCEDB, 0xCEDB, 0xCEDB,
0xCEDB, 0xCEDB, 0xCEDB, 0xCF5D, 0xD165, 0xD165, 0xD165, 0xD165, 0xD165, 0xD165, 0x0000, 0x0000 },
//70 + 0 градуса

{0xCEDB, 0xCEDB, 0xCEDB, 0xCEDB, 0xEF5C, 0x0297, 0x1338, 0x1338, 0x1338, 0x1338, 0x1338, 0x1338,
0x1338, 0xD964, 0xD165, 0xD165, 0xD165, 0xD165, 0xD165, 0xD104, 0xCEFC, 0xCEDB, 0xCEDB, 0xCEDB,
0xCEDB, 0xCEDB, 0xCEDB, 0xCF5D, 0xD145, 0xD165, 0xD165, 0xD165, 0xD165, 0xD165, 0x0000, 0x0000 },
//80 + 0 градуса

					ЕЛІТ 6.171.00.02.267 ПЗ	Арк
						46
Вим.	Арк.	№ докум.	Підпис	Дата		

{0xCEDB, 0xCEDB, 0xCEDB, 0xCEDB, 0x0297, 0x1338, 0x1338, 0x1338, 0x1338, 0x1338, 0x1338, 0x1338, 0x1338, 0xE943, 0xD165, 0xD165, 0xD165, 0xD165, 0xD165, 0xD124, 0xCF1C, 0xCEDB, 0xCEDB, 0xCEDB, 0xCEDB, 0xCEDB, 0xD165, 0xD165, 0xD165, 0xD165, 0xD165, 0xD165, 0xD165, 0x0000, 0x0000 },
//90 + 0 градуса

{0xCEDB, 0xCEDB, 0xCEDB, 0xCEDB, 0xE71B, 0x02D7, 0x1338, 0x1338, 0x1338, 0x1338, 0x1338, 0x1338, 0x1338, 0xE144, 0xD165, 0xD165, 0xD165, 0xD165, 0xD165, 0xD124, 0xCF1C, 0xCEDB, 0xCEDB, 0xCEDB, 0xCEDB, 0xCEDB, 0xCF1C, 0xD104, 0xD165, 0xD165, 0xD165, 0xD165, 0xD165, 0xD165, 0x0000, 0x0000 },
//100 + 0 градуса

{0xCEDB, 0xCEDB, 0xCEDB, 0xCEDB, 0xCEDB, 0xDEFB, 0x1338, 0x1338, 0x1338, 0x1338, 0x1338, 0x1338, 0x1338, 0xC187, 0xD165, 0xD165, 0xD165, 0xD165, 0xD165, 0xD165, 0xCF1C, 0xCEDB, 0xCEDB, 0xCEDB, 0xCEDB, 0xCEDB, 0xCEDB, 0xCEFC, 0xD145, 0xD165, 0xD165, 0xD165, 0xD165, 0xD165, 0xD165, 0x0000, 0x0000 },
//110 + 0 градуса

{0xCEDB, 0xCEDB, 0xCEDB, 0xCEDB, 0xCEDB, 0xCEDB, 0xCEDB, 0xCEDB, 0xDF1B, 0x02B7, 0x1338, 0x1338, 0x1338, 0xE943, 0xD165, 0xD165, 0xD165, 0xD165, 0xD165, 0xD186, 0xD618, 0xCEDB, 0xCEDB, 0xCEDB, 0xCEDB, 0xCEDB, 0xCEDB, 0xD124, 0xD165, 0xD165, 0xD165, 0xD165, 0xD165, 0xD165, 0x0000, 0x0000 },
//120 + 0 градуса

{0xCEDB, 0xCEDB, 0xCEDB, 0xCEDB, 0xCEDB, 0xCEDB, 0xCEDB, 0xCEDB, 0xCEDB, 0xE73B, 0x6CD9, 0x02B7, 0x1338, 0xB9A7, 0xD165, 0xD165, 0xD165, 0xD165, 0xD165, 0xD165, 0xCFBF, 0xCEDB, 0xCEDB, 0xCEDB, 0xCEDB, 0xCEDB, 0xCEDB, 0xD0E3, 0xD165, 0xD165, 0xD165, 0xD165, 0xD165, 0xD165, 0x0000, 0x0000 },
//130 + 0 градуса

{0xCEDB, 0xCEDB, 0xCEDB, 0xCEDB, 0xCEDB, 0xCEDB, 0xD6DB, 0x1B58, 0x0AF7, 0x1338, 0x1338, 0x1338, 0x1338, 0xA1EA, 0xD165, 0xD165, 0xD165, 0xD165, 0xD165, 0xD165, 0xCF7E, 0xCEDB, 0xCEDB, 0xCEDB, 0xCEDB, 0xCEDB, 0xCEDB, 0xCEBB, 0xD0A2, 0xD165, 0xD165, 0xD165, 0xD165, 0xD165, 0x0000, 0x0000 },
//140 + 0 градуса

{0xCEDB, 0xCEDB, 0xCEDB, 0xCEDB, 0xCEDB, 0xA5FA, 0x0297, 0x1338, 0x1338, 0x1338, 0x1338, 0x1338, 0x1338, 0xE943, 0xD165, 0xD165, 0xD165, 0xD165, 0xD165, 0xD186, 0xD3F0, 0xCEDB, 0xCEDB, 0xCEDB, 0xCEDB, 0xCEDB, 0xCEDB, 0xCEBB, 0xD0A2, 0xD165, 0xD165, 0xD165, 0xD165, 0xD165, 0xD165, 0x0000, 0x0000 },
//150 + 0 градуса

{0xCEDB, 0xCEDB, 0xCEDB, 0xCEDB, 0xEF5C, 0x0B17, 0x1338, 0x1338, 0x1338, 0x1338, 0x1338, 0x1338, 0x1338, 0x5A91, 0xD165, 0xD165, 0xD165, 0xD165, 0xD165, 0xD165, 0xCF5E, 0xCEDB, 0xCEDB, 0xCEDB, 0xCEDB, 0xCEDB, 0xCEDB, 0xCEBB, 0xD0A2, 0xD165, 0xD165, 0xD165, 0xD165, 0xD165, 0xD165, 0x0000, 0x0000 },
//160 + 0 градуса

