

МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ
СУМСЬКИЙ ДЕРЖАВНИЙ УНІВЕРСИТЕТ
Кафедра електроніки і комп'ютерної техніки

ПОЯСНЮВАЛЬНА ЗАПИСКА

ДО ВИПУСКНОЇ КВАЛІФІКАЦІЙНОЇ РОБОТИ МАГІСТРА НА ТЕМУ:

**«СИСТЕМА КЕРУВАННЯ
ЕЛЕКТРИЧНИМ СТРУМОМ ТА ПОТУЖНІСТЮ»**

Виконав студент

гр. КСмз-71С

Гайдар Я.В.

Завідуючий кафедрою

Опанасюк А.С.

Консультант

з техніко-економічної частини

Маценко О.М.

Керівник

кваліфікаційної роботи

Горячев О.Є.

Суми 2018

КАЛЕНДАРНИЙ ПЛАН

№	Назва етапів проектування	Термін виконання	Примітка
1	Огляд літератури та формулювання задачі розробки		
2	Вибір алгоритму функціонування		
3	Вибір та обґрунтування схеми структурної		
4	Розробка схеми функціональної		
5	Вибір елементної бази та розробка схеми принципової		
6	Науково-дослідна частина		
7	Економічна частина		
8	Оформлення пояснювальної записки та креслень		
9	Представлення матеріалів керівнику і одержання відгуку на проект		
10	Здача проекту на кафедру для рецензування та захист		

Студент _____
(підпис)

Керівник _____
(підпис)

“ _____ ” _____ 20 ____ р.

РЕФЕРАТ

Тема дипломної роботи: «Система керування електричним струмом та потужністю». Робота містить 95 сторінок. Кваліфікаційна робота складається з 8 розділів, розділи у свою чергу розбиті на окремі підрозділи. Робота включає у себе 35 рисунків, 6 таблиць, схему алгоритму, структурну схему пристрою, електричну функціональну схему пристрою, електричну принципову схему пристрою та Додатки А-Д. При написанні роботи було використано інформацію з 19 джерел.

У розділі «Огляд літератури і постановка задачі проектування» дипломної роботи був зроблений вибір напрямку й пристрою для проектування, описана актуальність проектування а також визначені основні характеристики майбутнього приладу. У розділі «Вибір та обґрунтування структурної схеми» наведено огляд необхідних блоків, що входять до складу проектованого пристрою, розроблені структурні схеми пристрою. У розділі «Розробка функціональної схеми пристрою» була розроблена функціональна схема, на основі якої, у розділі 4 була побудована схема електрична принципова, здійснено розрахунок та підбір необхідних елементів. У розділі 5 та у додатках А-Г приводиться огляд програмного забезпечення пристрою. Розділи 6 та 7 містять опис послідовності виготовлення пристрою, налаштувань а також деякі результати випробувань. Розділ 8 включає розрахунок та економічне обґрунтування щодо можливості впровадження пристрою в серійне виробництво.

Зроблені висновки щодо результату проведеної роботи.

Ключові слова: МІКРОКОНТРОЛЕР, ОПЕРАЦІЙНИЙ ПІДСИЛЮВАЧ, ГРАФІЧНИЙ ДИСПЛЕЙ, ФАЗОВЕ КЕРУВАННЯ ПОТУЖНІСТЮ, КОНТАКТНЕ ЗВАРЮВАННЯ МЕТАЛУ, ЗАРЯД АКБ, ШІМ.

ЗМІСТ

с.

Перелік скорочень.....	5
Вступ.....	6
Розділ 1 - Огляд літератури і постановка задачі проектування	7
1.1 Огляд літератури за темою проекту.....	7
1.2 Технічні умови та функціональні можливості пристрою.....	11
Розділ 2 - Вибір та обґрунтування структурної схеми.....	12
Розділ 3 – Розробка функціональної схеми.....	17
Розділ 4 - Розроблення та розрахунки принципів електричних схем	
вузлів і блоків пристрою.....	19
4.1 Блок 1 фазового керування потужністю	19
4.2 Схема стабілізації та керування струмом	20
4.3 Блок 2 фазового керування потужністю	22
4.4 Схема вимірювання напруги	23
4.5 Схема вимірювання вихідного струму	25
4.6 Температурні сенсори DS18B20.....	26
4.7 Мікроконтролер	30
4.8 Графічний дисплей	34
4.9 Блок підготовки живлення.....	37
4.10 Клавіатура.....	38
4.11 Розробка загальної електричної схеми пристрою.....	42
Розділ 5 – Розробка алгоритму та програмне забезпечення пристрою.....	44
Розділ 6 – Реалізація спроектованого пристрою.....	46
Розділ 7 – Науково – дослідна частина.....	50

ФЗ 8.17100.04.370 ПЗ										
Зм.	Арк.	№ докум.	Підпис	Дата	Система керування електричним струмом та потужністю Пояснювальна записка					
Розроб.		Гайдар Я.В.						Літ.	Арк.	Аркушів
Перевір.		Горячев О.Є.						3	95	
Реценз.								СумДУ, КСмз-71С		
Н. Контр.		Гапич В.М.								
Затв.		Опанасюк О.С.								

Розділ 8 – Економічна частина	59
8.1 Розрахунок повної собівартості системи.....	59
8.2 Визначення ціни системи.....	66
8.3 Визначення річних витрат користувача	67
8.4 Розрахунок повної ціни користувача.....	69
8.5 Оцінка конкурентоспроможності	70
Висновок.....	71
Список літератури.....	72
Додатки	74

ПЕРЕЛІК СКОРОЧЕНЬ

В пояснювальній записці зустрічаються деякі скорочення, що відповідають наступним визначенням:

АКБ – акумуляторна батарея

АЦП – аналогово-цифровий перетворювач

ДН – дільник напруги

ІМС – інтегральна мікросхема

КЗ – коротке замкнення

ККД – коефіцієнт корисної дії

МК – мікроконтролер

ОП – операційний підсилювач

СТ – силовий трансформатор

ТС – температурний сенсор

ШІМ – широтно-імпульсна модуляція

					ФЗ 8.17100.04.370 ПЗ	Арк
						5
ЗМ.	Лист	№ докум.	Підпис	Дата		

ВСТУП

У зв'язку із зростаючою необхідністю здобути у персональне користування потужний апарат для контактного зварювання металів, а також пускозарядного пристрою для заряджання та тестування характеристик 12В автомобільних акумуляторних батарей, було проведено вивчення сучасного ринку таких пристроїв промислового походження. В результаті, зважаючи на пропоновану магазинами вартість та функціонал таких виробів, було прийнято рішення про самостійне проектування та виготовлення пристрою, що поєднує бажаний функціонал та має необхідні характеристики.

Результатом проведеної роботи і є даний дипломний проект.

					ФЗ 8.17100.04.370 ПЗ	Арк
ЗМ.	Лист	№ докум.	Підпис	Дата		6

1 ОГЛЯД ЛІТЕРАТУРИ І ПОСТАНОВКА ЗАДАЧІ ПРОЕКТУВАННЯ

1.1 Огляд літератури за темою проекту

Так як проєктований пристрій поєднує в собі функції контактного зварювання металів та заряду кислотних автомобільних акумуляторних батарей, розглянемо основні принципи щодо його реалізації.

Контактне зварювання металу

Спочатку, деталі що з'єднуються, орієнтуються в потрібному положенні та розміщуються між електродами сварочного апарата і надійно притискаються між собою. Після цього вони розігріваються до стану пластичності і спільно піддаються пластичному деформуванню у точці з'єднання. В результаті локального розігріву формується точкове з'єднання.

Деталі нагріваються за рахунок проходження зварювального струму у вигляді короткочасного імпульсу тривалістю 0,01 ... 0,1 секунд в залежності від умов зварювання. Цим імпульсом забезпечується розплавлення металу в зоні дії електродів і утворення спільного рідкого ядра обох деталей, діаметр якого може складати від 4 до 12 мм. Після припинення дії імпульсу струму деталі протягом деякого часу утримуються під тиском, щоб розплавлене ядро охоллоло і кристалізувалося.

Значення тиску між електродами має забезпечувати надійний контакт деталей в місці з'єднання. Воно залежить від металу, що зварюється і товщини деталей. Тиск після нагрівання має важливе значення, так як його величина забезпечує дрібнозернисту структуру металу в місці зварювання, а міцність точки з'єднання може досягати міцності базового металу.

При зварюванні деталей з сталей, схильних до загартування і можливого утворення тріщин (наприклад, вуглецеві сталі), рекомендується збільшувати час нагрівання для уповільнення подальшого охолодження металу. Зварювання деталей з нержавіючих аустенітних сталей треба виконувати навпаки, з якомога

					ФЗ 8.17100.04.370 ПЗ	Арк
ЗМ.	Лист	№ докум.	Підпис	Дата		7

меншою тривалістю нагріву. Це робиться для запобігання небезпеки нагрівання зовнішньої поверхні точки з'єднання до температури структурних перетворень, що може спричинити за собою порушення високих антикорозійних властивостей зовнішніх шарів металу.

Якість зварювання залежить також і від правильного вибору діаметра мідного електрода. Діаметр точки з'єднання повинен перевищувати товщину самого тонкого елемента зварного з'єднання бути в 2 - 3 рази. Притиском деталей в момент проходження зварювального імпульсу забезпечується утворення поблизу розплавленого ядра особливого ущільнюючого паска, який перешкоджає виплеску розплавленого матеріалу із зони зварювання.

Заряд кислотних АКБ

Для того щоб акумуляторна батарея нормально відпрацювала встановлений термін (5-11 років), необхідно підтримувати акумуляторну батарею в зарядженому стані.

Існує кілька варіантів зарядки акумулятора:

- заряд стабілізованим струмом,
- заряд стабілізованою напругою,
- зарядка імпульсним струмом (пульсуючим і асиметричним струмом),
- контрольно-тренувальний цикл,
- зарядка по Вудбріджу (правило ампер-годин).

Заряд стабілізованим струмом

Зарядний пристрій вимірює напругу на клеммах акумулятора і, якщо це значення нижче нижнього граничного рівня, включається ключ і відбувається заряд встановленим струмом. При досягненні верхнього порогу пристрій управління відключає ключ і заряд припиняється. У разі зниження напруги, процес повторюється.

Переваги: відсутня залежність величини струму заряду від коливань напруги мережі і ступеня зарядженості акумулятора; як правило, більш високий ККД; автоматизація процесу заряду.

					ФЗ 8.17100.04.370 ПЗ	Арк
ЗМ.	Лист	№ докум.	Підпис	Дата		8

Недоліки: не завжди можливо зарядити акумулятор до 100% ємності, особливо при великому зарядному струмі, також не виключена можливість перезарядження. При заряді великим струмом напруга на клеммах акумулятора зростає відносно швидко і до відключення акумулятор не встигає набрати повний заряд. При малому струмі напруга на клеммах зростає повільніше, акумулятор при цьому може набрати 100% ємності. Але цього струму може не вистачити для досягнення верхньої межі відключення. Акумулятор починає кипіти і, якщо не відключити зарядний пристрій то можливий його перезаряд.

Заряд стабілізованою напругою.

Цей спосіб заряду застосовується, як правило, на автомобілях, коли потрібно швидко відновити заряд акумулятора. Стабілізатором напруги в цьому випадку є генератор постійного струму, напруга якого підтримується автоматично за допомогою реле-регулятора. Напруга бортової мережі при цьому повинна бути 2,4 В при перерахунку на одну банку (або 14,4 В на 12-вольтну батарею).

На початку заряду ток має найбільше значення внаслідок значної різниці між напругою джерела та напругою акумулятора. При цьому, чим більше потужність зарядного джерела струму і чим сильніше розряджений акумулятор, тим більше початковий зарядний струм. Впродовж заряду напруга батареї збільшується і величина зарядного струму спадає до мінімального значення.

Переваги: струм автоматично зменшується в процесі зарядки батареї.

Недоліки:

- потрібна точна установка напруги джерела зарядного струму, щоб уникнути систематичного недозаряда або перезаряда,
- значний початковий зарядний струм.

Комбінований заряд

Заряд по методу IU. Заряд АКБ відбувається в два етапи.

Перший етап - заряд стабілізованим струмом (I). Другий етап - заряд стабілізованою напругою (U). Порог стабілізації по напрузі складає 13,8 В на

					ФЗ 8.17100.04.370 ПЗ	Арк
ЗМ.	Лист	№ докум.	Підпис	Дата		9

АКБ. Перший етап заряду дозволяє порівняно швидко набрати основну ємність акумулятора, не доводячи електроліт до кипіння. Якщо заряджати акумулятор, застосовуючи лише режим стабілізації по струму, то для повного заряду потрібно збільшити напругу на елементі АКБ більш ніж 2,3В. При цьому підвищується інтенсивність електрохімічних процесів і, як результат, знижується термін її служби.

Для виключення перерахованих недоліків застосовується другий етап - заряд стабілізованою напругою. Зарядний пристрій переходить в цей режим після досягнення напруги 2,3В. Струм починає поступово падати, і через деякий час зменшується до величини, рівної струму саморозряда акумулятора. В залежності від стану, ємності АКБ та температури навколишнього середовища ця величина коливається від десятків до сотень міліамперів. Такий алгоритм заряду зводить до мінімуму процес сульфатації, виключає перезарядження і дозволяє заряджати акумулятор до 100% ємності. При цьому можна тривалий час не відключати акумулятор від зарядного пристрою, підтримуючи його в постійній готовності до роботи. Як недоліки, слід зазначити тривалий час заряду і більш високу ціну зарядного пристрою.

Контрольно-тренувальний цикл

Контрольно-тренувальний цикл заряду-розряду проводиться для запобігання сульфатації пластин і для визначення ємності акумулятора. Контрольно-тренувальні цикли виконуються наступним чином:

- заряджають АКБ нормальним струмом (будь-яким з описаних способів) до повного заряду;
- витримують АКБ 3 години після припинення заряду;
- проводять контрольну розрядку постійним нормальним струмом 10-годинного режиму і контролюють час повного розряду до напруги 1,7 В (10,2 В на АКБ); ємність батареї визначають за величинами розрядного струму і часу розряду.

					ФЗ 8.17100.04.370 ПЗ	Арк
						10
ЗМ.	Лист	№ докум.	Підпис	Дата		

Після контрольного розряду батарею відразу ж ставлять на зарядку і повністю заряджають. Якщо виявилось, що ємність АКБ складає менше 50% від номінальної, вона вважається несправною.

1.2 Технічні умови та функціональні можливості пристрою

Пристрій, що проектується, має виконувати наступні функції:

1. Забезпечення можливості контактного зварювання металів. Орієнтовна потужність блоку складає близько 6,5кВт.
 - Автоматичне (по збереженим у пам'яті МК налаштуванням) та ручне керування потужністю та тривалістю вихідного імпульсу;
 - Можливість формування як одиночного, так і серії імпульсів із заданими параметрами;
 - Контроль температури потужних елементів блоку;
 - Відображення необхідної інформації на графічний дисплей;
2. Лабораторний блок живлення. Орієнтовні параметри блоку: $U_{\text{вих}} = 0 \dots 16\text{В}$, $I_{\text{вих}} = 0 \dots 15\text{А}$.
 - Можливість регулювання як вихідної напруги так і струму;
 - Стабілізація по кожному із заданих параметрів;
 - Захист від перевищення допустимого вихідного струму;
 - Контроль температури потужних елементів блоку;
 - Відображення необхідної інформації на графічний дисплей;
3. Автоматичний зарядний пристрій для кислотних АКБ. Передбачається використання лише 12В АКБ.
 - Можливість вимірювання ємності АКБ;
 - Формування різних режимів заряда, тестування та відновлення АКБ із можливістю збереження профілів до енергонезалежної пам'яті;
 - Контроль температури АКБ зовнішнім сенсором;
 - Відображення необхідної інформації на графічний дисплей;
 - Звукове інформування.

					ФЗ 8.17100.04.370 ПЗ	Арк
						11
ЗМ.	Лист	№ докум.	Підпис	Дата		

2 ВИБІР ТА ОБҐРУНТУВАННЯ СТРУКТУРНОЇ СХЕМИ

Так як пристрій розрахований на 2 основні режими: контактне зварювання металу та зарядка кислотних АКБ, в першому наближенні заданий функціонал можна реалізувати за наступною функціональною блок-схемою:

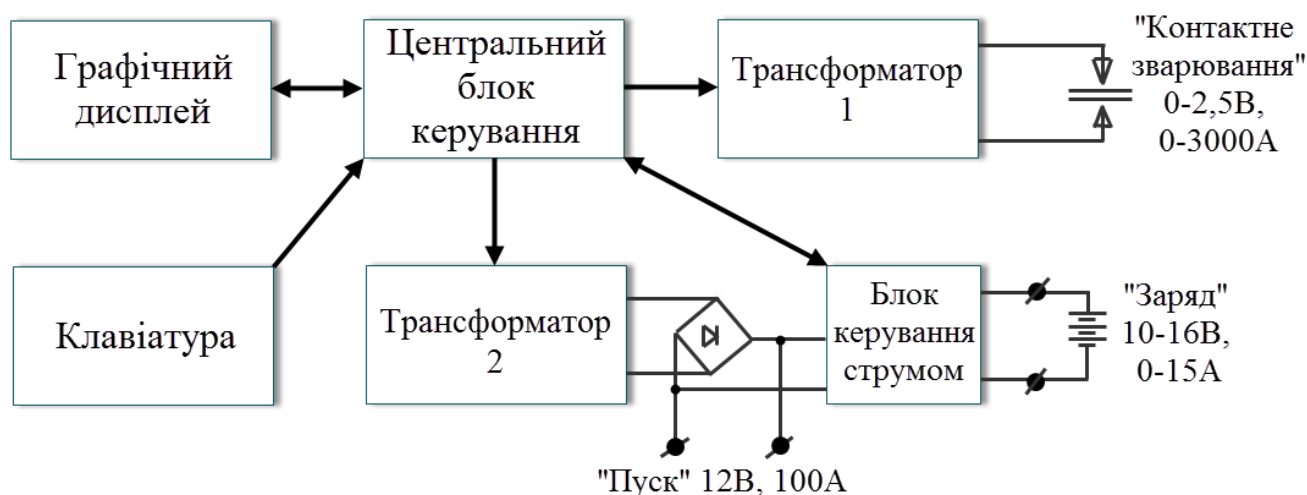


Рисунок 2.1 – спрощена структурна схема пристрою

Пристрій Рис 2.1 включає в себе блок кнопок з енкодером для керування роботою, графічний дисплей, центральний блок керування та 2 потужних трансформатори. Один з них призначений для реалізації контактного зварювання металів, тому використаємо трансформатор від промислового апарата електродугової сварки, заявленою потужністю 8кВА. Другий трансформатор використаємо від мікрохвильової печі Samsung, заявленою потужністю 1,2кВА, що забезпечить можливість допомоги при пуску двигуна автомобіля та для реалізації функції зарядки кислотних автомобільних АКБ.

Розглянемо більш детальну блок-схему пристрою Рис 2.2. Вона складається з наступних складових:

- Мікроконтролер;
- Блок підготовки живлення;
- Графічний дисплей;
- Клавіатура;
- Температурні сенсори;

- Блок вимірювання струму;
- Блок вимірювання напруг;
- Блок керування струмом;
- Блок 1 фазового керування потужністю;
- Блок 2 фазового керування потужністю;
- Трансформатор блоку контактного зварювання металів
- Трансформатор блоку зарядки АКБ та лабораторного блоку живлення

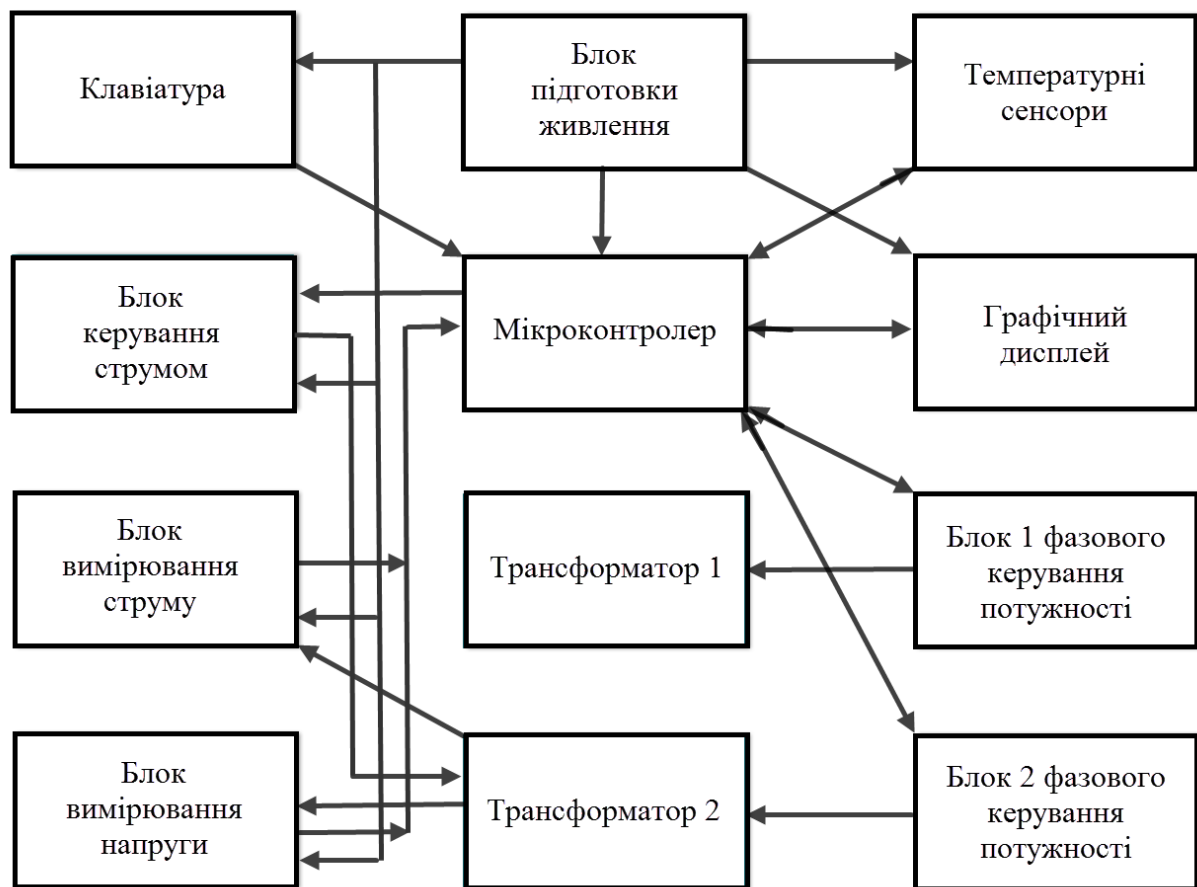


Рисунок 2.2 – структурна схема пристрою

Центральним та основним блоком пристрою є *мікроконтролер*, що взаємодіє з іншими складовими елементами та керує їх роботою. Враховуючи максимальну простоту проектування, виготовлення та налаштування роботи пристрою, вартість МК та наявність достатньої кількості портів, таймерів і іншої периферії, попередньо визначимо, що буде застосовано 8-бітний [11] мікроконтролер Atmega32 фірми ATMEL.

Блок підготовки живлення виконує функцію стабілізованого забезпечення інших блоків необхідними рівнями напруг. Передбачається його реалізація на розповсюджених стабілізаторах типу LM78xx.

Графічний дисплей буде використано від мобільного телефону Nokia1280, так як він має достатню область для відображення необхідної інформації, є достатньо розповсюдженим і відома система команд його внутрішнього контролера PCF8814 [2]. Дисплей підключається до виходів мікроконтролера через дільники напруги, оскільки працює від 3,3В.

Клавіатура призначена для керування роботою пристрою, та включає в себе енкодер і допоміжні 3 кнопки мембранного типу. Підключаються безпосередньо до входів МК, при цьому на відповідних входах МК будуть програмно ініціалізовані внутрішні резистори підтяжки.

Температурні сенсори. Зважаючи на значну потужність, для безпечного використання та надійності необхідно забезпечити моніторинг температури силових елементів пристрою. Для цього буде використано 4 цифрових сенсора температури DS18B20 компанії Dallas Semiconductor. Вони є досить прості у використанні і мають невеликі розміри, що дозволить розмістити сенсори безпосередньо на джерела тепла, а саме у спеціальній кишені вторинної обмотки потужних трансформаторів і на радіаторі польового транзистора блоку керування струмом заряду АКБ.

Блок керування вихідним струмом виконує функції захисту від перенавантаження та стабілізації заданого користувачем вихідного струму при роботі пристрою в режимі лабораторного блоку живлення, або при заряді АКБ. Блок виконаємо за лінійною схемою, тобто, при керуванні зайва потужність буде розсіюватись у вигляді тепла на радіаторі.

Схема представляє собою операційний підсилювач, до виходу якого під'єднано силовий MOSFET транзистор. Таке схемотехнічне рішення дозволяє досягти високої швидкодії при стабілізації, надійність і простоту при керуванні, високу лінійність вихідного струму і низький рівень вихідного шуму та перешкод у порівнянні з імпульсними схемами.

					ФЗ 8.17100.04.370 ПЗ	Арк
ЗМ.	Лист	№ докум.	Підпис	Дата		14

Змінюючи рівень напруги на прямому вході ОП, будемо керувати транзистором і, відповідно, значенням вихідного струму. Стабілізацію струму забезпечимо зворотнім зв'язком на інверсний вхід ОП.

Блок вимірювання вихідного струму побудуємо на операційному підсилювачі, на вхід якого буде поступати напруга з вимірювального резистора і після підсилення до необхідного значення, подана на вхід АЦП МК. По отриманому спаду напруги можна програмно розрахувати поточне значення вихідного струму.

Блок вимірювання напруг. Для функціонування пристрою в автоматичному режимі зарядки АКБ, необхідно забезпечити вимірювання наступних параметрів:

- значення напруги джерела живлення;
- значення напруги безпосередньо на клеммах АКБ;
- значення напруги на стоку керуючого транзистора;

Відповідно до отриманих значень, з урахуванням поточного струму, програмним забезпеченням буде формуватись процес зарядки акумулятора, проводиться його тестування чи відновлення працездатності.

Вимірювання напруг джерела та на стоку транзистора виконаємо за допомогою дільників для зниження рівня напруг до безпечного вхідного значення АЦП мікроконтролера. Додатково встановимо 5В стабілітрони для захисту входів МК. Напругу на клеммах АКБ будемо вимірювати диференційним способом за допомогою операційного підсилювача.

Блок 2 фазового керування потужністю. Так як у пристрої реалізовано функцію керування струмом, при його стабілізованому значенні знижуючи чи підвищуючи потужність джерела живлення у вигляді трансформатора, ми маємо можливість керувати вихідною напругою. Недоліком такого способу є некоректна робота схеми без навантаження, або при незначному вихідному струмі.

					ФЗ 8.17100.04.370 ПЗ	Арк
ЗМ.	Лист	№ докум.	Підпис	Дата		15

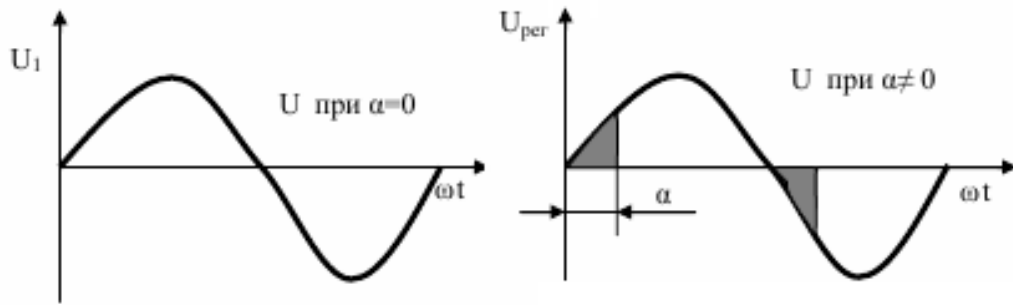


Рисунок 2.3 – принцип фазо-імпульсного керування

Керування потужністю трансформатора організуємо фазо-імпульсним методом за допомогою симістора.

У початковий момент часу на пристрій подається напруга L-N (фаза і нуль), при цьому керуючі імпульси не подаються, симістор закритий, струм у первинній обмотці трансформатора відсутній.

Спочатку мікроконтролер синхронізується з мережею, тобто визначає момент часу, в який напруга мережі L-N дорівнює нулю. Відраховується певний час T1 від моменту переходу через нуль і подається імпульс керування на симістор Рис 2.3. При цьому він відкривається і через навантаження протікає струм по шляху L-VS-Rн-N. При досягненні наступного переходу через нуль симістор автоматично закривається, знову відраховується час T1 відносно нового моменту переходу напруги через нуль і формується другий імпульс керування, симістор відкривається, і через навантаження протікає струм по шляху N-Rн-VS-L. Такий спосіб керування називається фазо-імпульсним.

Блок 1 фазового керування потужністю призначений для роботи з потужним трансформатором, що реалізує функцію контактного зварювання металів. Принцип дії та схемотехнічне рішення аналогічне до попереднього блоку, за виключенням застосування більш потужного симістора.

До переваг фазо-імпульсного способу керування можна віднести простоту в реалізації та високий ККД, що дозволяє керувати надзвичайно потужним навантаженням з мінімальними тепловими втратами.

3 РОЗРОБКА ФУНКЦІОНАЛЬНОЇ СХЕМИ

Згідно матеріалу, зазначеному в Розділі 2, на основі структурної, скомпонуємо функціональну схему пристрою Рис.3.1.

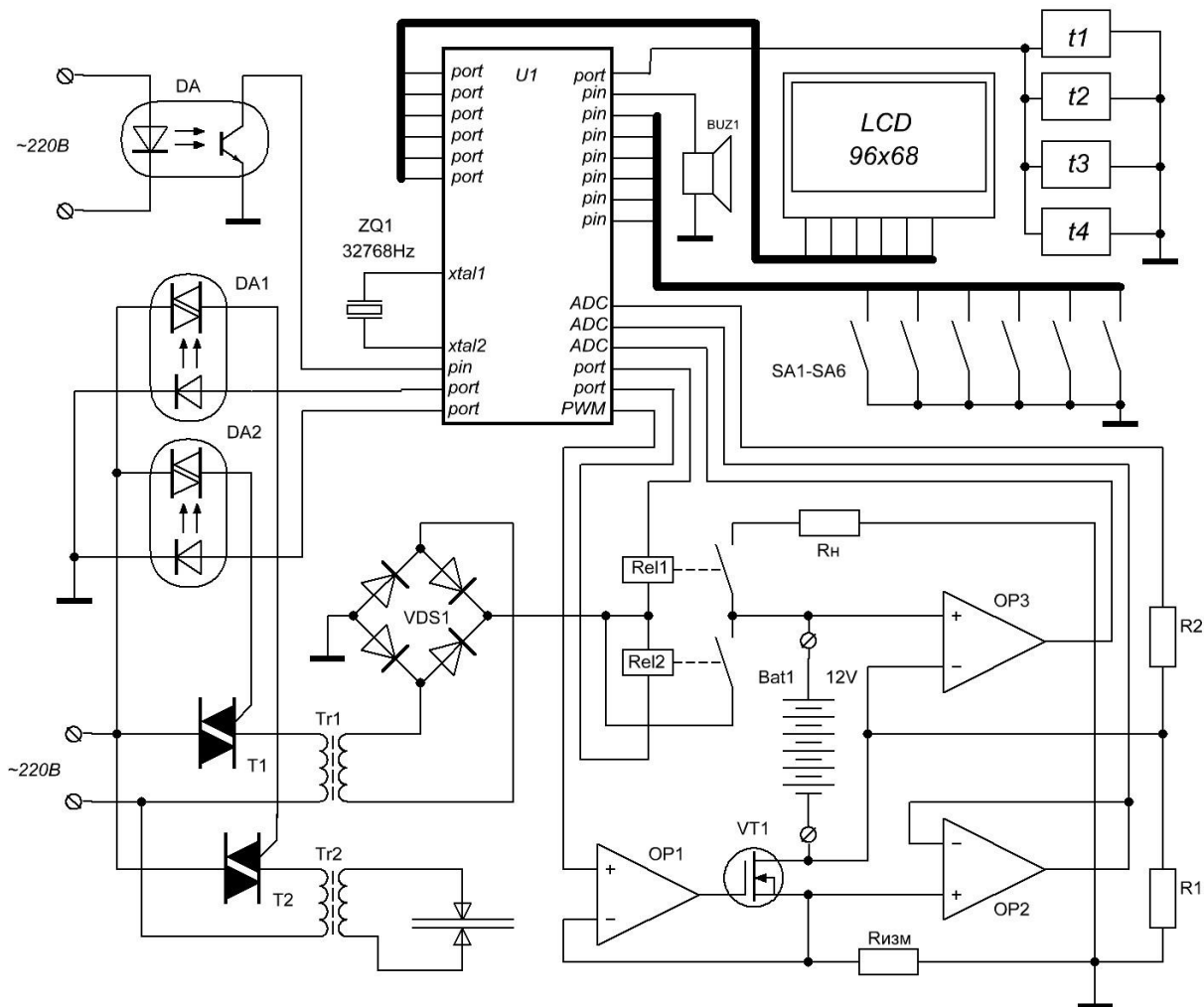


Рисунок 3.1 - функціональна схема пристрою

Схема складається із наступних основних елементів

- U1 – керуючий мікроконтролер Atmega32;
- LCD – графічний дисплей Nokia 1280;
- OP1-OP3 – операційні підсилювачі LM358N. OP1 використовується для керування транзистором VT1, OP2 та OP3 узгоджують та надають інформацію МК про поточні рівні вимірюваних ними напруг;
- DA – оптопара PC817 для синхронізації МК з мережею 220В;

ЗМ.	Лист	№ докум.	Підпис	Дата
-----	------	----------	--------	------

- T1, T2 – симістори серії ВТА для керування потужністю трансформаторів Tr1 та Tr2;
- -до вторинної обмотки Tr2 підключено електроди для контактного зварювання металів;
- DA1, DA2 – оптопары МOC3052, що забезпечують якісне керування симісторами T1, T2 та гальванічно розв`язують низьковольтну частину пристрою з мережею 220В;
- VT1 n-канальні МОП транзистор, що керує струмом розряда та заряда АКБ;
- Rизм – низькоомний вимірювальний резистор, по спаду напруги на якому можна програмно визначити поточне значення діючого струму;
- Rн – потужне навантаження для розряда АКБ в режимі тестування;
- BAT1 – АКБ, що підлягає заряду чи тестуванню;
- RL1, RL2 – реле, що комутують АКБ в режими заряда або розряда;
- t1-t3 – цифрові температурні сенсори, що забезпечують контроль за температурою потужних складових пристрою, таких як трансформатори, радіатори польового транзистора та симісторів. t4 є зовнішнім сенсором для контролю за температурою АКБ в процесі заряджання.
- SA1-SA6 – органи керування роботою пристрою, уявляють собою енкодер та кнопки без фіксації;
- BUZ1 – п`єзоелектричний елемент для організації можливості звукового сповіщення пристроєм.

4 РОЗРОБКА ПРИНЦИПОВИХ ЕЛЕКТРИЧНИХ СХЕМ

4.1 Блок 1 фазового керування потужністю

Схема фазового регулятора потужності Рис 4.1 виконана на симісторі ВТА80-800, що керується спеціальним оптодрайвером МОС3052. Для синхронізації роботи симістора з мережею 220В, додано оптопари U3 та U4 (РС817), за допомогою яких мікроконтролер розпізнає моменти переходу мережі через нуль.

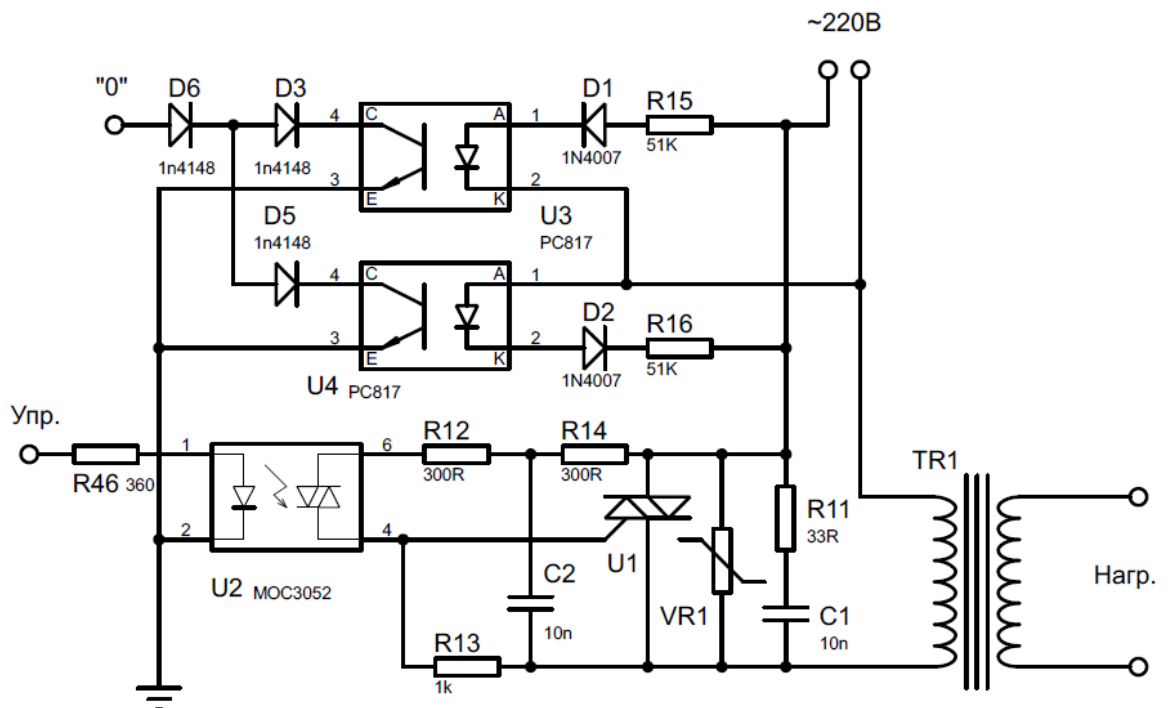


Рисунок 4.1 – фазовий регулятор потужності

Розрахуємо номінали R15 та R16. Робочий струм діода оптопари [16] для РС817 лежить у межах 5-10мА, приймаємо 6мА, відповідно номінали R15 та R16:

$$R15 = \sqrt{2} \cdot \frac{U_c}{I_F} = \sqrt{2} \cdot \frac{220}{0,006} = 51854(\text{Ом}),$$

Враховуючи стандартний ряд, приймаємо R15,16 = 51(кОм).

Орієнтовний номінал потужності R15,16 = 220·0,005 = 1,1(Вт)

Згідно даташита на оптосимістор МОС3052 [15], струм керуючого світлодіода приймаємо 10мА та розрахуємо номінал R46 за формулою:

					ФЗ 8.17100.04.370 ПЗ	Арк
ЗМ.	Лист	№ докум.	Підпис	Дата		19

$$R46 = \frac{+VCC - 1,5}{I_F} = \frac{5 - 1,5}{0,01} = 350(\text{Ом})$$

Враховуючи стандартний ряд, приймаємо $R46 = 360(\text{Ом})$.

Керуючий електрод МОС3052 може витримати струм до 1А, відповідно, можна розрахувати R12:

$$R12 = \sqrt{2} \cdot \frac{U_c}{I_{\max}} = \sqrt{2} \cdot \frac{220}{1} = 311(\text{Ом})$$

З урахуванням рекомендацій даташита [15] по встановленню захисних снаберних ліній R14, C2 і R11, C1, номінал R12 приймаємо 300(Ом)

R14, C2, R11, C1 встановлюються для зменшення випадкових перемикань симістора від можливих імпульсних перешкод при роботі з індуктивним навантаженням.

4.2 Схеми стабілізації та керування струмом

Лінійне керування струмом Рис 4.2 виконується за допомогою транзистора Q1, реле K1 вмикається при умові правильного підключення АКБ, а при перевищенні максимального струму чи виході з ладу силових елементів виконує роль захисту, від'єднуючи живлення силової частини схеми від джерела.

Наприклад, при струмі в 10А спад напруги на ньому буде 0,5 вольт. Ця напруга буде підсилена ОП і подана на АЦП МК для подальшої обробки та виведення на дисплей.