{0xCEDB, 0xCEDB, 0xCEDB, 0xCEDB, 0x6CD9, 0x1338, 0x1338, 0x1338, 0x1338, 0x1338, 0x1338, 0x1338, 0x1338, 0xD186, 0xD165, 0xD165, 0xD165, 0xD165, 0xD165, 0xD165, 0xCEDB, 0xCEDB, 0xCEDB, 0xCEDB, 0xCEDB, 0xCEDB, 0xCEDB, 0xD165, 0xD165, 0xD165, 0xD165, 0xD165, 0xD165, 0x0000, 0x0000 },
//170 + 0 градуса

{0xCEDB, 0xCEDB, 0xCEDB, 0xCEDB, 0xCEDB, 0xD6DB, 0x1338, 0x1338, 0x1338, 0x1338, 0x1338, 0x1338, 0x1338, 0x5291, 0xD165, 0xD165, 0xD165, 0xD165, 0xD165, 0xD165, 0xCE9A, 0xCEDB, 0xCEDB, 0xCEDB, 0xCEDB, 0xCEDB, 0xCEDB, 0xD0A2, 0xD165, 0xD165, 0xD165, 0xD165, 0xD165, 0xD165, 0x0000, 0x0000 },
//180 + 0 градуса

{0xCEDB, 0xCEDB, 0xCEDB, 0xCEDB, 0xCEDB, 0xCEDB, 0xCEBB, 0x7519, 0x1338, 0x1338, 0x1338, 0x1338, 0x1338, 0xB9A7, 0xD165, 0xD165, 0xD165, 0xD165, 0xD165, 0xD165, 0xD67A, 0xCEDB, 0xCEDB, 0xCEDB, 0xCEDB, 0xCEDB, 0xCEDB, 0xCEDB, 0xCEDB, 0xCEDB, 0xCEDB, 0x0000, 0x0000 },
//190 + 0 градуса

					ЕЛІТ 6.171.00.02.267 ПЗ	Арк
						47
Вим.	Арк.	№ докум.	Підпис	Дата		

0xCEDB, 0xCEDB, 0xCEDB, 0xD228, 0xD165, 0xD165, 0xD165, 0xD165, 0xD165, 0xD165, 0x0000, 0x0000 },
//190 + 0 градуса

{0xCEDB, 0xCEDB, 0xCEDB, 0xCEDB, 0xCEDB, 0xCEDB, 0xCEDB, 0xCEDB, 0xCEDB, 0xCEDB, 0xCEDB, 0xDF1B, 0xA5FA,
0x02B7, 0x1B17, 0xD165, 0xD165, 0xD165, 0xD165, 0xD165, 0xD165, 0xCE9A, 0xCEDB, 0xCEDB, 0xCEDB,
0xCEDB, 0xCEDB, 0xCEDB, 0xD124, 0xD165, 0xD165, 0xD165, 0xD165, 0xD165, 0xD165, 0x0000, 0x0000 },
//200 + 0 градуса

{0xCEDB, 0xCEDB, 0xCEDB, 0xCEDB, 0xCEDB, 0xCEDB, 0xCEDB, 0xCEDB, 0xD6DB, 0x4C59, 0x1317, 0x1338,
0x1338, 0x99EB, 0xD165, 0xD165, 0xD165, 0xD165, 0xD165, 0xD165, 0xD144, 0xCEBB, 0xCEBB, 0xC69A,
0xCEDB, 0xCEBB, 0xCEBB, 0xD104, 0xD165, 0xD165, 0xD165, 0xD165, 0xD165, 0xD165, 0x20A1, 0x0000 },
//210 + 0 градуса

{0xCEDB, 0xCEDB, 0xCEDB, 0xCEDB, 0xCEDB, 0xCEDB, 0xD6DB, 0x64B9, 0x2BB8, 0x2BB8, 0x2BB8, 0x2BB8,
0x2BB8, 0x1BDA, 0xE1C6, 0xE1C6, 0xE1C6, 0xE1C6, 0xE1C6, 0xE1C6, 0xE1C6, 0xECD2, 0xFFDF, 0xFFDF, 0xFFDF, 0xFFDF,
0xFFDF, 0xFFDF, 0xE1E7, 0xE1C6, 0xE1C6, 0xE1C6, 0xE1C6, 0xE1C6, 0xE1C6, 0x0000, 0x0000 }, //220 + 0
градуса

{0xCEDB, 0xCEDB, 0xCEDB, 0xCEDB, 0xCEDB, 0xDF1B, 0x1B58, 0x2BB8, 0x2BB8, 0x2BB8, 0x2BB8, 0x2BB8,
0x2BB8, 0x141B, 0xE1C6, 0xE1C6, 0xE1C6, 0xE1C6, 0xE1C6, 0xE1C6, 0xEC92, 0xFFDF, 0xFFDF, 0xFFDF, 0xFFDF,
0xFFDF, 0xFFDF, 0xE1E7, 0xE1C6, 0xE1C6, 0xE1C6, 0xE1C6, 0xE1C6, 0xE1C6, 0x0000, 0x0000 }, //230 + 0
градуса

{0xCEDB, 0xCEDB, 0xCEDB, 0xCEDB, 0xCEDB, 0x0AF7, 0x2378, 0x2378, 0x1338, 0x1338, 0x1B58, 0x1317,
0x1317, 0x920C, 0xD165, 0xD165, 0xD165, 0xD165, 0xD165, 0xD165, 0xD124, 0xCEBB, 0xCEBB, 0xC69A,
0xCEDB, 0xCEBB, 0xCEBB, 0xD104, 0xD165, 0xD165, 0xD165, 0xD165, 0xD165, 0xD165, 0xD165, 0x4082, 0x0000 },
//240 + 0 градуса

{0xCEDB, 0xCEDB, 0xCEDB, 0xCEDB, 0xCEDB, 0xCEDB, 0x959A, 0x1338, 0x1338, 0x1338, 0x1338, 0x1338,
0x1338, 0x1B17, 0xD165, 0xD165, 0xD165, 0xD165, 0xD165, 0xD165, 0xCE9A, 0xCEDB, 0xCEDB, 0xCEDB,
0xCEDB, 0xCEDB, 0xCEDB, 0xD124, 0xD165, 0xD165, 0xD165, 0xD165, 0xD165, 0xD165, 0x0000, 0x0000 },
//250 + 0 градуса

{0xCEDB, 0xCEDB, 0xCEDB, 0xCEDB, 0xCEDB, 0xCEDB, 0xCEDB, 0xDF1B, 0xCEBB, 0x02D7, 0x1338, 0x1338,
0x1338, 0xB9A7, 0xD165, 0xD165, 0xD165, 0xD165, 0xD165, 0xD165, 0xCE9A, 0xCEDB, 0xCEDB, 0xCEDB,
0xCEDB, 0xCEDB, 0xCEDB, 0xD228, 0xD165, 0xD165, 0xD165, 0xD165, 0xD165, 0xD165, 0x0000, 0x0000 },
//260 + 0 градуса

{0xCEDB, 0xCEDB, 0xCEDB, 0xCEDB, 0xCEDB, 0xCEDB, 0xCEDB, 0xCEDB, 0xCEDB, 0xCEDB, 0xCEDB, 0xEF5C,
0x8D7A, 0x5291, 0xD165, 0xD165, 0xD165, 0xD165, 0xD165, 0xD165, 0xCE9A, 0xCEDB, 0xCEDB, 0xCEDB,
0xCEDB, 0xCEDB, 0xCEDB, 0xD0A2, 0xD165, 0xD165, 0xD165, 0xD165, 0xD165, 0xD165, 0x0000, 0x0000 },
//270 + 0 градуса

{0xCEDB, 0xCEDB, 0xCEDB, 0xCEDB, 0xCEDB, 0xCEDB, 0xCEDB, 0xEF5C, 0x5479, 0x1338, 0x1338, 0x1338,
0x1338, 0xD166, 0xD165, 0xD165, 0xD165, 0xD165, 0xD165, 0xD165, 0xCEDB, 0xCEDB, 0xCEDB, 0xCEDB,
0xCEDB, 0xCEDB, 0xCEDB, 0xD165, 0xD165, 0xD165, 0xD165, 0xD165, 0xD165, 0xD165, 0x0000, 0x0000 },
//280 + 0 градуса

{0xCEDB, 0xCEDB, 0xCEDB, 0xCEDB, 0xCEDB, 0xD6FB, 0x2BB8, 0x1338, 0x1338, 0x1338, 0x1338, 0x1338,
0x1338, 0x5A91, 0xD165, 0xD165, 0xD165, 0xD165, 0xD165, 0xD165, 0xCF5E, 0xCEDB, 0xCEDB, 0xCEDB,
0xCEDB, 0xCEDB, 0xCEDB, 0xD0A2, 0xD165, 0xD165, 0xD165, 0xD165, 0xD165, 0xD165, 0x0000, 0x0000 },
//290 + 0 градуса