Приймаємо номінал вимірювального резистора $R39 = 0,05(\text{Ом})$, при максимальному струмі 15А, спад напруги на ньому складатиме:

$$U_{39} = 15 * 0,05 = 0,75(\text{В})$$

При цьому потужність R39 складає $Q_{39} = 0,75 * 15 = 11(\text{Вт})$,

Так як довготривалий робочий струм не буде перевищувати 10(A), приймаємо номінал $R_{39} = 0,05(\text{Ом})$, 10Вт

Для керування транзистором на вхід ОП повинна поступати напруга в межах $0 - 0,75(\text{В})$, відповідно розрахуємо номінали дільника напруги:

$$U_2 = (U_1 * R_{v1}) / (R_{23} + R_{v1}); \Rightarrow$$

$$R_{v1} = R_{23} / ((U_1 / U_2) - 1);$$

$$U_1 = 5\text{В}; U_2 = 0,75\text{В}; R_{23} = 20\text{k};$$

$$R_{v1} = 20000 / ((5 / 0,75) - 1); = 3529(\text{Ом}), \text{ приймаємо } R_{v1} = 5\text{k}$$

Забезпечимо постійну часу інтегруючої пари R_{21} , C_{12} на два порядки більше за період частоти частоти керуючого ШІМ: приймаємо $C_{12} = 10(\text{мкФ})$, відповідно:

$$R_{21} = 0,003 / 0,00001 = 300(\text{Ом})$$

R_{23} та C_{13} є додаткою парою для зменшення пульсацій на вході ОП, R_{24} та C_{14} розраховано з постійною часу, близькою до одного півперіода мережі, для виключення можливого самозбудження на високих частотах в керуних і силових ланцюгах

$$C_{14} = 10(\text{мкФ}), \text{ відповідно } R_{21} = 0,01 / 0,00001 = 1000(\text{Ом})$$

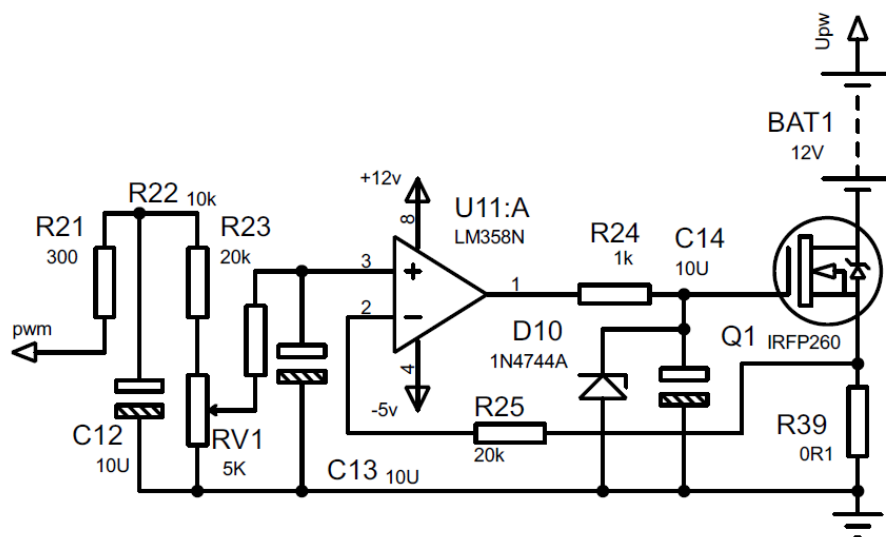


Рисунок 4.2 – схема стабілізації та керування струмом

Схема стабілізації та керування струмом Рис 4.2 працює наступним чином: за допомогою органів керування виставляється шпаруватість ШІМ на виході РВЗ МК, що відповідає бажаному струму на виході, ШІМ проходить через інтегруючі пари R21, C12, R23, C13 та дільник напруги R₂₃, R_{v1} де формується до певного постійного значення напруги і поступає на вхід 3 ОП (припустимо 0,5В, що відповідає бажаному струму 10А). Одночасно, на вході 2 ОП присутня напруга з вимірювального резистора R39 (наприклад 0,3В, що відповідає поточному струму 6А). Прагнучи урівняти значення на обох входах, ОП підвищить значення напруги на виході 1, чим забезпечить відкриття транзистора Q1 та зростання струму до моменту відповідності поточного та бажаного значення вихідного струму, в даному випадку 10А.

Недоліком такої лінійної схеми є низький ККД, бо зайва потужність буде виділена у вигляді тепла на Q1, тому для зниження цих втрат буде додано допоміжну схему регулювання вихідної напруги та потужності трансформатора ТР2 Рис 4.3. Таким чином можна забезпечити мінімально необхідну величину напруги джерела живлення для досягнення заданого користувачем вихідного струму. Також, можливість автоматичного керування як струмом так і напругою значно розширює функціонал пристрою, як для формування різних режимів заряду АКБ, так і для використання його як лабораторний блок живлення.

4.3 Блок 2 фазового керування потужністю

Схема регулювання вихідної напруги та потужності трансформатора ТР2 Рис 4.3 є майже аналогічною до схеми Рис 4.1, що керує потужністю трансформатора ТР1 для блоку контактного зварювання металу. Схема і розрахунки ті самі, що наведені у підрозділі 4.1, за винятком застосування менш потужного симістора U9 (ВТА20-800)

					ФЗ 8.17100.04.370 ПЗ	Арк
ЗМ.	Лист	№ докум.	Підпис	Дата		22

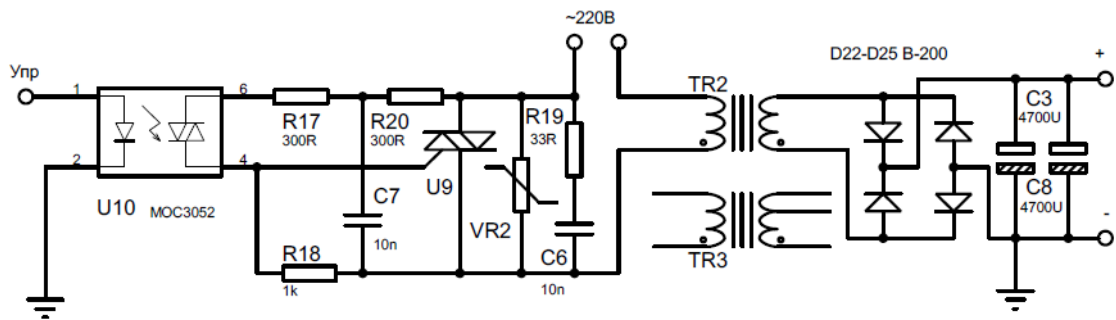


Рисунок 4.3 - Схема регулювання вихідної напруги та потужності трансформатора TR2

При роботі пристрою в режимі джерела струму, керування напругою буде проводитись по результатам обчисленого програмою значення теплових втрат та поточної напруги джерела живлення. Знаючи вихідний струм та напругу на стоку транзистора Q1, можна оцінити загальні теплові втрати на Q1, R39 та інших елементах, порівняти їх з допустимими, чи заданим користувачем пристрою значенням.

4.4 Схема вимірювання напруги

Організуємо вимірювання напруги на стоку транзистора Q1 за допомогою дільника R35 R36 та вхідної напруги за допомогою дільника R48 R49 Рис 4.4

Максимальна вхідна напруга АЦП МК [1], [5] дорівнює +5В, верхня межа регулювання напруги джерела живлення відповідає +16В, розрахуємо номінали дільників напруги R35 R36 та R48 R49:

$$U_2 = (U_1 * R_{35}) / (R_{36} + R_{35}); \Rightarrow$$

$$R_{35} = R_{36} / ((U_1 / U_2) - 1);$$

$$U_1 = 16\text{В}; U_2 = 5\text{В}; R_{36} = 10\text{k};$$

$$R_{35} = 10000 / ((16 / 5) - 1); = 4545(\text{Ом}), \text{приймаємо } R_{35} = 4,7\text{k}$$

Номінали R48 R49 будуть аналогічними.

Після дільників встановлено стабілітрони D13 та D35, які разом із R36 і R49 утворюють параметричні 5В стабілізатори для захисту входів АЦП МК, від можливого перевищення значення допустимої вхідної напруги.

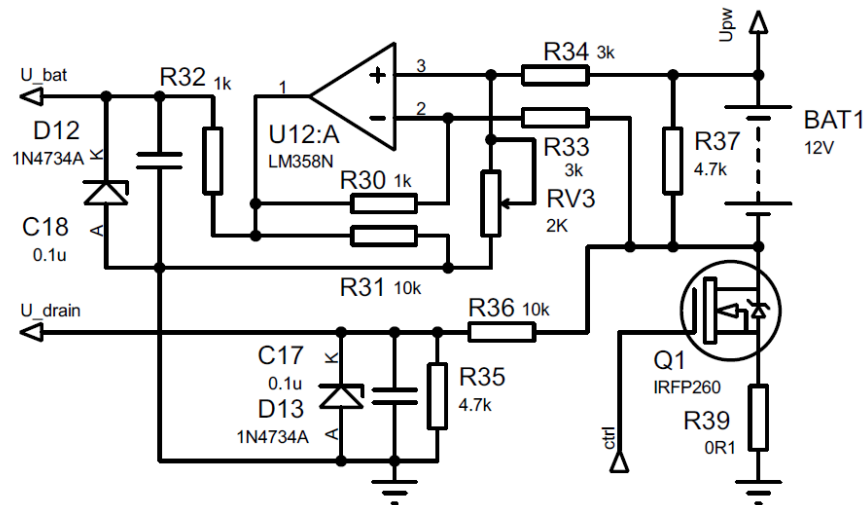


Рисунок 4.4 – Схема вимірювання напруги заряду АКБ

Вимірювання напруги АКБ буде проводитись безпосередньо з контактних клем диференційним способом Рис 4.4. Для цього застосуємо вільний елемент ОП U11.

При роботі в режимі блока живлення, максимальний рівень вимірюваної напруги досягає 16В, тому для узгодження із входом АЦП МК розрахуємо елементи схеми при необхідному коефіцієнті підсилення ОП = 5/16 = 0,31.

Вихідний рівень ОП розраховується наступним чином:

$$U_{\text{вих}} = U_2 \frac{(R33 + R30) \cdot Rv3}{(R34 + Rv3) \cdot R33} - U_1 \frac{R30}{R33} - U_{3M} \left(1 + \frac{R30}{R33}\right)$$

Якщо прийняти $R30/R33 = Rv3/R34$, формула має наступний вигляд:

$$U_{\text{вих}} = (U_2 - U_1) \frac{R30}{R33} - U_{3M} \left(1 + \frac{R30}{R33}\right)$$

$$R30 = \frac{(U_{\text{вих}} + U_{3M}) \cdot R33}{U_2 - U_1 - U_{3M}}$$

припустимо, що $R33 = R34 = 3000(\text{Ом}) \Rightarrow$

$$R30 = \frac{(5 + 0.0002) \cdot 3000}{16 - 0.0002} = 937(\text{Ом})$$

Приймаємо $R30 = 1\text{к}$

На виході ОП встановлено стабілітрон D12, який разом із R32 утворює параметричний 5В стабілізатор для захисту входу АЦП МК, від можливого перевищення значення допустимої вхідної напруги.

4.5 Схема вимірювання вихідного струму

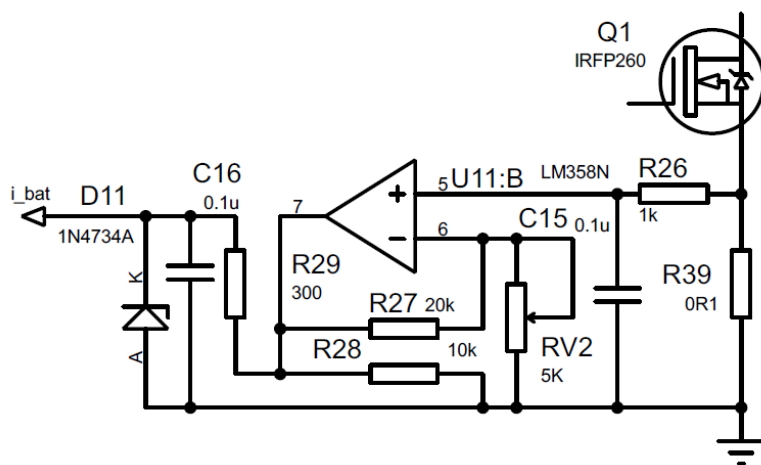


Рисунок 4.5 – Схема вимірювання вихідного струму

Для вимірювання значення вихідного струму використаємо останній вільний елемент U11 Рис 4.5, схема ввімкнення ОП є подібною до попередньої Рис 4.4

Вимірювальним елементом є резистор $R39$, по спаду напруги на якому можна програмно розрахувати поточне значення вихідного струму. Номінал та потужність $R39$ було обчислено у підрозділі №4.2

Рівень вхідної напруги ОП при максимальному значенні струму 15А, відповідає спаду напруги 0,75В на $R39$, тому для узгодження із входом АЦП МК розрахуємо елементи схеми при необхідному коефіцієнті підсилення ОП = $5/0,75 = 6,66$.

Згідно до рекомендацій даташита на LM358 [17], вихідний рівень ОП розраховується наступним чином:

$$U_{\text{вих}} = U_5 \left(1 + \frac{R27}{Rv2} \right)$$

приймаємо, що $R27 = 20\text{k} \Rightarrow$

$$Rv2 = \frac{U_5 \cdot R27}{U_{\text{вих}} - U_5} = \frac{0,75 \cdot 20000}{5 - 0,75} = 3529(\text{Ом})$$

Враховуючи розраховане значення 2/3 діапазона та стандартний ряд, визначимо номінал $Rv2 = 4,7(\text{kОм})$

4.5 Температурні сенсори DS18B20

Для контролю температури потужних складових пристрою використаємо 3 внутрішніх та 1 зовнішній сенсор температури DS18B20 компанії Dallas Semiconductor [18].

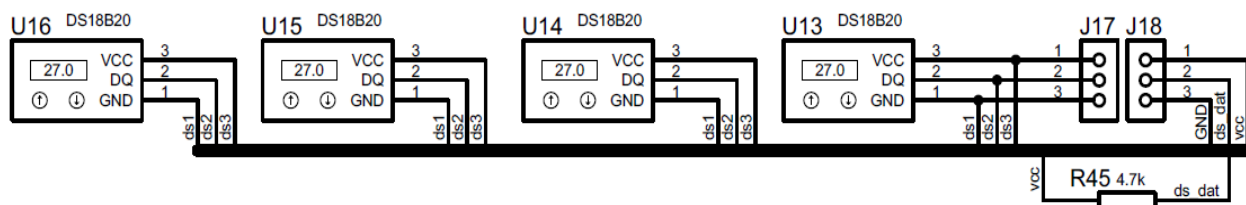


Рисунок 4.6 - сенсор температури DS18B20

DS18B20 це цифровий датчик температури, з розрядністю перетворення 9 - 12 розрядів і функцією тривожного сигналу контролю за температурою. Параметри контролю можуть бути задані користувачем і збережені в енергонезалежній пам'яті датчика.

DS18B20 обмінюється даними з мікроконтролером за однопровідною лінією зв'язку, використовуючи протокол інтерфейсу 1-Wire.

У мікросхемі DS18B20 для обміну даними використовується спеціалізований протокол 1-Wire корпорації Dallas. Для лінії зв'язку потрібен підтягуючий резистор, тому всі пристрої фізично підключені до однієї загальної шини і використовують вихід з трьома станами або вихід типу відкритий стік. У цій системі з однією шиною, мікроконтролер визначає наявність пристроїв на шині і обмінюється з ними, використовуючи унікальну адресу для кожного пристрою - 64-розрядний код. Оскільки кожен термодатчик має унікальний код, то число пристроїв, підключених до шини, практично ні чим не обмежена.

64-бітний ПЗП (ROM) зберігає унікальний серійний код пристрою. Оперативна пам'ять містить:

- значення вимірної температури (2 байта);
- верхній і нижній порогові значення тривожного сигналу (Th, Tl);
- реєстр конфігурації (1 байт).

Через реєстр конфігурації можна встановити розрядність перетворення термодатчика. Розрядність може бути задано 9, 10, 11 або 12 біт. Реєстр конфігурації і пороги тривожного сигналу містяться в (EEPROM).

Таблиця 4.1 – послідовність роботи МК з DS18B20

Режим	Шина даних	Команда
TX	Reset	МК формує імпульс скидання.
RX	Presence	DS18B20 відповідають імпульсом присутності.
TX	55h	МК виконує команду відповідності ROM кода.
TX	64-біт ROM код	МК надає ROM код DS18B20.
TX	44h	МК посилає команду конвертації температури.
TX	Reset	МК формує імпульс скидання.
RX	Presence	DS18B20 відповідають імпульсом

		присутності.
TX	55h	МК виконує команду відповідності ROM кода.
TX	64-біт ROM код	МК надає ROM код DS18B20.
TX	BEh	МК посилає команду читання з пам'яті.
RX	9 байтів даних	МК читає всю оперативну пам'ять, включаючи циклічний код CRC, обчислює CRC для перших восьми байтів і порівнює з прийнятим кодом. Якщо коди різні, МК читає пам'ять повторно.

DS18B20 вимірює температуру в градусах за шкалою Цельсія. Результат вимірювання представляється як 16-розрядний, знак температури надається в додатковому коді. Біт знака (S) дорівнює 0 для позитивних чисел і дорівнює 1 для негативних. При розрядності 12 біт, у регістра температури задіяні всі біти. При розрядності 9 біт, не достовірне значення мають біти 0, 1 і 2. На Рис 4.7 показана відповідність цифрових кодів значенням температури.

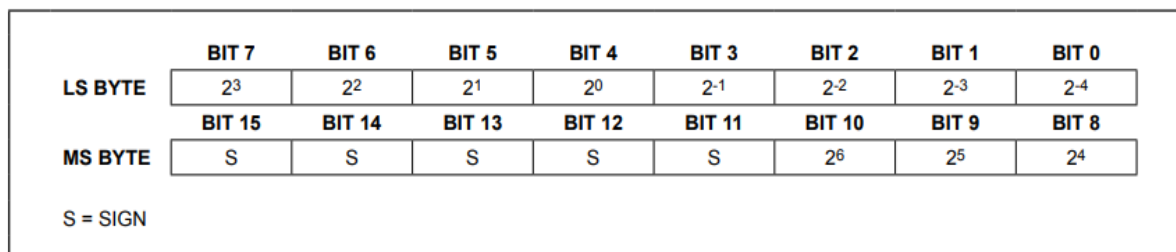


Figure 4. Temperature Register Format

Table 1. Temperature/Data Relationship

TEMPERATURE (°C)	DIGITAL OUTPUT (BINARY)	DIGITAL OUTPUT (HEX)
+125	0000 0111 1101 0000	07D0h
+85*	0000 0101 0101 0000	0550h
+25.0625	0000 0001 1001 0001	0191h
+10.125	0000 0000 1010 0010	00A2h
+0.5	0000 0000 0000 1000	0008h
0	0000 0000 0000 0000	0000h
-0.5	1111 1111 1111 1000	FFF8h
-10.125	1111 1111 0101 1110	FF5Eh
-25.0625	1111 1110 0110 1111	FE6Fh
-55	1111 1100 1001 0000	FC90h

Рисунок 4.7 - відповідність цифрових кодів значенням температури

Для обчислення температури потрібно:

- при позитивному значенні ($S = 0$) код перевести в десятковий і помножити на $0,0625 \text{ }^\circ\text{C}$.
- при від'ємному значенні ($S = 1$) спочатку необхідно інвертувати кожен розряд двійкового коду і додати 1. А потім перевести в десятковий і помножити на $0,0625 \text{ }^\circ\text{C}$.

Генерація циклічного коду (CRC)

Байти циклічного коду (CRC) розташовані в 64-бітовому ROM коді і в дев'ятому байті пам'яті SRAM. Циклічний код з ROM обчислюється для 56ти бітів коду ROM і розташовується в старшому байті ROM. Циклічний код з SRAM обчислюється з байтів 0 ... 7 SRAM. Циклічний код дозволяє контролювати правильність читання даних з DS18B20. МК обчислює циклічний код для отриманих даних і порівнює з прийнятим кодом. На підставі цього приймається рішення про коректність даних.

Поліном циклічного коду виглядає так:

$$C R C = X^8 + X^5 + X^4 + 1$$

МК може обчислити циклічний код використовуючи генератор полінома, за схемою на Рис 4.8. Він складається з регістра зсуву і логічних елементів типу "виключає АБО". Регістр зсуву спочатку знаходиться в стані 0. Біти надходять в регістр зсуву, починаючи з молодшого біта, коду з ROM або з SRAM, один біт в один такт зсуву. Після зсуву 56го біта ROM або старшого біта 7го байта SRAM, в регістрі зсуву буде обчислений циклічний код. Якщо сдвинути в генератор 8 біт ROM або SRAM, прийняті з DS18B20, то в разі правильних даних, регістр зсуву буде містити всі 0.