					ЕЛІТ 6.171.00.02.267 ПЗ	Арк
Вим.	Арк.	№ докум.	Підпис	Дата		48

```
{0xCEDB, 0xCEDB, 0xCEDB, 0xCEDB, 0xCEDB, 0x0237, 0x1338, 0x1338, 0x1338, 0x1338, 0x1338, 0x1338,
0x1338, 0xE943, 0xD165, 0xD165, 0xD165, 0xD165, 0xD165, 0xD186, 0xD3EF, 0xCEDB, 0xCEDB, 0xCEDB,
0xCEDB, 0xCEDB, 0xCEBB, 0xD0A2, 0xD165, 0xD165, 0xD165, 0xD165, 0xD165, 0xD165, 0x0000, 0x0000 },
//300 + 0 градуса
```

```
{0xCEDB, 0xCEDB, 0xCEDB, 0xCEDB, 0xCEDB, 0xE71B, 0x02B7, 0x1338, 0x1338, 0x1338, 0x1338, 0x1338,
0x1338, 0xA1EA, 0xD165, 0xD165, 0xD165, 0xD165, 0xD165, 0xD165, 0xCF7E, 0xCEDB, 0xCEDB, 0xCEDB,
0xCEDB, 0xCEDB, 0xCEBB, 0xD0A2, 0xD165, 0xD165, 0xD165, 0xD165, 0xD165, 0xD165, 0x0000, 0x0000 },
//310 + 0 градуса
```

```
{0xCEDB, 0xCEDB, 0xCEDB, 0xCEDB, 0xCEDB, 0xCEDB, 0xD6DB, 0x5459, 0x1338, 0x1338, 0x1338, 0x1338,
0x1338, 0xB9A7, 0xD165, 0xD165, 0xD165, 0xD165, 0xD165, 0xD165, 0xCFBF, 0xCEDB, 0xCEDB, 0xCEDB,
0xCEDB, 0xCEDB, 0xCEDB, 0xD0E3, 0xD165, 0xD165, 0xD165, 0xD165, 0xD165, 0xD165, 0x0000, 0x0000 },
//320 + 0 градуса
```

```
{0xCEDB, 0xCEDB, 0xCEDB, 0xCEDB, 0xCEDB, 0xCEDB, 0xCEDB, 0xCEDB, 0xD6DB, 0x5459, 0x1338, 0x1338,
0x1338, 0xE943, 0xD165, 0xD165, 0xD165, 0xD165, 0xD165, 0xD186, 0xD618, 0xCEDB, 0xCEDB, 0xCEDB,
0xCEDB, 0xCEDB, 0xCEDB, 0xD124, 0xD165, 0xD165, 0xD165, 0xD165, 0xD165, 0xD165, 0x0000, 0x0000 },
//330 + 0 градуса
```

```
{0xCEDB, 0xCEDB, 0xCEDB, 0xCEDB, 0xCEDB, 0xCEDB, 0xCEDB, 0xCEDB, 0xCEDB, 0xCEDB, 0xDEFB, 0xBE5B,
0x02B7, 0xC187, 0xD165, 0xD165, 0xD165, 0xD165, 0xD165, 0xD165, 0xCF1C, 0xCEDB, 0xCEDB, 0xCEDB,
0xCEDB, 0xCEDB, 0xCEDB, 0xD145, 0xD165, 0xD165, 0xD165, 0xD165, 0xD165, 0xD165, 0x0000, 0x0000 },
//340 + 0 градуса
```

```
{0xCEDB, 0xCEDB, 0xCEDB, 0xCEDB, 0xCEDB, 0xCEDB, 0xCEDB, 0xCEBB, 0x8D9A, 0x1338, 0x1338, 0x1338, 0x1338,
0x1338, 0xE144, 0xD165, 0xD165, 0xD165, 0xD165, 0xD165, 0xD124, 0xCF1C, 0xCEDB, 0xCEDB, 0xCEDB,
0xCEDB, 0xCEDB, 0xCEDB, 0xCF1C, 0xD104, 0xD165, 0xD165, 0xD165, 0xD165, 0xD165, 0xD165, 0x0000, 0x0000 },
//350 + 0 градуса
```

```
{0xCEDB, 0xCEDB, 0xCEDB, 0xCEDB, 0xCEDB, 0xCEDB, 0xDF1B, 0x1338, 0x1338, 0x1338, 0x1338, 0x1338, 0x1338,
0x1338, 0xE943, 0xD165, 0xD165, 0xD165, 0xD165, 0xD165, 0xD124, 0xCF1C, 0xCEDB, 0xCEDB, 0xCEDB,
0xCEDB, 0xCEDB, 0xCEDB, 0xD165, 0xD165, 0xD165, 0xD165, 0xD165, 0xD165, 0xD165, 0x0000, 0x0000 },
//360 + 0 градуса
```

```
{0xCEDB, 0xCEDB, 0xCEDB, 0xCEDB, 0x855A, 0x1338, 0x1338, 0x1338, 0x1338, 0x1338, 0x1338, 0x1338,
0x1338, 0xD964, 0xD165, 0xD165, 0xD165, 0xD165, 0xD165, 0xD104, 0xCEFC, 0xCEDB, 0xCEDB, 0xCEDB,
0xCEDB, 0xCEDB, 0xCEDB, 0xCF5D, 0xD145, 0xD165, 0xD165, 0xD165, 0xD165, 0xD165, 0xD165, 0x0000, 0x0000 },
//370 + 0 градуса
```

```
};
```

```
const uint16_t frame2[][36]PROGMEM = {
```

```
{0x8E09, 0x8E09, 0x8E09, 0x8E09, 0x8E09, 0x8E09, 0x8E09, 0x8E09, 0x8E09, 0x8E09, 0x8E09, 0x8E09, 0x8E09,
0x8E09, 0x8E09, 0x8E29, 0x8E09, 0x8E09, 0x8E09, 0x8E09, 0x8E09, 0x8E09, 0x8E09, 0x8E09, 0x8E09, 0x8E09,
0x8E09, 0x8E09, 0x8E09, 0x8E09, 0x8E09, 0x8E09, 0x8E09, 0x8E09, 0x0000, 0x0000, 0x0000 }, //10 + 0 градуса
```

```
{0x8E09, 0x8E09, 0x8E09, 0x8E09, 0x8E09, 0x8E09, 0x8E09, 0x8E09, 0x8E09, 0x8E09, 0x8E09, 0x8E09, 0x8E09,
0x8E09, 0x8E09, 0x8E09, 0x8E09, 0x8E09, 0x8E09, 0x8E09, 0x8E09, 0x8E09, 0x8E09, 0x8E09, 0x8E09, 0x8E09,
0x8E09, 0x8E09, 0x8E09, 0x8E09, 0x8E09, 0x8E09, 0x8E09, 0x8E09, 0x8E09, 0x8E09, 0x0000, 0x0000 }, //20 + 0 градуса
```

					ЕЛІТ 6.171.00.02.267 ПЗ	Арк
						49
Вим.	Арк.	№ докум.	Підпис	Дата		

0xFE00, 0xFE00, 0xFE00, 0xC4A0, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000 }, //20 + 0
градуса

{0xFE00, 0xFE00, 0xFE00, 0xFE00, 0xD080, 0xD960, 0xD960, 0xD960, 0xD940, 0xFE40, 0xF460, 0xD960,
0xD960, 0xD940, 0xF9E0, 0xF9E0, 0xF9E0, 0xF9E0, 0xF9E0, 0xF9E0, 0xF9E0, 0xF9E0, 0xF9E0, 0xF9E0, 0xF9A0, 0xFE00,
0xFE00, 0xFE00, 0xB420, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000 }, //30 + 0
градуса

{0xFE00, 0xFE00, 0xFE00, 0xFE00, 0xD0A0, 0xD960, 0xD960, 0xD940, 0xFE00, 0xFE00, 0xD8C0, 0xD960,
0xD960, 0xF1C0, 0xF9E0, 0xF9E0, 0xF9E0, 0xF9E0, 0xF9E0, 0xF9E0, 0xF9E0, 0xF9E0, 0xF9E0, 0xF9E0, 0xF9E0, 0xF9E0, 0xFDE0,
0xFE00, 0xFE00, 0xFE00, 0xDD20, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000 }, //40 + 0
градуса

{0xFE00, 0xFE00, 0xFE00, 0xFE00, 0xD8A0, 0xD960, 0xD960, 0xFE00, 0xFE00, 0xFE00, 0xD8C0, 0xD960,
0xD960, 0xF1C0, 0xF9E0, 0xF9E0, 0xF9E0, 0xF9E0, 0xF9E0, 0xF9E0, 0xF9E0, 0xF9E0, 0xF9E0, 0xF9E0, 0xF9E0, 0xF9E0, 0xFDE0,
0xFE00, 0xFE00, 0xFE00, 0xD500, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000 }, //50 + 0
градуса

{0xFE00, 0xFE00, 0xFE00, 0xFE00, 0xFE81, 0xD900, 0xFE81, 0xFE00, 0xFE00, 0xFE00, 0xF460, 0xD960, 0xD960,
0xD940, 0xF9E0, 0xF9E0, 0xF9E0, 0xF9E0, 0xF9E0, 0xF9E0, 0xF9E0, 0xF9E0, 0xF9E0, 0xF9E0, 0xF9E0, 0xF9A0, 0xFE00, 0xFE00,
0xFE00, 0xFE00, 0xAC00, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000 }, //60 + 0 градуса

{0xFE00, 0xFE00, 0xFE00, 0xFE00, 0xFE20, 0xD000, 0xFE00, 0xFE00, 0xFE00, 0xFE00, 0xFE00, 0xD8C0, 0xD960,
0xD960, 0xD940, 0xF9E0, 0xF9E0, 0xF9E0, 0xF9E0, 0xF9E0, 0xF9E0, 0xF9E0, 0xF9E0, 0xF9E0, 0xF9E0, 0xF9C0, 0xFE20, 0xFE00,
0xFE00, 0xFE00, 0xC480, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000 }, //70 + 0 градуса

{0xFE00, 0xFE00, 0xFE00, 0xFE00, 0xFE00, 0xFE60, 0xFE00, 0xFE00, 0xFE00, 0xFE00, 0xFE00, 0xFE00, 0xFE00, 0xF460,
0xD960, 0xD960, 0xD960, 0xFA00, 0xF9E0, 0xF9E0, 0xF9E0, 0xF9E0, 0xF9E0, 0xF9E0, 0xF9A0, 0xFE20, 0xFE00,
0xFE00, 0xFE00, 0xFE00, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000 }, //80 + 0 градуса

{0xFE00, 0xFE00, 0xFE00, 0xFE00, 0xFE00, 0xFE00, 0xFE00, 0xFE00, 0xFE00, 0xFE00, 0xFE00, 0xFE00, 0xFE00, 0xF460,
0xD960, 0xD960, 0xD960, 0xD960, 0xD940, 0xF9E0, 0xF9E0, 0xF9E0, 0xF9E0, 0xF9E0, 0xF9E0, 0xF9E0, 0xF9E0, 0xF9E0, 0xF9E0,
0xFE00, 0xFE00, 0xED80, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000 }, //90 + 0 градуса

{0xFE00, 0xFE00, 0xFE00, 0xFE00, 0xFE00, 0xFE00, 0xFE00, 0xFE00, 0xFE00, 0xFE00, 0xFE00, 0xFE00, 0xFE00, 0xFE00,
0xFE00, 0xFE00, 0xFE00, 0xFE00, 0xFE20, 0xD8C0, 0xD960, 0xD960, 0xD960, 0xD940, 0xF1A0, 0xFE41, 0xFE00,
0xFE00, 0xFE00, 0xFE00, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000 }, //100 + 0 градуса

{0xFE00, 0xFE00, 0xFE00, 0xFE00, 0xFE00, 0xFE00, 0xFE00, 0xFE00, 0xFE00, 0xFE00, 0xFE00, 0xFE00, 0xFE00, 0xFE20,
0xFE00, 0xFE00, 0xFE00, 0xFE00, 0xFE00, 0xFE00, 0xFE00, 0xFE00, 0xFE00, 0xFE00, 0xFE00, 0xFE00, 0xFE40, 0xD4C1,
0xD4E1, 0xD4E1, 0xED61, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000 }, //110 + 0 градуса

{0xFE00, 0xFE00, 0xFE00, 0xFE00, 0xFE00, 0xFE00, 0xFE00, 0xFE00, 0xFE00, 0xFE00, 0xFE00, 0xFE00, 0xFE00, 0xFE00,
0xFE40, 0xD8E0, 0xD960, 0xD960, 0xD960, 0xD960, 0xD960, 0xD960, 0xD960, 0xD960, 0xD960, 0xD960, 0xD960, 0xD960, 0xD541,
0xD4E1, 0xD4E1, 0xD4E1, 0xD4E1, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000 }, //120 + 0
градуса

{0xFE00, 0xFE00, 0xFE00, 0xFE00, 0xFE00, 0xFE00, 0xFE00, 0xFE00, 0xFE00, 0xFE00, 0xFE00, 0xFE00, 0xFE00, 0xFE00,
0xFE00, 0xFE00, 0xFE00, 0xFE20, 0xD960, 0xD960, 0xD960, 0xD960, 0xE980, 0xF9E0, 0xF9E0, 0xFE40, 0xFE00,
0xFE00, 0xFE00, 0xFE00, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000 }, //130 + 0 градуса

{0xFE00, 0xFE00, 0xFE00, 0xFE00, 0xFE00, 0xFE00, 0xFE00, 0xFE00, 0xFE00, 0xFE00, 0xFE00, 0xFE00, 0xFEE1,
0xFDC0, 0xD960, 0xD080, 0xD960, 0xD960, 0xD960, 0xD960, 0xD960, 0xD960, 0xD960, 0xD960, 0xD960, 0xD960, 0xFE81,

					ЕЛІТ 6.171.00.02.267 ПЗ	Арк
						53
Вим.	Арк.	№ докум.	Підпис	Дата		

0xD000, 0xD000, 0xD000, 0xD000, 0xD000, 0xA820, 0x2104, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000 },
//240 + 0 градуса

{0xFFFF, 0xFFFF, 0xFFFF, 0xFFFF, 0xC618, 0x18E3, 0x6882, 0xD000, 0x2104, 0xE800, 0xD000, 0xD000, 0xD000,
0xD000, 0xD000, 0xE000, 0xD000, 0xD000, 0xD000, 0xD000, 0xD000, 0xD000, 0xD000, 0xD000, 0xD000,
0xD000, 0xD000, 0x1124, 0x1904, 0xE000, 0xD000, 0x2104, 0x0000, 0x0000, 0x0000, 0x0000 }, //250 + 0
градуса

{0xFFFF, 0xFFFF, 0xFFFF, 0xFFFF, 0xFFFF, 0xFFFF, 0x2104, 0x9061, 0x2104, 0xA041, 0xA041, 0xD000, 0xD000,
0xD000, 0xE800, 0x2104, 0xE800, 0xD000, 0xD000, 0xD000, 0xD000, 0xD000, 0xD000, 0xD000, 0xD000,
0xD000, 0xD000, 0x0924, 0x0000, 0x1082, 0x2104, 0x0841, 0x0000, 0x0000, 0x0000, 0x0000 }, //260 + 0
градуса