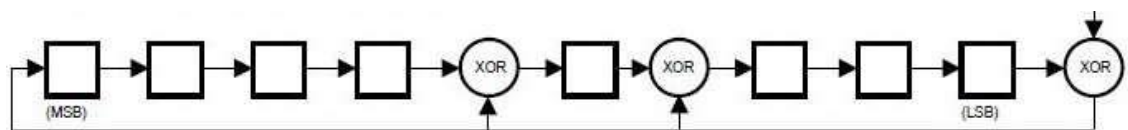


Рисунок 4.8 - генератор полінома для обчислення CRC

4.7 Мікроконтролер

Виходячи із вимог до приладу та спроектованих вище блоків, визначимо бажані характеристики мікроконтролера.

- кількість ліній вводу-виводу: не менше 25,
- кількість портів: не менше 4,
- кількість 8-ми бітних таймерів: не менше 2,
- кількість 16-ми бітних таймерів: не менше 1,
- розрядність внутрішнього АЦП: не менше 10біт,
- кількість входів зовнішніх переривань: не менше 2,
- об'єм флеш пам'яті: не менше 16кБайт,
- об'єм SRAM пам'яті: не менше 512Байт,
- частота тактування: не менше 8МГц.

Для забезпечення запланованого функціоналу та параметрів, а також враховуючи що використання графічного дисплея потребує значного об'єму пам'яті МК, побудуємо пристрій на основі 8-ми бітного мікроконтролера Atmega32 фірми ATMEL [1] Рис.4.9. Основні характеристики наведено на Рис.4.10

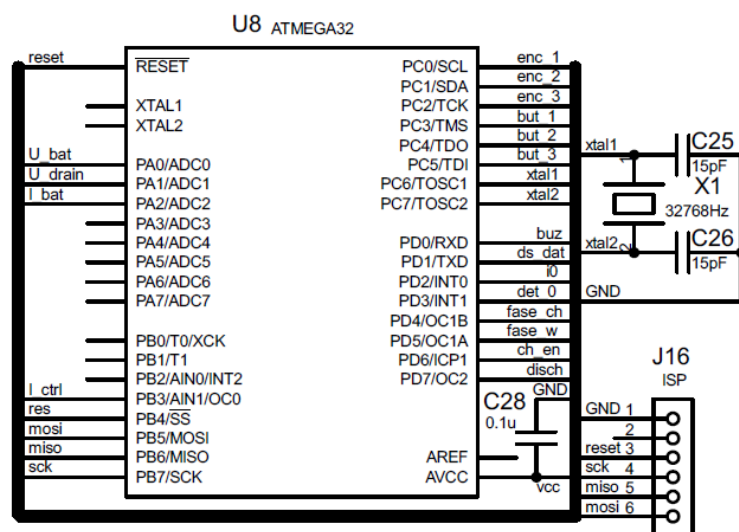


Рисунок 4.9 – мікроконтролер Atmega32

Features

- High-performance, Low-power Atmel® AVR® 8-bit Microcontroller
- Advanced RISC Architecture
 - 131 Powerful Instructions – Most Single-clock Cycle Execution
 - 32 × 8 General Purpose Working Registers
 - Fully Static Operation
 - Up to 16 MIPS Throughput at 16MHz
 - On-chip 2-cycle Multiplier
- High Endurance Non-volatile Memory segments
 - 32Kbytes of In-System Self-programmable Flash program memory
 - 1024Bytes EEPROM
 - 2Kbytes Internal SRAM
 - Write/Erase Cycles: 10,000 Flash/100,000 EEPROM
 - Data retention: 20 years at 85°C/100 years at 25°C⁽¹⁾
 - Optional Boot Code Section with Independent Lock Bits
 - In-System Programming by On-chip Boot Program
 - True Read-While-Write Operation
 - Programming Lock for Software Security
- JTAG (IEEE std. 1149.1 Compliant) Interface
 - Boundary-scan Capabilities According to the JTAG Standard
 - Extensive On-chip Debug Support
 - Programming of Flash, EEPROM, Fuses, and Lock Bits through the JTAG Interface
- Peripheral Features
 - Two 8-bit Timer/Counters with Separate Prescalers and Compare Modes
 - One 16-bit Timer/Counter with Separate Prescaler, Compare Mode, and Capture Mode
 - Real Time Counter with Separate Oscillator
 - Four PWM Channels
 - 8-channel, 10-bit ADC
 - 8 Single-ended Channels
 - 7 Differential Channels in TQFP Package Only
 - 2 Differential Channels with Programmable Gain at 1x, 10x, or 200x
 - Byte-oriented Two-wire Serial Interface
 - Programmable Serial USART
 - Master/Slave SPI Serial Interface
 - Programmable Watchdog Timer with Separate On-chip Oscillator
 - On-chip Analog Comparator
- Special Microcontroller Features
 - Power-on Reset and Programmable Brown-out Detection
 - Internal Calibrated RC Oscillator
 - External and Internal Interrupt Sources
 - Six Sleep Modes: Idle, ADC Noise Reduction, Power-save, Power-down, Standby and Extended Standby
- I/O and Packages
 - 32 Programmable I/O Lines
 - 40-pin PDIP, 44-lead TQFP, and 44-pad QFN/MLF
- Operating Voltages
 - 2.7V - 5.5V for ATmega32L
 - 4.5V - 5.5V for ATmega32
- Speed Grades
 - 0 - 8MHz for ATmega32L
 - 0 - 16MHz for ATmega32
- Power Consumption at 1MHz, 3V, 25°C
 - Active: 1.1mA
 - Idle Mode: 0.35mA
 - Power-down Mode: < 1µA



**8-bit AVR®
Microcontroller
with 32KBytes
In-System
Programmable
Flash**

**ATmega32
ATmega32L**

2503Q-AVR-02/11



Рисунок 4.21 – основні характеристики мікроконтролера Atmega 32

						Φ3 8.17100.04.370 ПЗ	Арк
							31
ЗМ.	Лист	№ докум.	Підпис	Дата			

При керуванні струмом заряда/розряда АКБ за допомогою режиму ШІМ, навіть зважаючи, що після проходження імпульсів через інтегруючі пари R21,C12 та R23,C13, а потім через операційний підсилювач U11:A на польовий транзистор надходитиме майже лінійний сигнал, але бажано забезпечити частоту ШІМ за межею слухового діапазону (>20кГц), тому частота тактування, з урахуванням можливостей таймера T0 мікроконтролера[6], становитиме:

$$f_{clk_I/O} = f_{OCnX} \cdot N \cdot TOP$$

де $f_{clk_I/O}$ – частота тактування,

f_{OCnX} – період переповнення таймера (частота ШІМ),

N – дільник частоти таймера,

TOP – межа лічильника таймера (256),

$$f_{clk_I/O} = 30000 \cdot 1 \cdot 0xFF = 7680000(\text{Гц})$$

Так як мікроконтролер має можливість тактування від внутрішнього генератора до частоти 8МГц, уточнимо частоту для вибраного режиму ШІМ :

$$f_{OCnX} = \frac{f_{clk_I/O}}{N \cdot TOP} = \frac{8000000}{0xFF} = 31250(\text{Гц})$$

Часові діаграми роботи таймера T0 в режимі ШІМ зображено на Рис.4.22

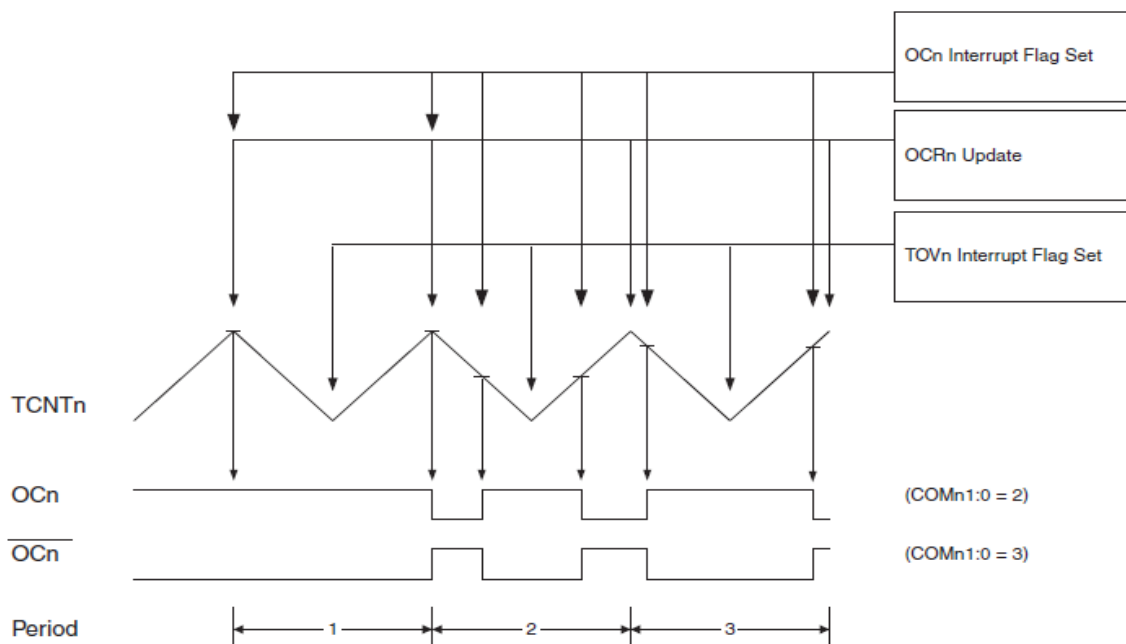


Рисунок 4.22 - Часові діаграми роботи таймера T0 в режимі ШІМ

Для спрощення внутрішнього відліку реального часу використаємо кварцевий резонатор X1 з частотою 32768Гц, та сконфігуруємо внутрішній таймер T2 на асинхронний режим тактування [7].

Так як таймер T2 має межу лічильника 256, разом із внутрішнім дільником частоти на 128 отримаємо одне переповнення лічильника за 1 секунду:

$$T = \frac{\text{clk}_{T2s}}{N \cdot TOP} = \frac{32768}{128 \cdot 256} = 1(\text{Гц})$$

Оскільки немає потреби в обчисленні та індикації днів, тижнів та інших значних проміжків часу, організуємо відлік наступним чином:

- створимо змінну для зберігання загальної кількості секунд (наприклад Time), яку в кожному перериванні по переповненню лічильника таймера T2 будемо збільшувати на 1;
- відповідно до режиму роботи розрахунок загального часу та важливих проміжків для індикації обчислимо за наступними співвідношеннями:

$$h = \text{time}/3600;$$

$$m = (\text{time} - h \cdot 3600)/60;$$

$$s = \text{time} - h \cdot 3600 - m \cdot 60;$$

де h – години, m – хвилини, s – секунди.

Наприклад змінна time = 57891,

$$h = 57891/3600 = 16;$$

$$m = (57891 - 16 \cdot 3600)/60 = 4$$

$$s = 57891 - 16 \cdot 3600 - 4 \cdot 60 = 51$$

це відповідає наступному часу: 16:04:51

4.8 Графічний дисплей

Для виводу інформації, використовуємо графічний монохромний дисплей від мобільного телефону Nokia, який керується внутрішнім контролером PCF8814 [2] і дозволяє виводити на екран 96 пікселів по горизонталі і 68 пікселів по вертикалі. Відлік починається від верхнього лівого кута.

Дані про зображення записуються побайтно у внутрішню пам'ять (DDRAM) контролера PCF8814. Адресація по горизонталі (X) йде попіксельно і змінюється від 0 до 95, а по вертикалі (Y) задається номер банку, який змінюється від 0 до 8. Адреси поза цими діапазонами не допускаються. Всього є 9 банків по 96 байт. У дев'ятому банку використовується тільки 1 піксель. Залежно від режиму, значущим є або старший біт переданого байта, або молодший.

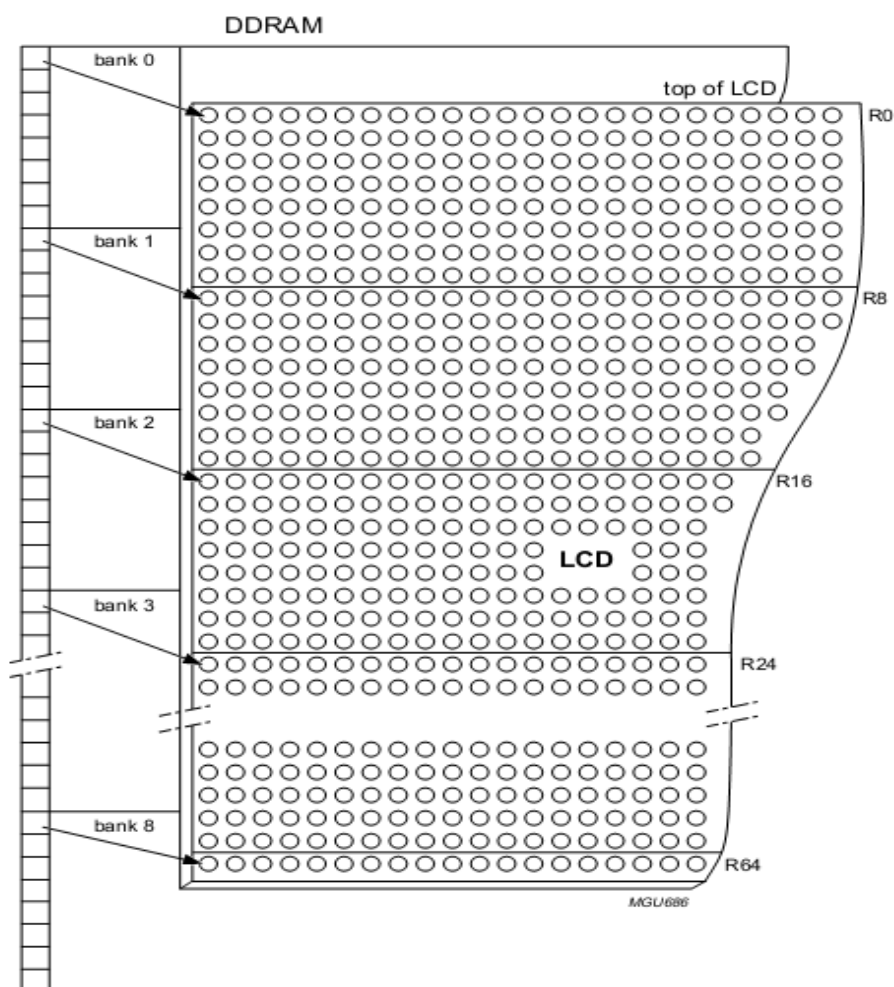


Рисунок 4.11 - адресація пам'яті

Адресація пам'яті можлива в двох режимах - горизонтальному і вертикальному. У горизонтальному режимі під час запису кожного байти в пам'ять лічильник X збільшується на одиницю. При досягненні межі по горизонталі (95) він скидається в 0, а лічильник Y збільшується на одиницю.

У вертикальному режимі адресації, при кожному записі байта в пам'ять, на одиницю збільшується лічильник Y, при досягненні межі (8) він скидається в 0, а горизонтальний лічильник збільшується на одиницю.

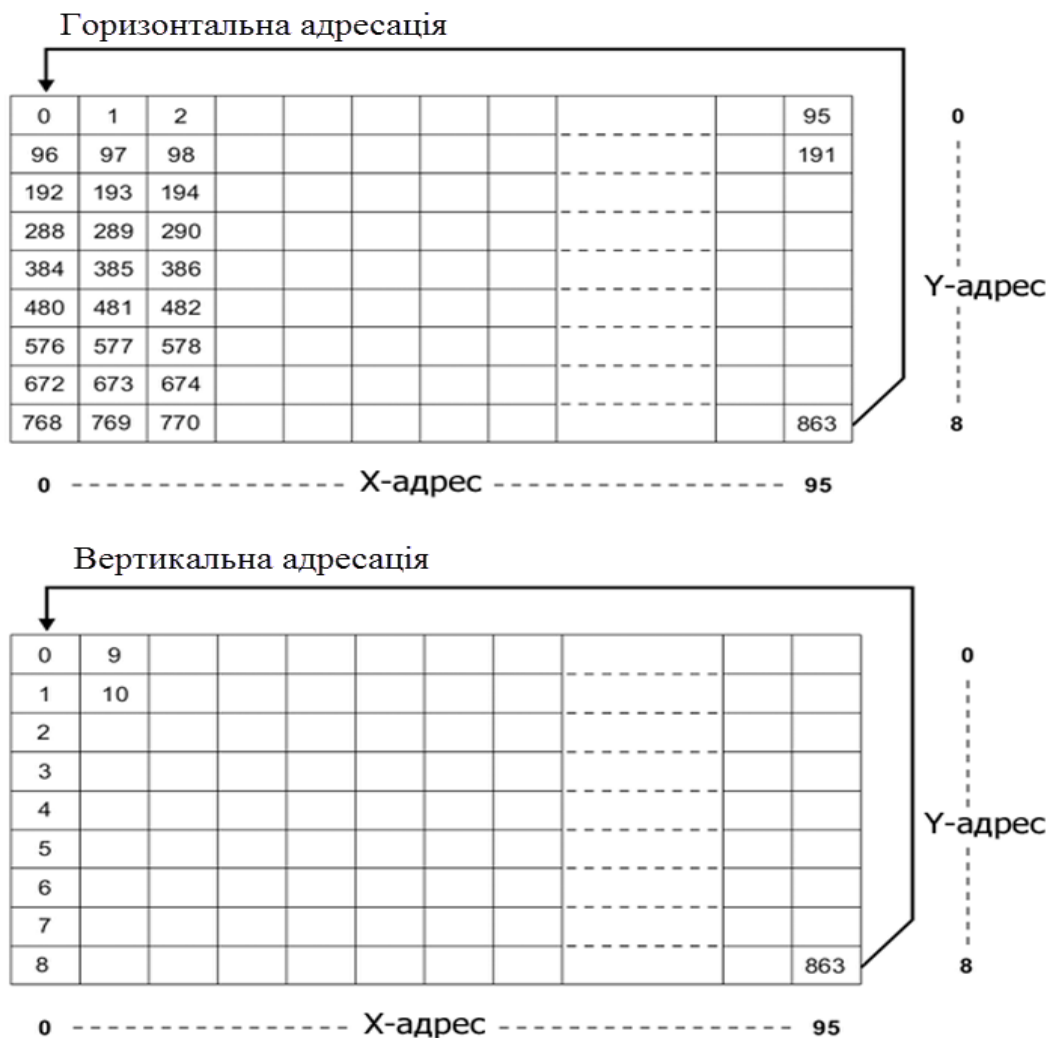


Рисунок 4.12 - режими адресації

Дисплей дозволяє тільки записувати дані в пам'ять (DDRAM) контролера дисплея. Читання з пам'яті не реалізовано, хоча сам контролер PCF8814 це дозволяє. Тому, що б визначити, що записано в пам'яті дисплея, необхідно в мікроконтролері програмно організувати буфер з даними для дисплея і відстежувати те, що записано, по ньому.

У різних модифікаціях дисплеїв контролер РСF8814 може підключатися до мікроконтролера за допомогою:

- SPI по трьох лініях,
- SPI по чотирьох лініях,
- шини I2C,
- послідовного 3-х проводового інтерфейсу.

У вибраному дисплеї Рис 4.12.передбачено підключення по 4 лініях SPI.

При підключенні використовуються сигнали:

- SCLK - лінія тактування,
- SDA - лінія передачі даних і команд,
- CS - лінія, що визначає передачу команди або даних,
- RST - лінія апаратного скидання.

Контролер дисплея працює від 3,3В, тому на виводи VDDI та VDD напруга подається через параметричний стабілізатор на елементах R6, D4, а на сигнальні SCLK, SDA, CS, RST, безпосередньо з мікроконтролера, через дільники напруги R2-R5, R7-R10

Розрахуємо номінали дільника напруги:

$$U_2 = (U_1 * R_7) / (R_2 + R_7); \Rightarrow$$

$$R_7 = R_2 / ((U_1 / U_2) - 1);$$

$$U_1 = 5В; U_2 = 3,3В; R_2 = 2к;$$

$$R_7 = 2000 / ((5 / 3,3) - 1); = 3882(Ом), \text{ приймаємо } R_7 = 3,9к$$

Розрахуємо номінали параметричного стабілізатора:

Для забезпечення напруги 3,3В використаємо стабілітрон 1n4728A

Спад напруги на $U_{R6} = 5 - 3,3 = 1,7(В)$,

Враховуючи прогнозований ток стабілізації 7мА:

$$R_6 = U_{R6} / I_{ст} \Rightarrow$$

$$R_6 = 1,7 / 0,007 = 242(Ом), \text{ приймаємо } R_6 = 250(Ом)$$

					ФЗ 8.17100.04.370 ПЗ	Арк
ЗМ.	Лист	№ докум.	Підпис	Дата		36

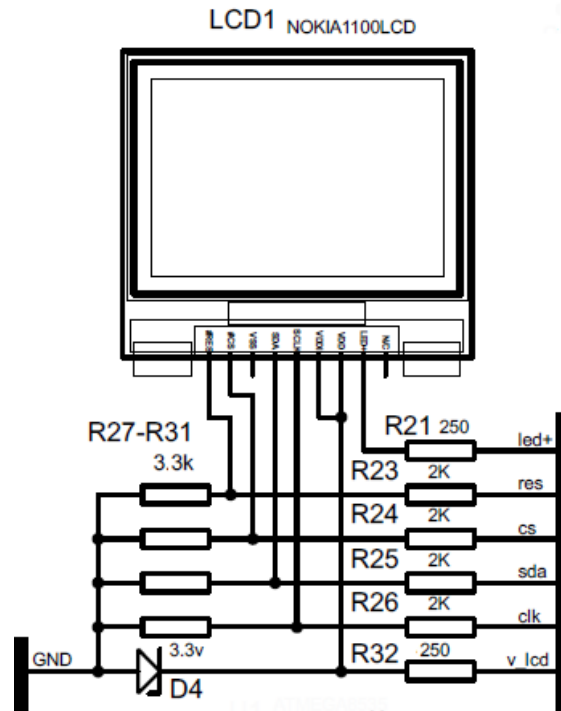


Рисунок 4.12 – схема підключення графічного дисплея

4.9 Блок підготовки живлення

Блок підготовки живлення пристрою Рис 4.13 уявляє собою стабілізатори напруг $+5V$, $+12V$ та $-5V$ відносно загального терміналу, для роботи мікроконтролера, логіки, операційних підсилювачів і т. д.

Стабілізатори включають в себе спеціалізовані ІМС LM7805, LM7905, LM7812, фільтруючі конденсатори C4, C5, C9, C10, C11, C27, C29, деякі з них встановлені безпосередньо біля виводів живлення мікроконтролера [4], дисплея та ін. заводочутливих елементів пристрою. Також у кожен шинув встановлено стабілітрони D7, D8, D9 та індуктивності L1-L3 для видалення з лінії VCC можливих короточасних перешкод підвищеного рівня при роботі пристрою на потужне індуктивне навантаження.

Також, частково реалізовано гальванічну розв'язку від силової частини пристрою для зменшення можливих перешкод по ланцюгам живлення, на входи стабілізаторів подається попередньо спрямлена діодами D14 – D21 напруга від окремого трансформатора TR3.

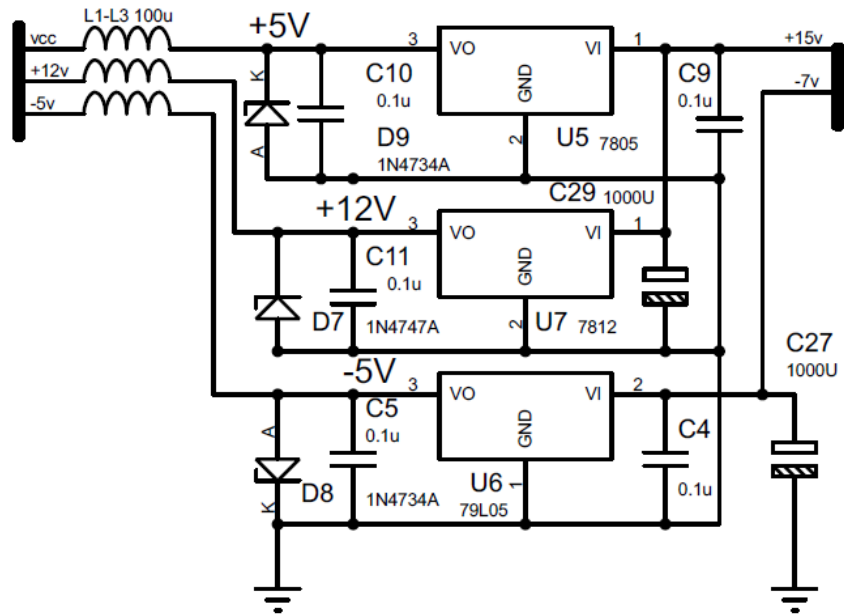


Рисунок 4.13 - блок підготовки живлення

4.10 Клавіатура

Клавіатура Рис.4.14 призначена для керування роботою пристрою, та включає в себе енкодер, зображений у вигляді кнопок But1-But3, і допоміжні 3 кнопки мембранного типу But4-But6.

Для прискорення спрощення програмної обробки використовуються діоди D26-D31, які анодами підключено до виводу зовнішнього переривання INT0 мікроконтролера. При натисканні будь-якої з кнопок на INT0 формується низький рівень, що викликає відповідне переривання основної програми і дозволяє якісно і своєчасно керувати роботою пристрою.

Паралельно до кнопок підключені конденсатори C19-C24, що запобігають шурхоту контактів, та стабілізують роботу мікроконтролера.

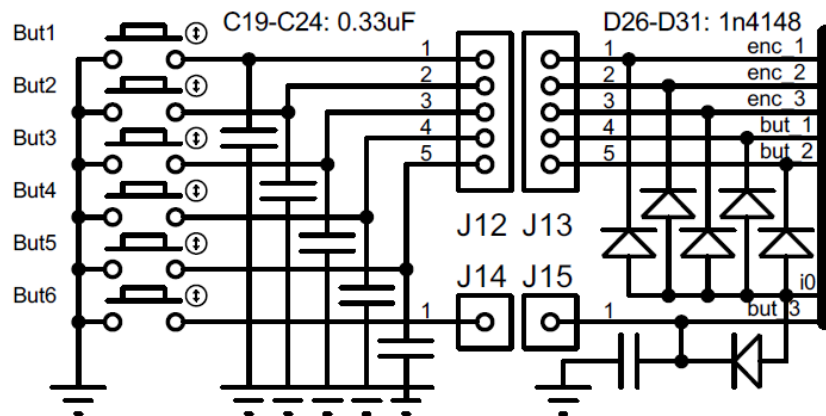


Рисунок 4.14 - клавіатура

На Рис.4.15 зображено осцилограму нестабільності контакту мембрани кнопки, що призводить до некоректної роботи пристрою. Як виявилось, шурхіт контактів для мембранного типу кнопок виникає лише при її відпусканні.

Конкретне конструктивне виконання використаних кнопок має частоту коливань контактної мембрани близько 30кГц, і час зростання сигналу до 2,5В становить 16мкс, що викликає багаторазові спрацювання за одне натискання.

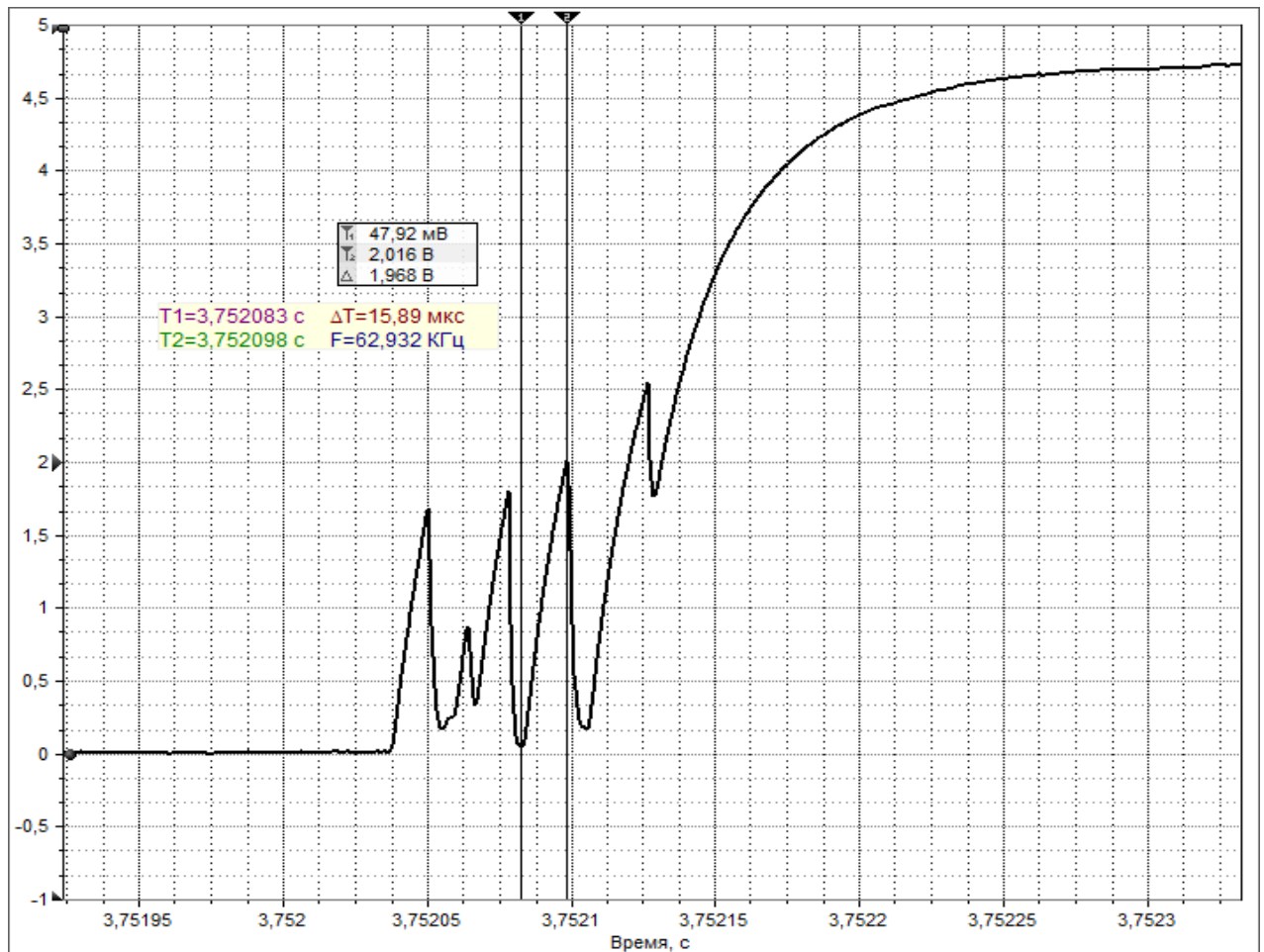


Рисунок 4.15 - осцилограма нестабільності контакту мембрани кнопки

Для запобігання некоректної роботи пристрою, паралельно до кнопок підключено фільтруючі конденсатори С19-С24, номіналом 0,33мкФ, що приймають заряд через підтягуючі резистори входів мікроконтролера.

Оптимальний номінал конденсаторів визначено експериментальним шляхом і для різних типорозмірів кнопок буде різним. Результат приведено на осцилограмі Рис.4.16. За рахунок сповільнення зростання фронту напруги до 3мс, шурхіт контакту повністю вилучається, при подальшому збільшенні номіналу конденсаторів можливе значне скорочення ресурсу кнопки.

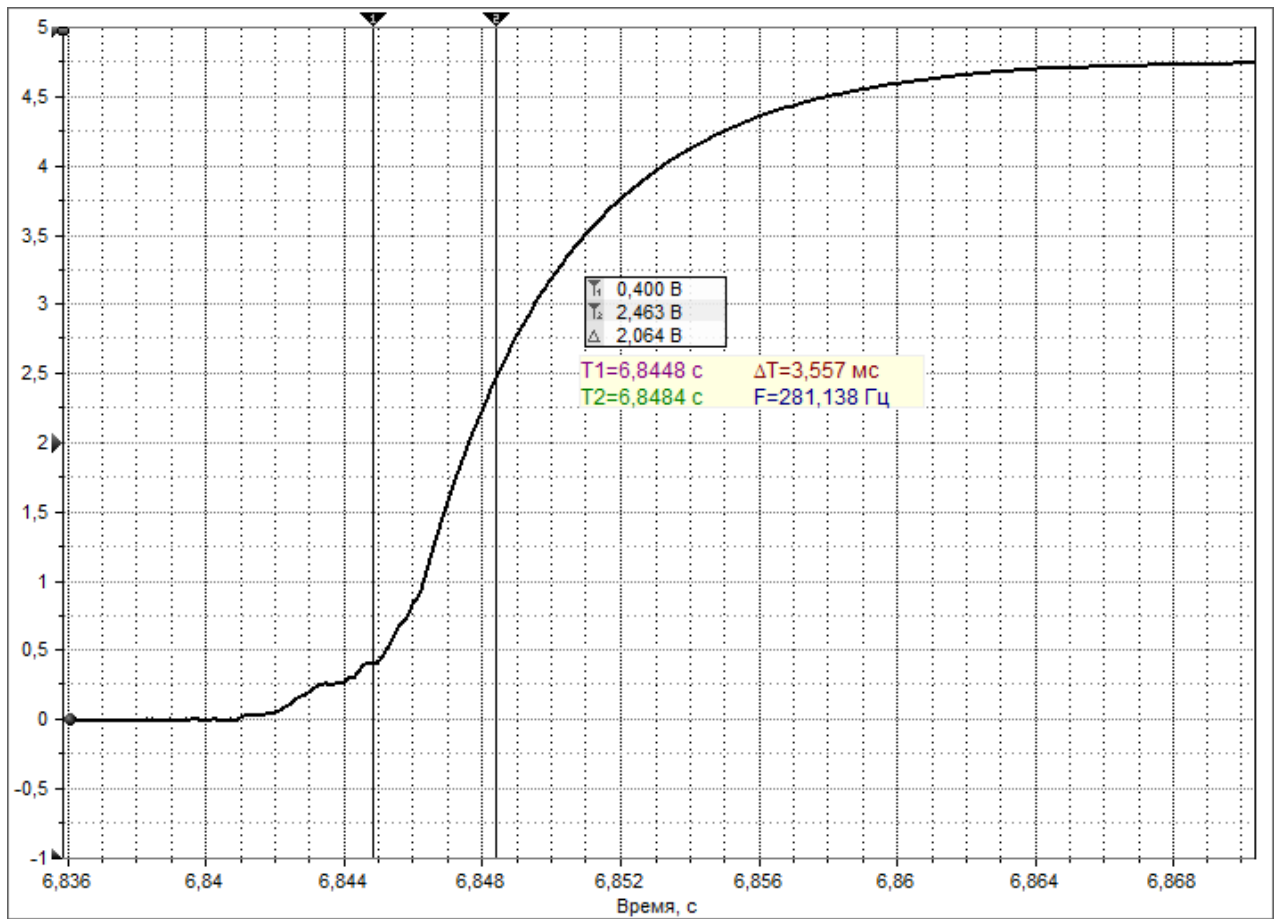


Рисунок 4.16 - осцилограма контакту мембрани кнопки при наявності конденсатора

4.11 Розробка загальної електричної схеми пристрою

Враховуючи попередній розділ, можна скомпонувати загальну принципову схему пристрою Рис 4.17.

Всі вищезазначені блоки пристрою об'єднаємо в електричну схему, що має наступні комунікаційні роз'єми:

J1, J2 – 14 пін, роз'єм підключення силової частини пристрою до блоку керування;

J12, J13 – 5 пін, роз'єм підключення енкодера та кнопок керування;

J17, J18 – 3 пін, роз'єм підключення температурних сенсорів;

J16 – 6 пін, роз'єм для програмування мікроконтролера;

J9 – вихід підключення навантаження, наприклад галогенової лампи, для можливості розрядки при проведенні тестування акумуляторів;

J10, J11 – клеми для підключення акумулятора;

J7, J8 – безпосередній вихід спрямленої напруги з трансформатора TP2;

J5, J6 – вихід на електроди для контактного зварювання металів;

J5, J6 – мережа 220В.

Розробка та формування принципової схеми виконувалась з використанням програмного комплексу PROTEUS.

					ФЗ 8.17100.04.370 ПЗ	Арк
ЗМ.	Лист	№ докум.	Підпис	Дата		42

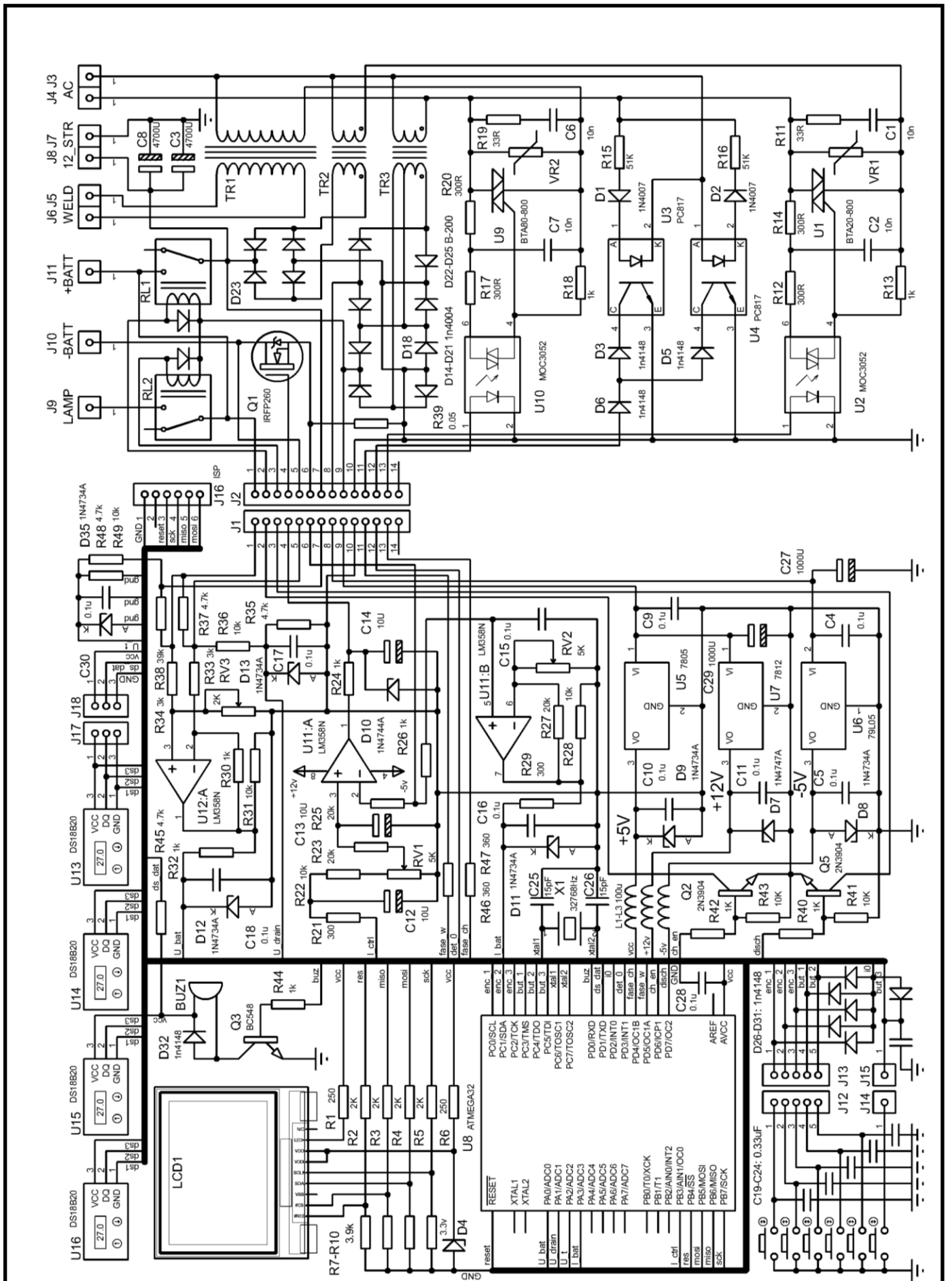


Рисунок 4.17 - загальна принципова схема пристрою

5 РОЗРОБКА АЛГОРИТМУ ТА ПРОГРАМНЕ ЗАБЕЗПЕЧЕННЯ

В загальному вигляді принцип роботи пристрою має виглядати наступним чином:

- на дисплей виводиться головне меню, де надається можливість вибору режиму роботи.
- передбачено наступні режими:
 1. Швидкий заряд АКБ;
 2. Повний заряд за принципом I-U-I-U;
 3. Тест АКБ;
 4. Контактне зварювання;
 5. Блок живлення.
- кожен режим роботи повинен мати персональне меню, де буде можливість ввести необхідні налаштування та параметри роботи пристрою у цьому режимі, а також здійснювати запуск та зупинку роботи;
- перед кожним запуском роботи пристрою, а також у деяких режимах роботи повинен відбуватися контроль за роботою вихідних елементів для попередження їх руйнування від перевантаження за струмом;
- постійно має контролюватися температура потужних складових пристрою;

У додатках наведено спрощені алгоритми роботи пристрою у деяких режимах, а саме:

лист 1 – формування пунктів меню, вибір режиму роботи користувачем та розгалудження програми по відповідних функціях.