{0xFFFF, 0xFFFF, 0xFFFF, 0xFFFF, 0xFFFF, 0xFFFF, 0x18E3, 0x2104, 0x2104, 0x0000, 0x0000, 0x18C3, 0x2104,
0xA841, 0x2104, 0x9841, 0xD000, 0xD000, 0xD000, 0xD000, 0xD000, 0xD000, 0xD000, 0xD000, 0xD000,
0xD000, 0xD000, 0xD000, 0x1104, 0x10A2, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000 }, //270 + 0
градуса

{0xFFFF, 0xFFFF, 0xFFFF, 0xFFFF, 0xFFFF, 0xFFFF, 0x2104, 0x0020, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000,
0x2104, 0x8082, 0xD000, 0xD000, 0xD000, 0xD000, 0xD000, 0xD000, 0xD000, 0xF000, 0xE800, 0xD000,
0xD000, 0xD000, 0xE800, 0x2104, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000 }, //280 + 0
градуса

{0xFFFF, 0xFFFF, 0xFFFF, 0xFFFF, 0xFFFF, 0x7BCF, 0x2104, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0861,
0x0124, 0xD000, 0xD000, 0xD000, 0xD800, 0xD000, 0xD000, 0xD000, 0x0924, 0x0861, 0x0124, 0xD000,
0xE000, 0x1904, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000 }, //290 + 0
градуса

{0xFFFF, 0xFFFF, 0xFFFF, 0xFFFF, 0xFFFF, 0x1082, 0x0020, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x1124,
0xD000, 0xD000, 0xF000, 0x2104, 0xB020, 0xD000, 0xB020, 0x2104, 0x0000, 0x0000, 0x2104, 0x18E3, 0x0000,
0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000 }, //300 + 0 градуса

{0xFFFF, 0xFFFF, 0xFFFF, 0xFFFF, 0xFFFF, 0x2104, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x2104, 0xF000,
0xD000, 0x0924, 0x1082, 0x0124, 0xE800, 0x0924, 0x0020, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000,
0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000 }, //310 + 0 градуса

{0xFFFF, 0xFFFF, 0xFFFF, 0xFFFF, 0xFFFF, 0x2104, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0124, 0xA820,
0x1904, 0x0841, 0x2104, 0x2104, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000,
0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000 }, //320 + 0 градуса

{0xFFFF, 0xFFFF, 0xFFFF, 0xFFFF, 0xFFFF, 0x2104, 0x0000, 0x0000, 0x0000, 0x0000, 0x2104, 0x2104, 0x0841,
0x0000, 0x2104, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000,
0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000 }, //330 + 0 градуса

{0xFFFF, 0xFFFF, 0xFFFF, 0xFFFF, 0xFFFF, 0x2104, 0x0000, 0x0000, 0x0000, 0x0000, 0x2104, 0x0000, 0x0000,
0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000,
0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000 }, //340 + 0 градуса

{0xFFFF, 0xFFFF, 0xFFFF, 0xFFFF, 0xFFFF, 0x18E3, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000,
0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000,
0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000 }, //350 + 0 градуса

					ЕЛІТ 6.171.00.02.267 ПЗ	Арк
						58
Вим.	Арк.	№ докум.	Підпис	Дата		


```
{0xFEAO, 0xFEAO, 0xFEAO, 0xFEAO, 0xFEAO, 0xFEAO, 0x4128, 0xFFFF, 0xFFFF, 0xEF3C, 0x5A06, 0x5A06, 0x5A06,
0xD581, 0xFEAO, 0xFEAO, 0xFEAO, 0xFEAO, 0xFEAO, 0xFEAO, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000,
0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000 }, //370 + 0 градуса
};
```

```
const uint16_t frame6[][36]PROGMEM={
```

```
{0xFFFF, 0xFFFF, 0x5E9A, 0x05D8, 0x05F8, 0x05F8, 0x05F8, 0x05D7, 0x0597, 0x0596, 0xC77D, 0xFFFF, 0xFFFF,
0x0576, 0x0597, 0x05B7, 0x0597, 0x0596, 0x0576, 0xFFFF, 0xFFFF, 0xFFFF, 0xFFFF, 0x667A, 0x0535, 0x0556,
0x0556, 0x0555, 0x0555, 0x0515, 0x0514, 0x0472, 0x0492, 0x0000, 0x0000, 0x0000 }, //10 + 0 градуса
```

```
{0xFFFF, 0xFFFF, 0xBF9E, 0x0596, 0x05D8, 0x05F8, 0x05D7, 0x05F8, 0x05F8, 0x05D7, 0x05B7, 0x0597, 0x0597,
0x05B7, 0x05B7, 0x0597, 0x0535, 0xFFFF, 0xFFFF, 0xFFFF, 0xFFFF, 0xFFFF, 0x04F4, 0x0535, 0x0556, 0x0556,
0x0556, 0x0555, 0x0535, 0x0514, 0x04F4, 0x0472, 0x0492, 0x0000, 0x0000, 0x0000 }, //20 + 0 градуса
```

```
{0xFFFF, 0xFFFF, 0xFFFF, 0xFFFF, 0x0597, 0x05B7, 0x05D7, 0x05D7, 0x05D7, 0x05B7, 0x05B7, 0x0556,
0x973D, 0xFFFF, 0xFFFF, 0xFFFF, 0xFFFF, 0xFFFF, 0xFFFF, 0x04F4, 0x0555, 0x0576, 0x0576, 0x0576, 0x0556,
0x0555, 0x0555, 0x0535, 0x0514, 0x04B3, 0x0472, 0x030C, 0x0000, 0x0000, 0x0000 }, //30 + 0 градуса
```

```
{0xFFFF, 0xFFFF, 0xFFFF, 0xFFFF, 0xFFFF, 0xFFFF, 0xFFFF, 0xFFFF, 0xFFFF, 0xFFFF, 0xFFFF, 0xFFFF,
0xFFFF, 0xFFFF, 0xFFFF, 0x0514, 0x0555, 0x0576, 0x0596, 0x0576, 0x0576, 0x0576, 0x0576, 0x0556,
0x0556, 0x0555, 0x0535, 0x0514, 0x04D3, 0x0472, 0x0451, 0x0000, 0x0000, 0x0000 }, //40 + 0 градуса
```

```
{0xFFFF, 0xFFFF, 0xFFFF, 0xFFFF, 0xFFFF, 0xFFFF, 0xFFFF, 0xFFFF, 0xFFFF, 0xFFFF, 0xFFFF, 0xFFFF,
0xF7DF, 0x0535, 0x0576, 0x0576, 0x0596, 0x0597, 0x0596, 0x0596, 0x0576, 0x0576, 0x0576, 0x0576, 0x0556,
0x0555, 0x0555, 0x0535, 0x0514, 0x04D3, 0x0472, 0x03AF, 0x0000, 0x0000, 0x0000 }, //50 + 0 градуса
```

```
{0xFFFF, 0xFFFF, 0xFFFF, 0xFFFF, 0xFFFF, 0xFFFF, 0xFFFF, 0xFFFF, 0xFFFF, 0xFFFF, 0x4639, 0x0576, 0x0596,
0x0597, 0x05B7, 0x05B7, 0x0597, 0x0597, 0x0596, 0x0596, 0x0596, 0x0596, 0x0576, 0x0576, 0x0556, 0x0556,
0x0555, 0x0555, 0x0535, 0x0514, 0x0472, 0x0472, 0x0082, 0x0000, 0x0000, 0x0000 }, //60 + 0 градуса
```

```
{0xFFFF, 0xFFFF, 0xFFFF, 0xFFFF, 0xFFFF, 0xFFFF, 0xFFFF, 0x0556, 0x0597, 0x0597, 0x05B7, 0x05B7, 0x05D7,
0x05B7, 0x05B7, 0x05B7, 0x0597, 0x0597, 0x0597, 0x0596, 0x0596, 0x0576, 0x0576, 0x0576, 0x0576, 0x0576,
0x0556, 0x0535, 0x0535, 0x0514, 0x04B3, 0x0472, 0x0166, 0x0000, 0x0000, 0x0000 }, //70 + 0 градуса
```

```
{0xFFFF, 0xFFFF, 0xFFFF, 0xFFFF, 0x973D, 0x0556, 0x0597, 0x05B7, 0x05D8, 0x05D7, 0x05D8, 0x05D7, 0x05D7,
0x05B7, 0x05B7, 0x05B7, 0x0597, 0x0596, 0x0596, 0x0596, 0x0576, 0x0576, 0x0576, 0x0576, 0x0576, 0x0556,
0x0555, 0x0555, 0x0535, 0x0514, 0x0493, 0x0472, 0x01C7, 0x0000, 0x0000, 0x0000 }, //80 + 0 градуса
```

```
{0xFFFF, 0xFFFF, 0xFFFF, 0x0596, 0x0576, 0x0597, 0x05D7, 0x05F8, 0x05F8, 0x05D8, 0x05D7, 0x05D7, 0x05D7,
0x05B7, 0x05B7, 0x05B7, 0x0597, 0x0597, 0x0597, 0x0596, 0x0596, 0x0576, 0x0576, 0x0576, 0x0576, 0x0576,
0x0555, 0x0555, 0x0535, 0x0514, 0x0472, 0x0472, 0x0493, 0x0000, 0x0000, 0x0000 }, //90 + 0 градуса
```

```
{0xFFFF, 0xFFFF, 0xFFFF, 0xFFFF, 0x0596, 0x0576, 0x05B7, 0x05D7, 0x05D8, 0x05D8, 0x05D7, 0x05D7, 0x05D7,
0x05B7, 0x05B7, 0x05B7, 0x0597, 0x0597, 0x0597, 0x0596, 0x0596, 0x0596, 0x0576, 0x0576, 0x0556, 0x0555,
0x0555, 0x0555, 0x0535, 0x0514, 0x04B3, 0x0472, 0x02EC, 0x0000, 0x0000, 0x0000 }, //100 + 0 градуса
```

```
{0xFFFF, 0xFFFF, 0xFFFF, 0xFFFF, 0xFFFF, 0xFFFF, 0x76BA, 0x0556, 0x0597, 0x05B7, 0x05D7, 0x05B7, 0x05B7,
0x05B7, 0x05B7, 0x05B7, 0x0597, 0x0597, 0x0597, 0x0596, 0x0576, 0x0596, 0x0576, 0x0556, 0x0556, 0x0556,
0x0555, 0x0555, 0x0535, 0x0514, 0x04D3, 0x0472, 0x03AF, 0x0000, 0x0000, 0x0000 }, //110 + 0 градуса
```

					ЕЛІТ 6.171.00.02.267 ПЗ	Арк
						62
Вим.	Арк.	№ докум.	Підпис	Дата		

{0xFFFF, 0xFFFF, 0xFFFF, 0xFFFF, 0xFFFF, 0xFFFF, 0x05B7, 0x05F8, 0x05F8, 0x05F8, 0x05D8, 0x05D7, 0x05D7, 0x05B7, 0x05D7, 0x05B7, 0x05B7, 0x0597, 0x0597, 0x0596, 0x0596, 0x0596, 0x0596, 0x0576, 0x0576, 0x0576, 0x0576, 0x0556, 0x0555, 0x0555, 0x0555, 0x0535, 0x0514, 0x0472, 0x0514, 0x0000, 0x0000 }, //260 + 0 градуса

{0xFFFF, 0xFFFF, 0xFFFF, 0xFFFF, 0xDFBE, 0x05B7, 0x05F8, 0x05F8, 0x05F8, 0x05F8, 0x05D8, 0x05D8, 0x05D7, 0x05D7, 0x05B7, 0x05B7, 0x05B7, 0x05B7, 0x0596, 0x0596, 0x0596, 0x0596, 0x0576, 0x0596, 0x0576, 0x0576, 0x0576, 0x0555, 0x0555, 0x0555, 0x0515, 0x0514, 0x0472, 0x2575, 0x0000, 0x0000 }, //270 + 0 градуса

{0xFFFF, 0xFFFF, 0xFFFF, 0xFFFF, 0xFFFF, 0xFFFF, 0x6EDB, 0x05D7, 0x05F8, 0x05D8, 0x05F8, 0x05D8, 0x05D8, 0x05D7, 0x05D7, 0x05B7, 0x05B7, 0x05B7, 0x0597, 0x0596, 0x0596, 0x0596, 0x0596, 0x0576, 0x0576, 0x0576, 0x0576, 0x0556, 0x0555, 0x0556, 0x0555, 0x0514, 0x0514, 0x0472, 0x2555, 0x0000, 0x0000 }, //280 + 0 градуса