лист 2 – алгоритм роботи в режимі повного заряду АКБ;

лист 3 - алгоритм роботи в режимі швидкого заряду АКБ;

Мікроконтролер взаємодіє із іншими блоками схеми за алгоритмами закладеної в нього мікропрограми.

					ФЗ 8.17100.04.370 ПЗ	Арк
ЗМ.	Лист	№ докум.	Підпис	Дата		44

Програма написана на мові програмування C за допомогою програмного забезпечення CodeVisionAVR та має в своєму складі розділи, функції обробки даних, додаткові бібліотеки, тому в Додатках А,Б,В,Г наведено частину коду що формує склад та позиції пунктів меню,що відображуються на графічному дисплеї, а також формування роботи пристрою в режимах контактного зварювання і заряду АКБ.

					ФЗ 8.17100.04.370 ПЗ	Арк
Зм.	Лист	№ докум.	Підпис	Дата		45

6 РЕАЛІЗАЦІЯ СПРОЕКТОВАНОГО ПРИСТРОЮ

Для можливості подальшого виготовлення пристрою, необхідно спроектувати друковану плату. Згідно [19] та виходячи із кількості компонентів орієнтовно визначимо бажаний розмір плат:

- блок керування - 90x70мм;
- силовий блок – 105x95мм;
- плата клавіатури – 15x90мм.

За допомогою програмного комплексу PROTEUS, враховуючи рекомендації [1], [4], [5], [8], [17] виконаємо розміщення елементів пристрою, двостороннє розведення провідників, розмістимо необхідні перехідні отвори Рис.6.1, Рис.6.2

За результатами проектування, у спеціальному програмному модулі комплексу сформуємо 3Д модель пристрою Рис.6.3.

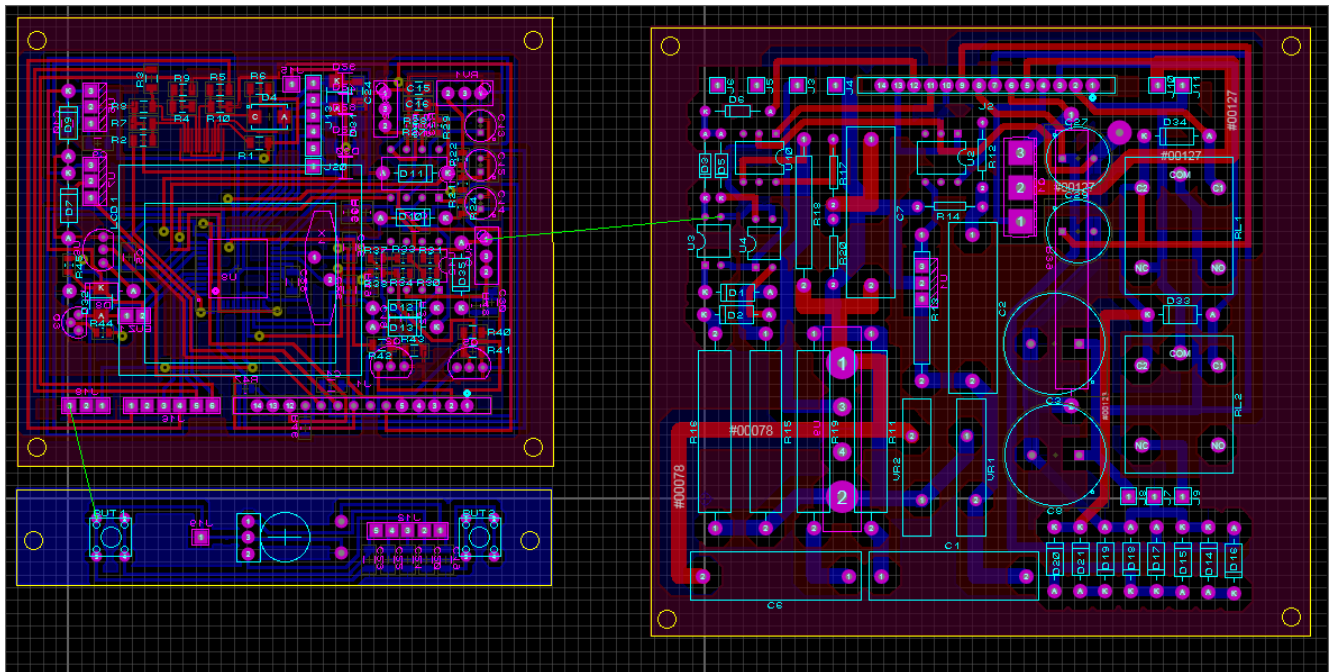


Рисунок 6.1 – друковані плати пристрою

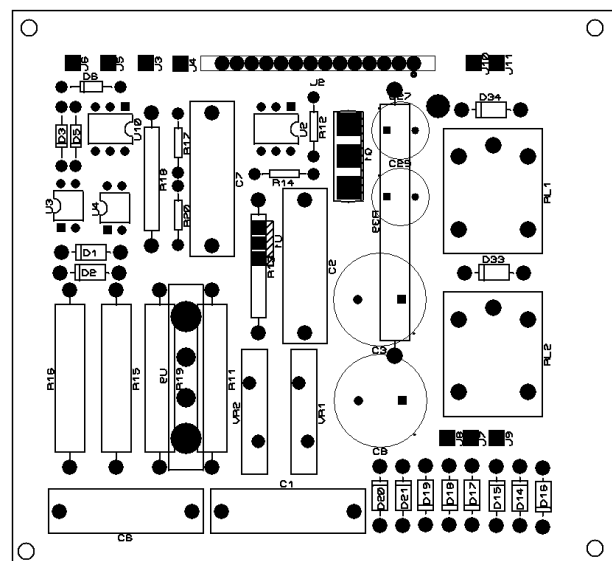
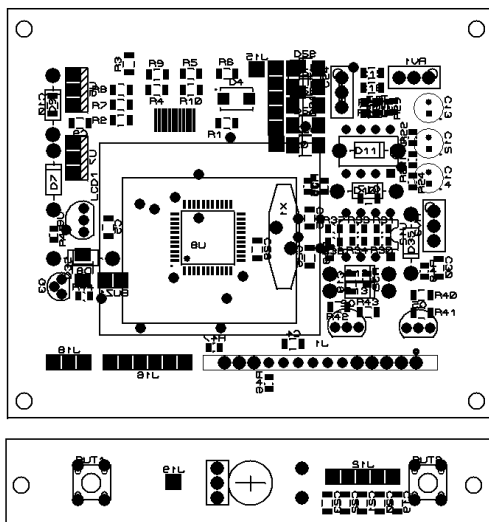
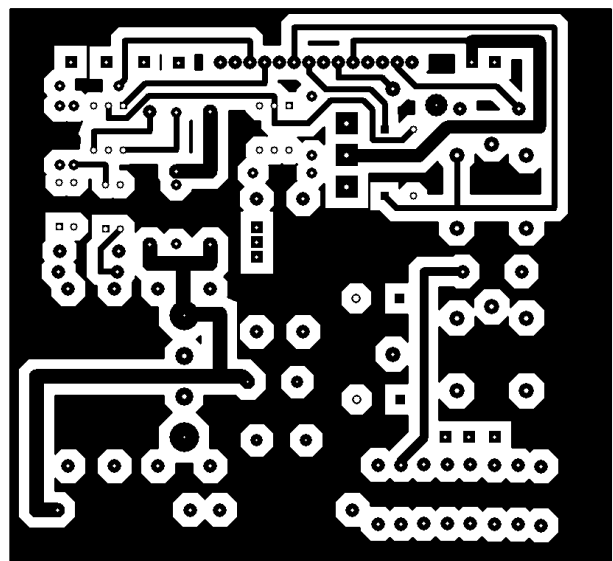
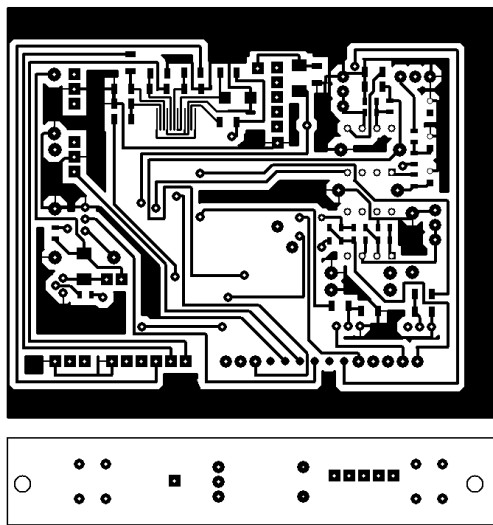
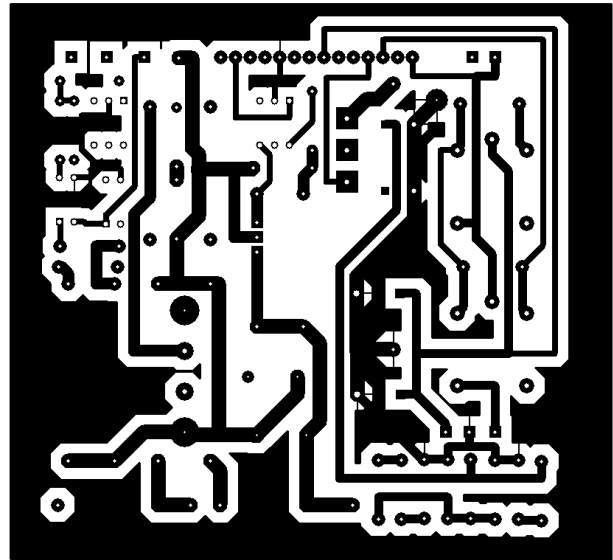
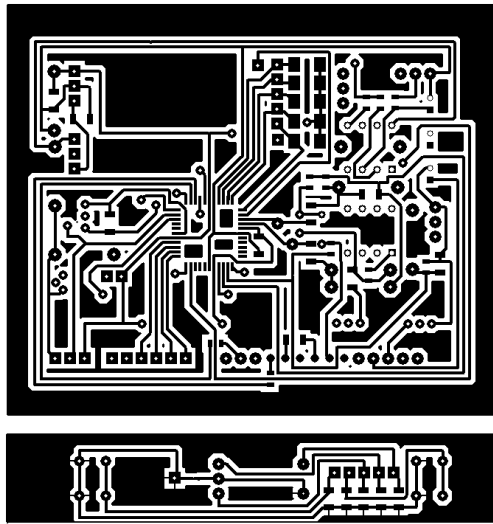


Рисунок 6.2 – друковані плати пристрою

3М.	Лист	№ докум.	Підпис	Дата
-----	------	----------	--------	------

ФЗ 8.17100.04.370 ПЗ

Арк

47

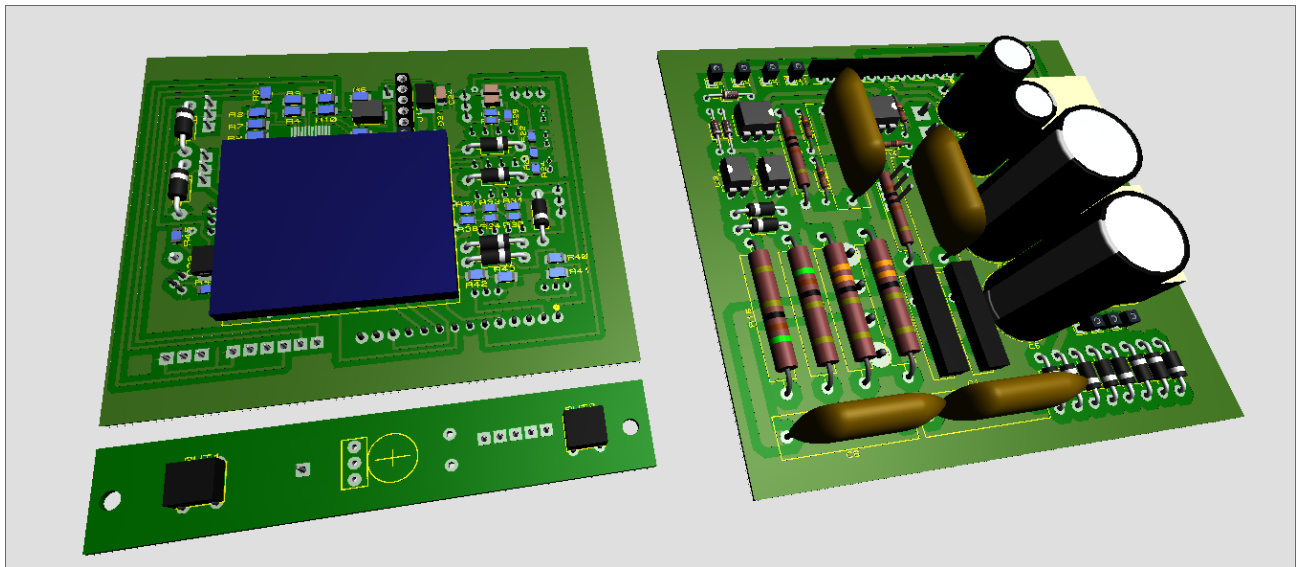
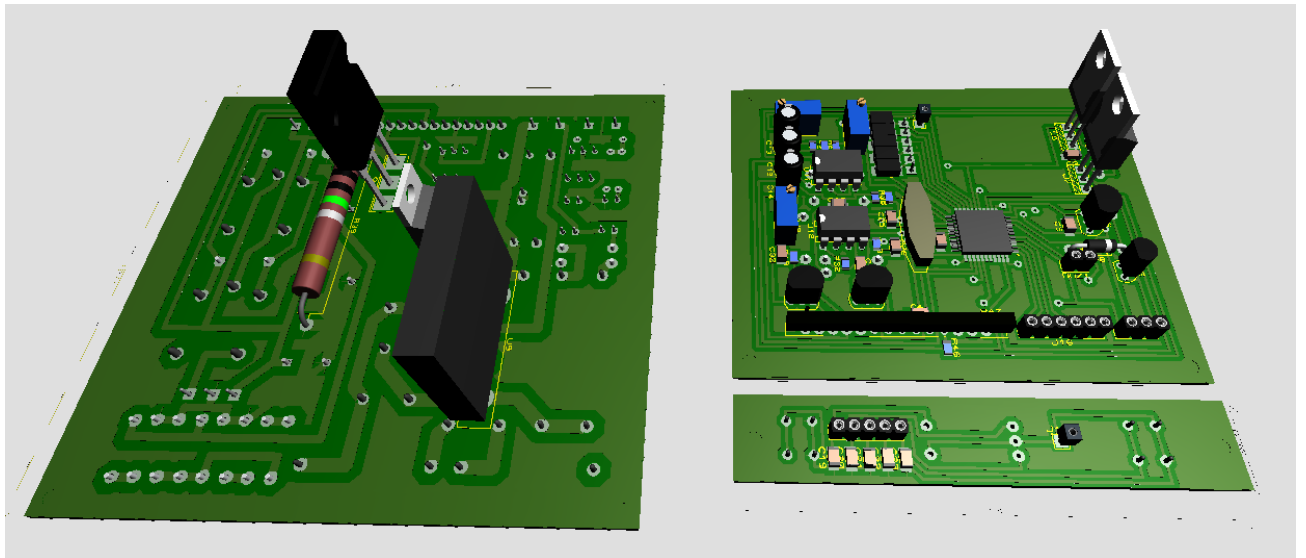


Рисунок 6.3 – 3Д модель плат пристрою

ЗМ.	Лист	№ докум.	Підпис	Дата
-----	------	----------	--------	------

Виготовлення плат одного екземпляру спроектованого пристрою Рис.6.4 виконаємо наступним чином:

- Друк на папері за допомогою лазерного принтеру;
- Перенесення малюнку плати на текстоліт термічним відтиском;
- Свердлення необхідних отворів;
- Травлення незахищеної області плати в хімічному розчині;
- Залудження провідників сплавом Розе;
- Монтаж електричних компонентів на плату;
- Програмування мікроконтролера.

					ФЗ 8.17100.04.370 ПЗ	Арк
ЗМ.	Лист	№ докум.	Підпис	Дата		49

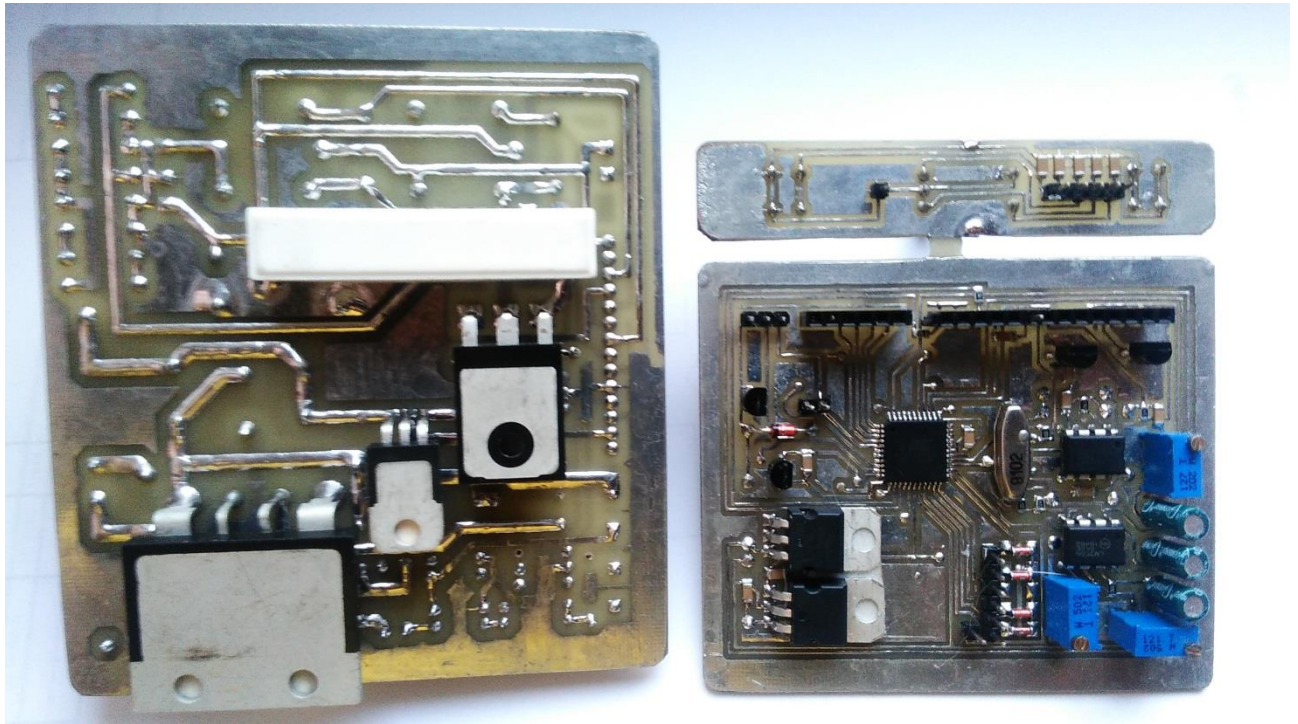
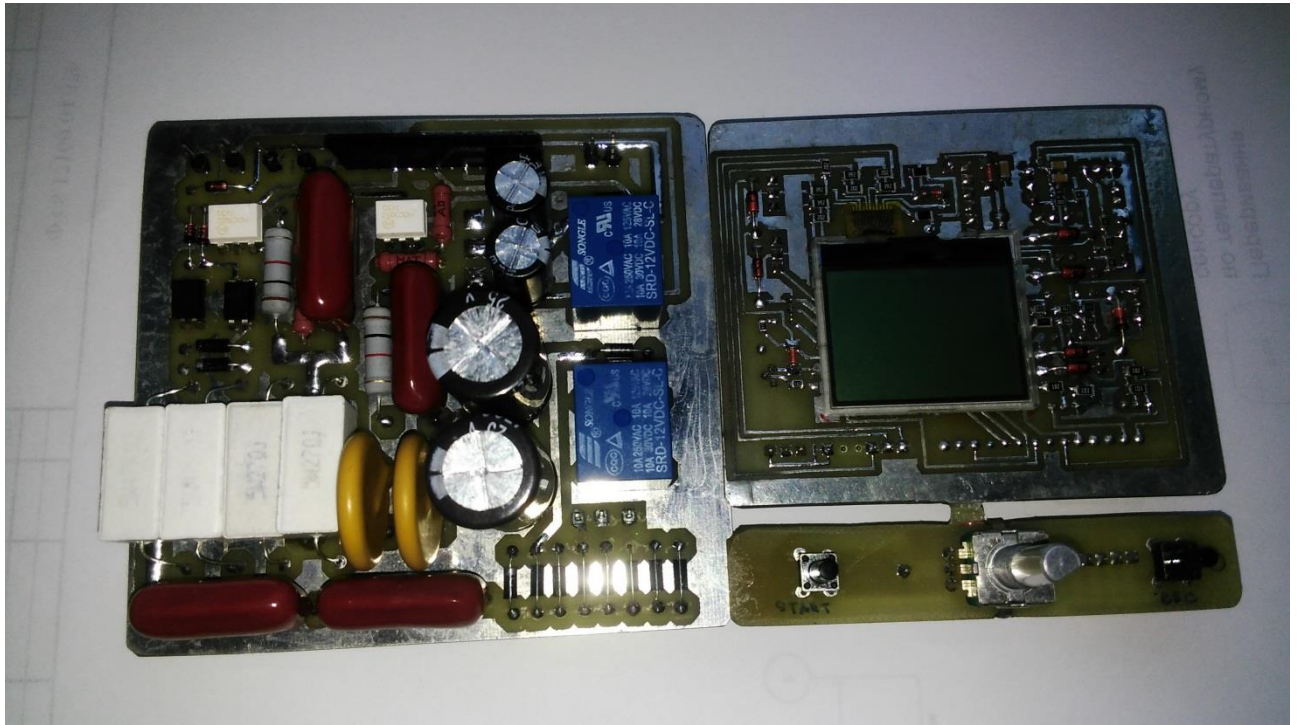


Рисунок 6.4 - фото плат пристрою

ЗМ.	Лист	№ докум.	Підпис	Дата

ФЗ 8.17100.04.370 ПЗ

Арк

50

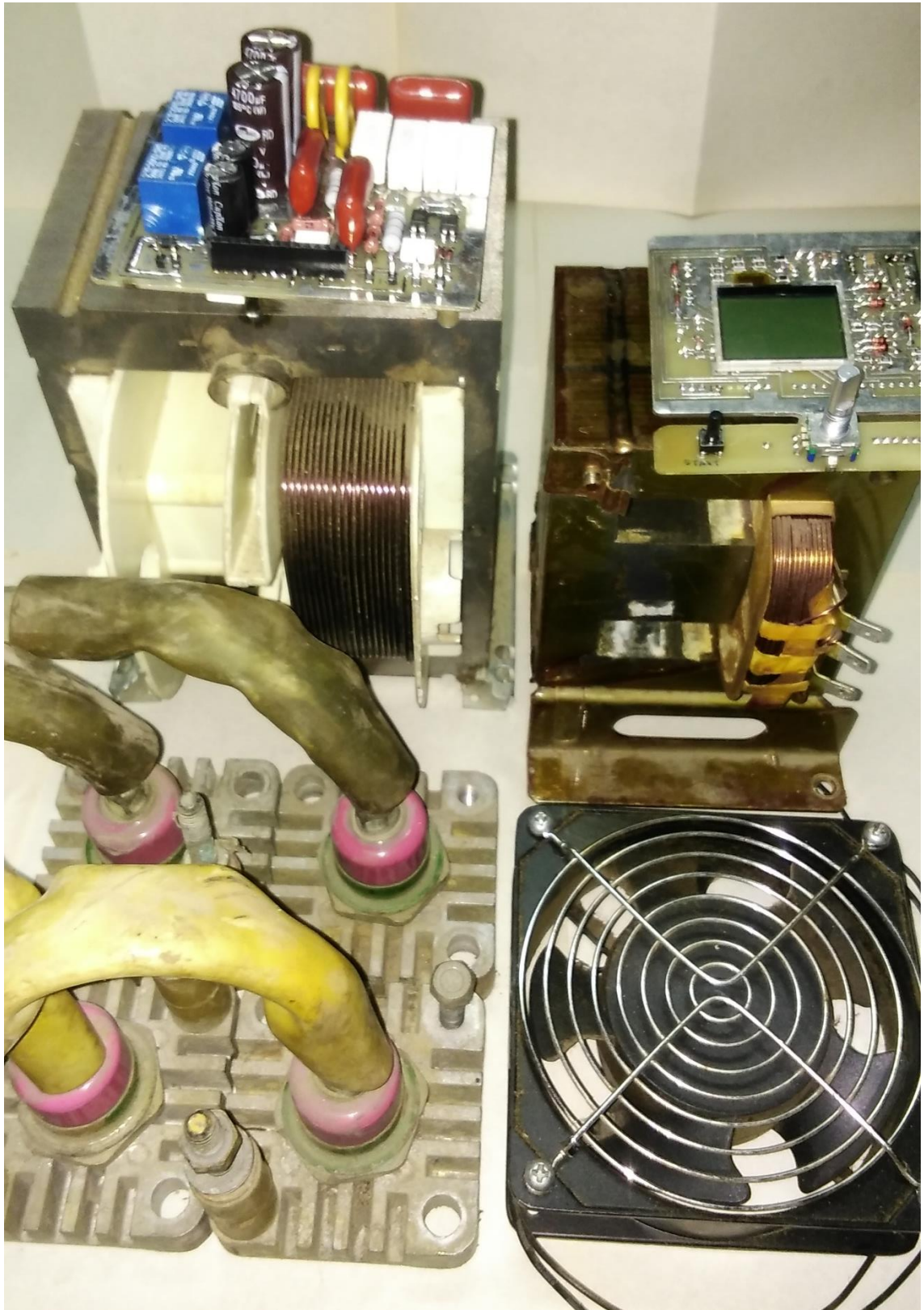


Рисунок 6.5 – фото основних складових пристрою

ЗМ.	Лист	№ докум.	Підпис	Дата

ФЗ 8.17100.04.370 ПЗ

Арк

51

7 НАУКОВО-ДОСЛІДНА ЧАСТИНА

Перед першим вмиканням пристрою перевіримо основні лінії курування та лінії живлення на нетипові значення опору та коротке замкнення, перевіримо якість та правильність монтажу електронних компонентів.

За допомогою зовнішнього лабораторного блоку живлення виставимо обмеження вихідного струму до 0,1А та проведемо перше ввімкнення керуючого блоку. За результатами виміру основних рівнів напруг на стабілізаторах та контрольних точках пристрою, отримані наступні значення:

Значення спрямлених напруг з Tr3:

- $U_1 = 16,3\text{В}$, після стабілізації $U_1 = 11,9\text{В}$,
- $U_2 = - 6,8\text{В}$, після стабілізації $U_2 = - 5.1\text{В}$,
- $U_3 = 16,3\text{В}$, після стабілізації $U_3 = 5.2\text{В}$,
- живлення дисплея: $U_{R6} = 3.2\text{В}$,

Після ввімкнення та візуальної оцінки, більш вдало сформуємо вигляд та позиції змінних та деяких пунктів меню пристрою Рис 7.1.

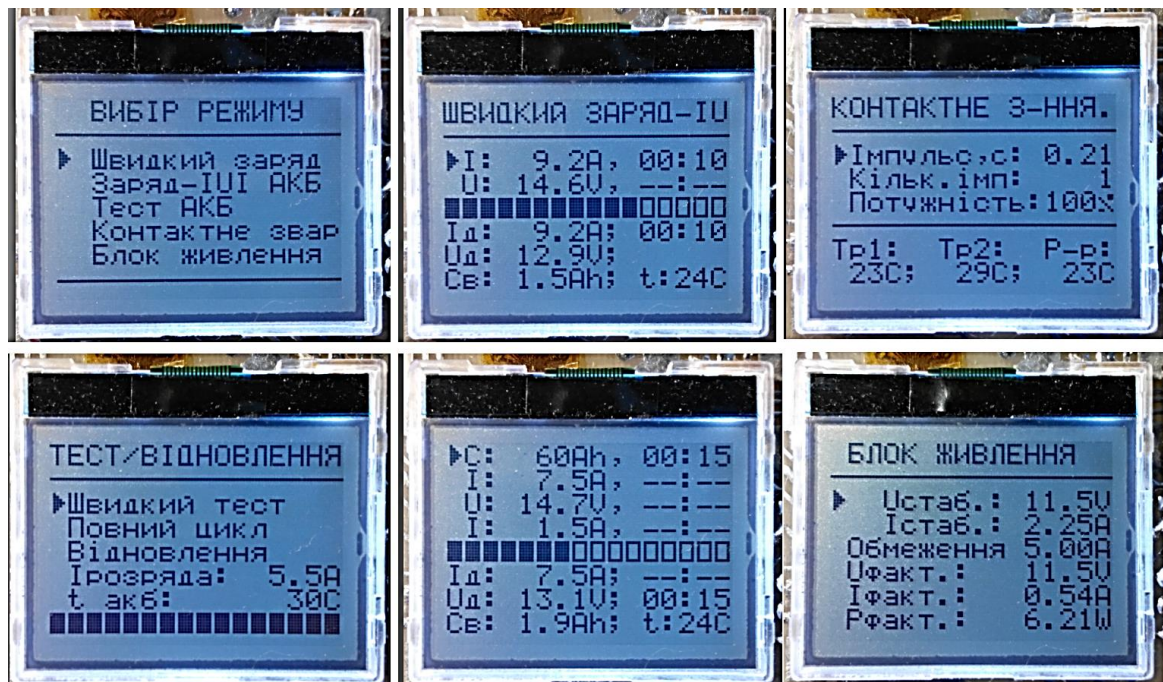


Рисунок 7.1 – графічне меню пристрою

Комфортну інтенсивність підсвічування дисплея налаштуємо підбором номіналу опору $R_6 = 330 \text{ Ом}$.

Обробка алгоритмів роботи описується наступними функціями (Додаток А):

- charge(),
- fast_charge(),
- test(),
- weld(),
- power_supply(),

Із функціями виводу зображення та графіки можна ознайомитись у Додатках Б та В:

- nlcd_Init();
- nlcd_Clear();
- nlcd_SendByte(char mode,unsigned char c);
- nlcd_Putc(unsigned char c);
- nlcd_Print(char * message, char x, char y);
- nlcd_Printf(unsigned flash char * message, char x, char y);
- nlcd_GotoXY(char x,char y);
- nlcd_GotoXY_pix(char x,char y);
- nlcd_Inverse(unsigned char mode);

Для подальшого налаштування переконаємось у правильності розрахунку та формування часових проміжків керуючого сигналу ШІМ (PORTB.3), що відповідає за стабілізацію значення встановленого струму через Q1, Рис 7.2.

Проаналізувавши осцилограму, можна зробити висновок, що задана частота та вихідний рівень відповідає дійсному, а також ширина імпульсу правильно співвідноситься до відображеного значення на дисплеї.

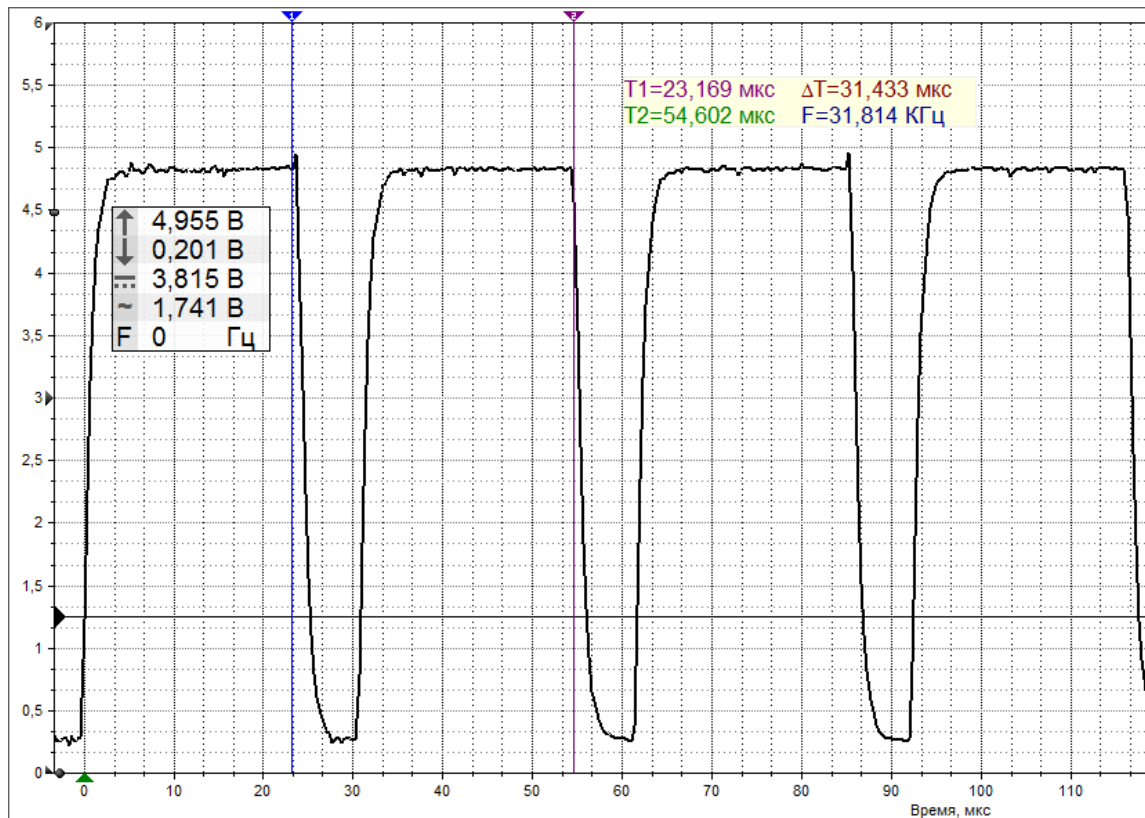


Рисунок 7.2 – ШІМ сигнал стабілізації вихідного струму пристрою

Перевіримо правильність розрахунку та роботу інтегруючих пар R21, C12, R23, C13, що перетворюють ШІМ сигнал в постійну напругу заданого рівня на вхід ОП, Рис 7.3.

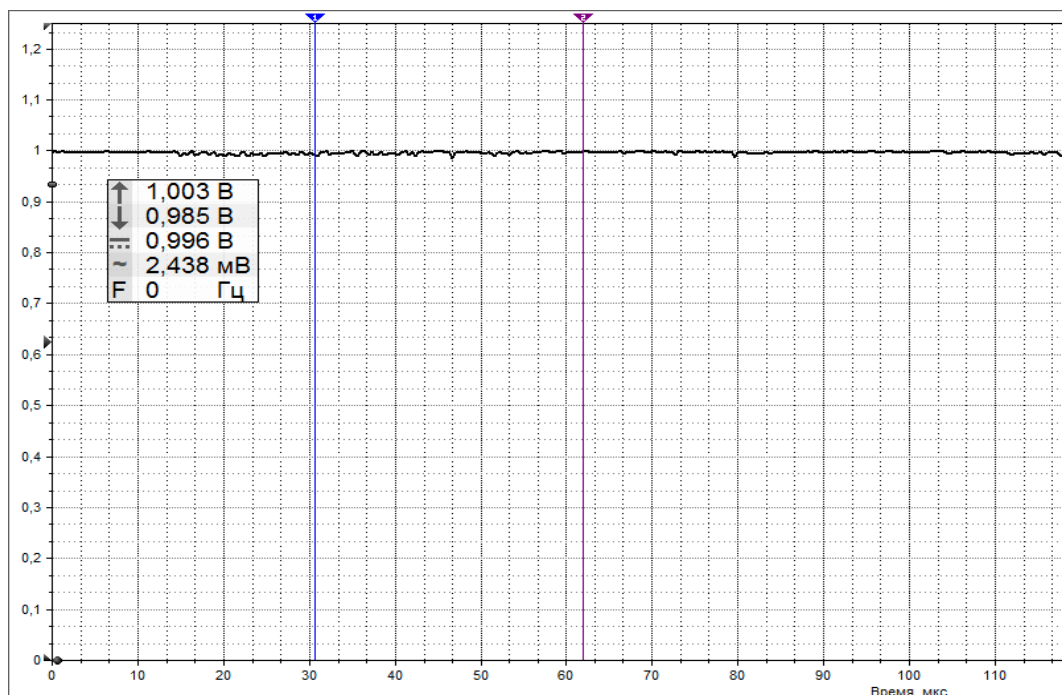


Рисунок 7.3 – калібрування вхідного рівня керуючої напруги ОП

Для правильного відображення та стабілізації струму, за допомогою багатообертового резистора RV1 встановимо рівень вхідної напруги ОП 1,0В Рис 7.3, що відповідає встановленому значенню струму на дисплеї 12,8А, значенню регістра OCR0=0xFF та фактичному спаду напруги на вимірювальному резисторі R39. Також перевіримо лінійність керування напругою та відповідність нульового значення.

На рис 7.3 можна визначити, що двохступеневе RC інтегрування ШІМ є досить ефективним та точним цифро-аналоговим методом перетворення, так як змінна складова сигналу не перевищує 2.5мВ.

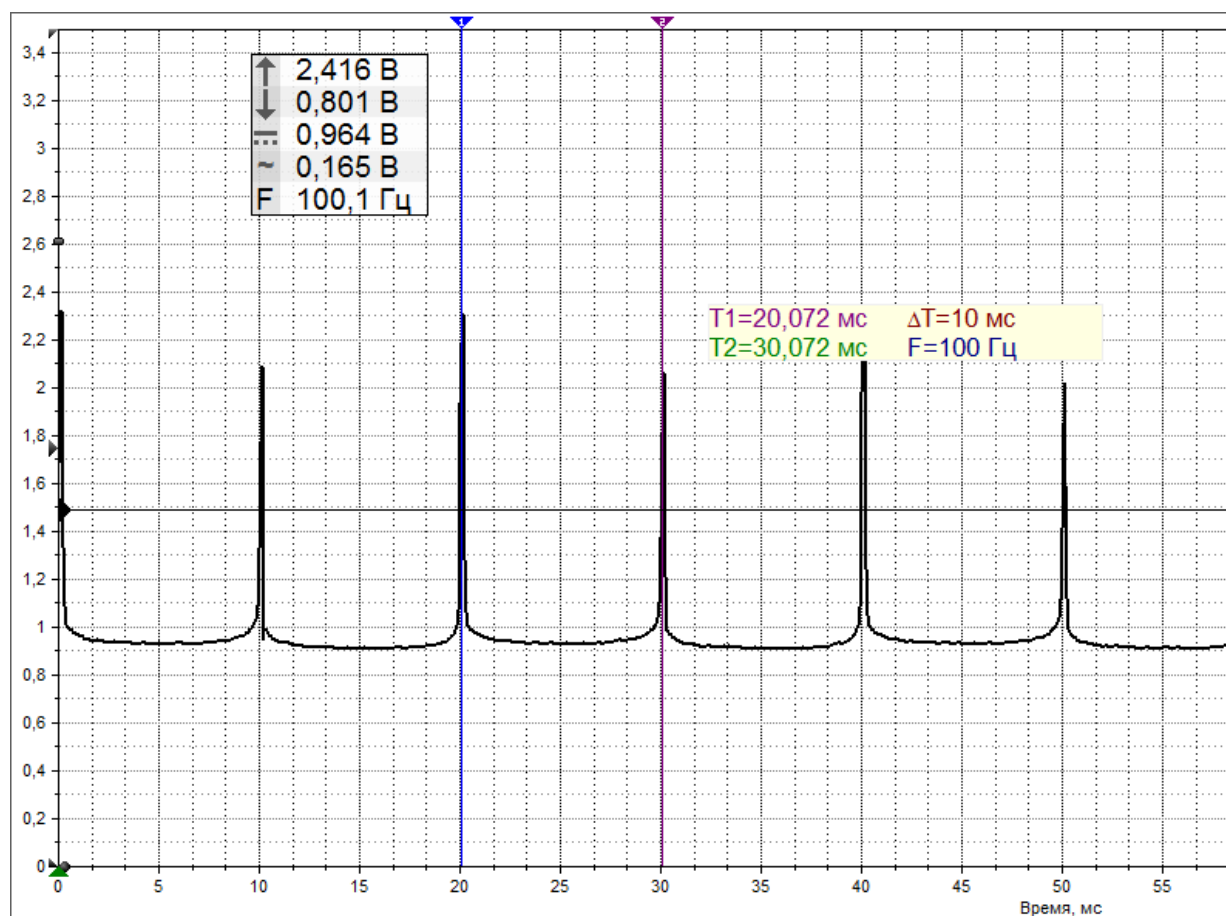


Рисунок 7.4 - осцилограма сигналу з оптопар синхронізації U3, U4

На Рис 7.4 зображено осцилограму сигналу з оптопар синхронізації з мережею 220В. Можна зробити висновок, що даний рівень не відповідає нормальним значенням для входу за зовнішнім перериванням INT1 МК [1]. Нижній рівень близько 1В та верхній 2.1-2.3В не забезпечать однозначного

переходу по вектору переривання МК, тому розділюючі діоди 1n4148 D3, D5 заміюємо на діоди Шоткі з низьким спадам напруги на них, а D6 вилучаємо.