{0xFFFF, 0xFFFF, 0xFFFF, 0xFFFF, 0xFFFF, 0xFFFF, 0xFFFF, 0xFFFF, 0xAF7D, 0x05B7, 0x05D7, 0x05D7, 0x05D7, 0x05D7, 0x05B7, 0x05B7, 0x05B7, 0x0596, 0x0596, 0x0596, 0x0596, 0x0576, 0x0576, 0x0576, 0x0576, 0x0555, 0x0576, 0x0555, 0x0555, 0x0515, 0x04F4, 0x0472, 0x1D76, 0x0000, 0x0000 }, //290 + 0 градуса

{0xFFFF, 0xFFFF, 0xFFFF, 0xFFFF, 0xFFFF, 0xFFFF, 0xFFFF, 0xFFFF, 0xFFFF, 0xFFFF, 0xFFFF, 0xFFFF, 0x467A, 0x05B7, 0x05B7, 0x05B7, 0x05B7, 0x0596, 0x0597, 0x0596, 0x0596, 0x0596, 0x0596, 0x0576, 0x0576, 0x0576, 0x0576, 0x0576, 0x0555, 0x0555, 0x0535, 0x0515, 0x04D3, 0x0472, 0x2535, 0x0000, 0x0000 }, //300 + 0 градуса

{0xFFFF, 0xFFFF, 0xFFFF, 0xFFFF, 0xFFFF, 0xFFFF, 0xFFFF, 0xFFFF, 0xFFFF, 0xFFFF, 0xFFFF, 0xFFFF, 0xFFFF, 0xFFFF, 0xFFFF, 0x0597, 0x0597, 0x05B7, 0x0596, 0x0596, 0x0596, 0x0596, 0x0576, 0x0576, 0x0576, 0x0556, 0x0556, 0x0556, 0x0555, 0x0535, 0x0514, 0x04F4, 0x0472, 0x4D96, 0x0000, 0x0000 }, //310 + 0 градуса

{0xFFFF, 0xFFFF, 0xFFFF, 0xFFFF, 0x5E9A, 0x0535, 0x0556, 0x0596, 0x0515, 0x0535, 0x0576, 0x7E9A, 0xFFFF, 0xFFFF, 0xFFFF, 0xFFFF, 0xFFFF, 0xFFFF, 0x0576, 0x0596, 0x0596, 0x0576, 0x0576, 0x0576, 0x0556, 0x0555, 0x0555, 0x0555, 0x0535, 0x0534, 0x04B3, 0x0472, 0x2BEF, 0x0000, 0x0000 }, //320 + 0 градуса

{0xFFFF, 0xFFFF, 0x76BB, 0x0596, 0x05B7, 0x05D7, 0x05D8, 0x05D8, 0x05D8, 0x05D7, 0x05D7, 0x05B7, 0x0597, 0x0596, 0x0576, 0x0514, 0xFFFF, 0xFFFF, 0xFFFF, 0xFFFF, 0xFFFF, 0x25B7, 0x0576, 0x0576, 0x0576, 0x0576, 0x0555, 0x0556, 0x0555, 0x0535, 0x0514, 0x0472, 0x0472, 0x03F0, 0x0000, 0x0000 }, //330 + 0 градуса

{0xFFFF, 0xFFFF, 0x05B7, 0x05D7, 0x05F8, 0x05F8, 0x05F8, 0x05F8, 0x05D8, 0x05D7, 0x05B7, 0xEFDF, 0x4E7A, 0x0576, 0x05B7, 0x05B7, 0x0576, 0x0555, 0xBF7D, 0xFFFF, 0xFFFF, 0xFFFF, 0xFFFF, 0x0535, 0x0576, 0x0556, 0x0555, 0x0555, 0x0555, 0x0535, 0x0514, 0x0472, 0x0472, 0x00A2, 0x0000, 0x0000 }, //340 + 0 градуса

{0xFFFF, 0xFFFF, 0x05B7, 0x05D8, 0x05F8, 0x05F8, 0x05F8, 0x0596, 0xFFFF, 0xAF3D, 0xEFDF, 0xFFFF, 0xFFFF, 0x0576, 0x05B7, 0x05B7, 0x0597, 0x0596, 0x0576, 0x6E79, 0xFFFF, 0xFFFF, 0xFFFF, 0xFFFF, 0x0535, 0x0556, 0x0576, 0x0555, 0x0555, 0x0535, 0x04F4, 0x0472, 0x0472, 0x0000, 0x0000, 0x0000 }, //350 + 0 градуса

{0xFFFF, 0xFFFF, 0x0596, 0x05F8, 0x05F8, 0x0618, 0x05D8, 0x0576, 0xFFFF, 0xFFFF, 0xFFFF, 0xFFFF, 0xFFFF, 0xFFFF, 0x0576, 0x0596, 0x05B7, 0x0596, 0x04F4, 0xFFFF, 0xFFFF, 0xFFFF, 0xFFFF, 0x0535, 0x0555, 0x0555, 0x0556, 0x0535, 0x0535, 0x04F4, 0x0472, 0x0472, 0x0000, 0x0000, 0x0000 }, //360 + 0 градуса

{0xFFFF, 0xFFFF, 0x5E9A, 0x05D8, 0x05F8, 0x05F8, 0x05F8, 0x05D7, 0x0597, 0x0596, 0xC77D, 0xFFFF, 0xFFFF, 0x0576, 0x0597, 0x05B7, 0x0597, 0x0596, 0x0576, 0xFFFF, 0xFFFF, 0xFFFF, 0xFFFF, 0x667A, 0x0535, 0x0556, 0x0556, 0x0555, 0x0555, 0x0515, 0x0514, 0x0472, 0x0492, 0x0000, 0x0000, 0x0000 }, //370 + 0 градуса

};

const uint16_t frame7[][36]PROGMEM={

					ЕЛІТ 6.171.00.02.267 ПЗ	Арк
						64
Вим.	Арк.	№ докум.	Підпис	Дата		