Так як оптопари працюють послідовно з перекриттям, видно що сигнал переходу через нуль є занадто короткочасним, тому збільшимо номінали резисторів R15 та R16, знизивши струм світлодіода оптопар ми розширимо фазу перекриття їх вихідних сигналів і підвищимо значення напруги у піку керуючого імпульсу. Експериментальним шляхом встановлено, що оптимальним значенням R15 та R16 є 220к, Рис 7.5.

Продовжимо налаштування роботи пристрою перевіривши правильність роботи функції контактного зварювання, а саме фазу, послідовність та час формування імпульсів керування симістором U9.

На Рис 7.5 зображено формування групи імпульсів керування симістором при наступних заданих параметрах:

- час загального зварювального імпульсу – 0,1 с;
- кількість повторів – без повтору;
- потужність – 50%.

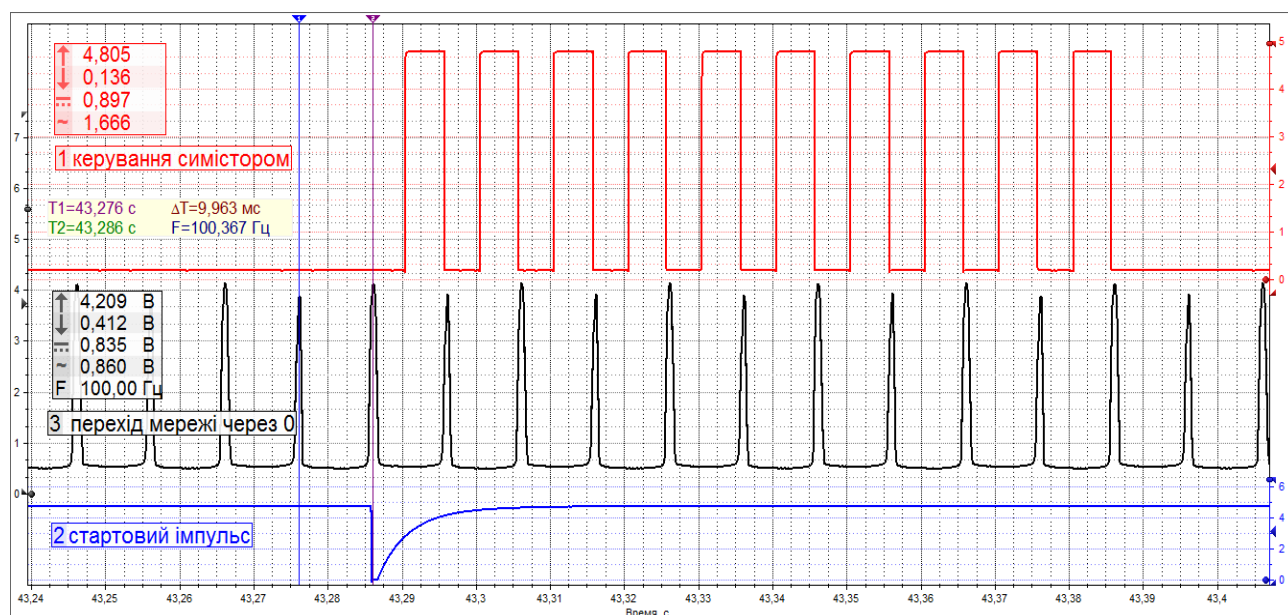


Рисунок 7.5 - група керуючих імпульсів

Проаналізувавши осцилограму переконуємось, що вихідний сигнал відповідає встановленим параметрам по фазі та тривалості.

Для надійного відкриття симістора високий рівень утримується майже до кінця діючого напівперіода мережі, а для закриття – необхідно щоб у момент переходу мережі через нуль на керуючому електроді був низький рівень

Рис.7.6, тому організовано незначне випередження по спаду сигнала у 0,5мс.

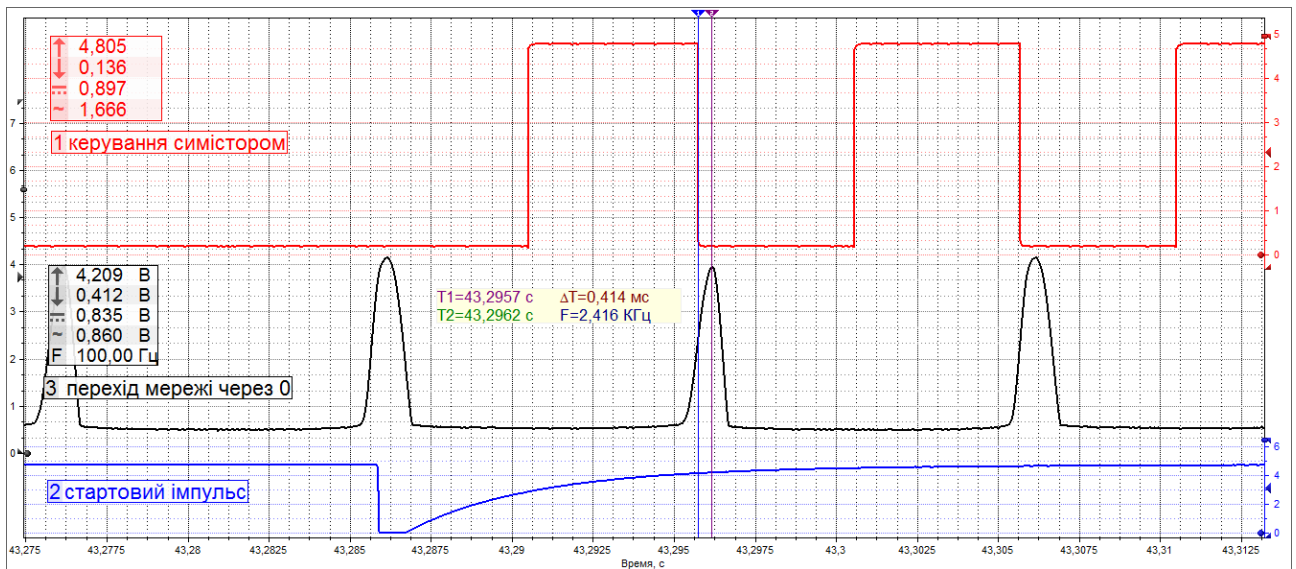


Рисунок 7.6 – сигнал керування симістором

Також перевіримо сигнал звукового супроводу при натисканні кнопок керування пристроєм Рис 7.7

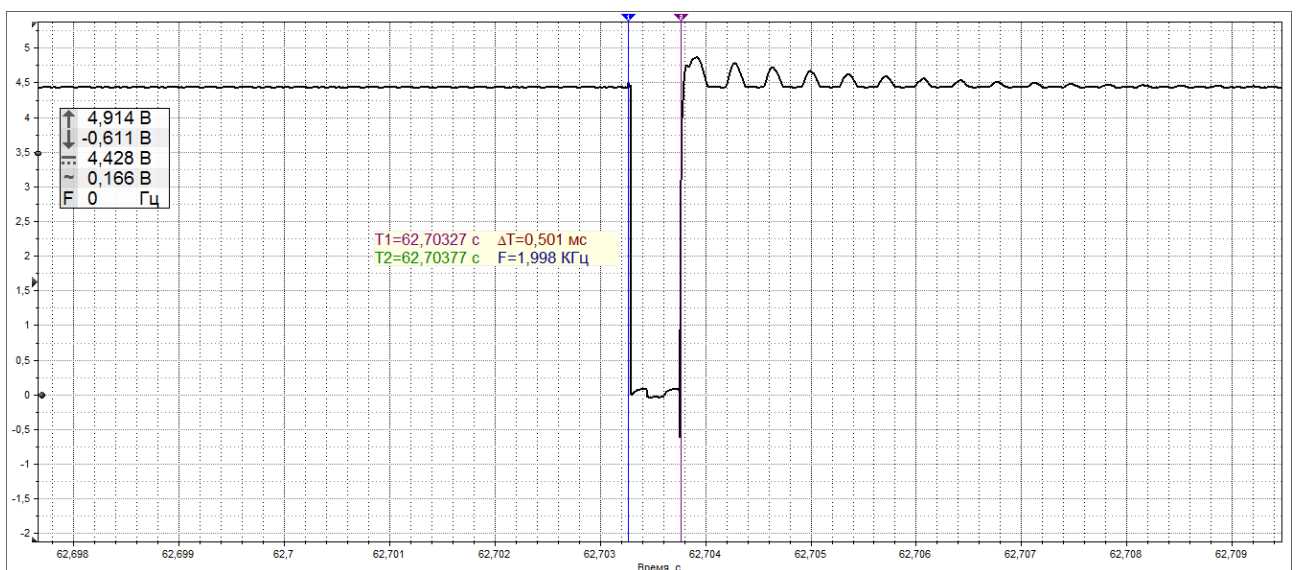


Рисунок 7.7 – сигнал на п`єзокерамічному елементі

За осцилограмою Рис 7.7 можна зробити висновок, що п`єзокерамічний елемент для звукового сповіщення є досить якісним.

За рахунок особливостей п`єзокераміки перетворювати напругу у механічні коливання та навпаки, коливання у напругу, можна бачити верхні періоди затухаючих коливань мембрани, які мають правильну форму і підтверджують нормальну роботу. Нижні напівперіоди обмежуються паралельно підключеним діодом D32.

					ФЗ 8.17100.04.370 ПЗ	Арк
ЗМ.	Лист	№ докум.	Підпис	Дата		58

8 ЕКОНОМІЧНА ЧАСТИНА

8.1 Розрахунок повної собівартості системи

Витрати на матеріали та комплектуючі вироби визначаються виходячи з ціни за одиницю матеріалу/комплектуючого та їх необхідної кількості. Дані про ціни на матеріали та комплектуючі взято з прайс – листів роздрібних постачальників матеріалів, сировини, комплектуючих, послуг в розрахунку на 1 одиницю випуску.

Результати розрахунків за цією статтею приведено в таблицях 8.1 та 8.2.

Таблиця 8.1 – Розрахунок витрат на комплектуючі

№ з/п	Найменування комплектуючого	Кільк. шт	Ціна за од., грн	Вартість, грн
Конденсатори				
	0.01uF	4	0.20	0.80
	4700uF-25V	2	17.00	34.00
	0.1uF	11	0.20	2.20
	10uF-25V	3	1.80	5.40
	0.33uF	6	0.20	1.20
	15pF	2	0.20	0.40
	1000uF-25V	2	12.00	24.00
Резистори				
	250	2	0.20	0.40
	2K	4	0.20	0.80
	3.9K	4	0.20	0.80
	33R (3W)	2	1.50	3.00
	300R (0.6W)	4	0.55	2.20
	1k (2W)	2	1.50	3.00
	51K (3W)	2	4.40	8.80

ЗМ.	Лист	№ докум.	Підпис	Дата
-----	------	----------	--------	------

ФЗ 8.17100.04.370 ПЗ

Арк

59

	300	2	0.20	0.40
	10k	5	0.20	1.00
	20k	3	0.20	0.60
	1k	5	0.20	1.00
	3k	2	0.20	0.40
	4.7k	4	0.20	0.80
	39k	1	0.20	0.20
	0.05 (10W)	1	7.00	7.00
	1K	2	0.20	0.40
	10K	2	0.20	0.40
	360	2	0.20	0.40
Мікросхеми				
	BTA20-800	1	24.00	24.00
	МОС3052	2	15.70	31.40
	PC817	2	3.50	7.00
	7805	1	6.50	6.50
	79L05	1	4.50	4.50
	7812	1	7.00	7.00
	ATMEGA32	1	65.00	65.00
	BTA80-800	1	115.00	115.00
	LM358N	2	8.00	16.00
	DS18B20	4	12.00	48.00
Транзистори				
	IRFP260	1	55.00	55.00
	2N3904	2	1.00	2.00
	BC548	1	1.00	1.00
Діоди				
	1N4007	2	0.60	1.20
	1n4148	3	0.40	1.20
	1N4728A	1	1.00	1.00

					ФЗ 8.17100.04.370 ПЗ	Арк
ЗМ.	Лист	№ докум.	Підпис	Дата		60

	1N4747A	1	1.50	1.50
	1N4734A	6	1.00	6.00
	1N4744A	1	1.00	1.00
	1N4004	10	1.00	10.00
	B-200	4	100.00	400.00
	1n4148	7	0.40	2.80
Дисплей				
	LCD_nokia1280	1	65.00	65.00
Інше				
	32768Hz	1	5.00	5.00
	RLY-NTE-46	2	35.00	70.00
	Button_dil	3	6.00	18.00
	encoder	1	45.00	45.00
	buzzer	1	15.00	15.00
	100uH	3	0.80	2.40
	5K	2	4.50	9.00
	2K	1	4.50	4.50
	S20K420	2	3.50	7.00
	8kVA	1	950.00	950.00
	1.2kVA	1	270.00	270.00
Всього(К):				2367.60

Таблиця 8.2 – Розрахунок витрат на сировину та матеріали

Матеріал, сировина	Одиниця виміру	Норма витрати	Ціна за одиницю, грн	Вартість, грн
Провід монтажний 0.5мм ²	кг	0.3	100	30
Провід обмоточний 100мм ²	кг	2	260	520
Склотекстоліт	м ²	0,02	600	12
Флюс	кг	0,01	150	1.5
Припій	кг	0,05	400	20
Лак	кг	0,2	50	10
Сумарні витрати (М)				593,5

З урахуванням транспортно-заготівельних витрат ($k_{m-3}=5\div 15\%$) вартість комплектуючих та матеріалів складе:

$$KM=(K+M)\cdot(100+k_{T-3})/100$$

$$KM=(2367,6+593,5)\cdot(100+10)/100=3257,2(\text{грн})$$

Витрати на основну заробітну плату:

$$Zo = \sum_{i=1}^n Tz_i \cdot Hч_i ,$$

де Tz_i – годинна тарифна ставка окремого спеціаліста (інженера-електронщика, лаборанта тощо), що задіяний у виробництві пристрою (установки), грн/год;

$Hч_i$ – витрачений час робітником на виробництво і наладку пристрою (установки);

					ФЗ 8.17100.04.370 ПЗ	Арк
ЗМ.	Лист	№ докум.	Підпис	Дата		62

n – кількість працівників, задіяних у виробництві пристрою (установки).

Годинна тарифна ставка розраховується, виходячи з величини місячного окладу спеціаліста:

$$T_{\text{гi}} = \frac{T_{\text{мi}}}{V\phi_i \cdot 8},$$

$T_{\text{мi}}$ – місячний оклад (ставка) спеціаліста, грн;

$V\phi_i$ – фактично відпрацьований час за розрахунковий період (місяць), днів (змін).

$$T_{\text{мi}}=7000 \text{ грн}; V\phi_i=24 \text{ дні}$$

$$T_{\text{гi}} = \frac{7000}{24 \cdot 8} = 36,45(\text{грн})$$

$H\text{ч}_i=30$ годин; $n=1$.

$$З_о = 30 \cdot 36,45 = 1093,5(\text{грн})$$

Додаткова заробітна плата (10÷30% від $З_о$):

$$З_д = З_о \cdot \frac{K_d}{100}$$

де K_d – відсоток додаткової заробітної плати.

Прийmemo $K_d = 10\%$

$$З_д = 1093,5 \cdot \frac{10}{100} = 109,3(\text{грн})$$

Відрахування на соціальні заходи містять відрахування від суми основної і додаткової зарплати за встановленими ставками:

- на обов'язкове державне пенсійне страхування;
- на державне страхування від нещасних випадків;

- на обов'язкове державне соціальне страхування на випадок безробіття;
- у зв'язку з тимчасовою втратою працездатності і витратами, зумовленими народженням дитини і похованням

$$B_{cz} = (Z_o + Z_d) \cdot \frac{36.3}{100} = (1093,5 + 109,3) \cdot 0,363 = 436,6(\text{грн})$$

Витрати на утримання і експлуатацію устаткування:

У разі, якщо устаткування перебуває на балансі підприємства витрати на утримання і експлуатацію устаткування (ВУЕУ) = основна зарплата * %ВУЕУ, (приймаємо %ВУЕУ=120÷150%).

$$B_{UEY} = Z_o * 1,5 = 1093,5 * 1,5 = 1640,25(\text{грн})$$

Загальновиробничі витрати являють собою витрати, пов'язані з управлінням підрозділом, витрати на службові відрядження співробітників підрозділу (цеху), амортизаційні відрахування від вартості основних фондів загально цехового призначення і т.д.

Визначаються в розмірі 130÷250% від основної зарплати. Прийmemo загальновиробничі витрати в розмірі 130% від основної зарплати :

$$B_{zv} = Z_o * 1,5 = 1093,5 * 1,5 = 1640,25(\text{грн})$$

Сума статей:

1. Матеріали та комплектуючі.
2. Основна заробітна плата.
3. Додаткова заробітна плата.
4. Відрахування на соціальні заходи.
5. Витрати на утримання і експлуатацію устаткування.
6. Загальновиробничі витрати.

становить виробничу собівартість системи і дорівнює:

					ФЗ 8.17100.04.370 ПЗ	Арк
						64
ЗМ.	Лист	№ докум.	Підпис	Дата		

$$\sum_{cc} = KM + Z_o + Z_d + B_{cz} + B_{UEY} + B_{zv} = 3257,2 + 1093,5 + 109,3 + 436,6 + \\ +1640,25 + 1640,25 = 8177.1 \text{ (грн)}$$

Адміністративні витрати можуть містити в собі:

- витрати, пов'язані з управлінням підприємства;
- витрати на службові відрядження адміністрації підприємства;
- витрати на пожежну й сторожову охорону;
- витрати, пов'язані з підготовкою (навчанням) і перепідготовкою кадрів;
- витрати на перевезення працівників до місця роботи і назад;
- витрати на сплату відсотків за фінансові кредити, а також відсотків за товарні і комерційні кредити; витрати, пов'язані зі сплатою відсотків за користування матеріальними цінностями, взятими в оренду (лізинг);
- витрати, пов'язані з оплатою послуг комерційних банків і інших кредитно-фінансових установ;
- податки, відрахування.

Визначаються в розмірі 140-200% від основної зарплати. Прийmemo за 150% , що дорівнює:

$$B_a = Z_o * 1,5 = 1093,5 * 1,5 = 1640,25 \text{ (грн)}$$

Витрати на збут включають витрати на рекламу та передреалізаційну підготовку системи. Орієнтовно ці витрати визначаються в розмірі 5-10% від виробничої собівартості і складуть приблизно 408 грн.

Отже повна собівартість системи становить:

$$\sum_{pc} = \sum_{cc} + B_a + B_p = 8177,1 + 1640,25 + 408 = 10225,35 \text{ (грн)}$$

					ФЗ 8.17100.04.370 ПЗ	Арк
ЗМ.	Лист	№ докум.	Підпис	Дата		65

Таблиця 8.3 – Калькуляція собівартості системи

Найменування статей калькуляції	Проектний варіант
1. Матеріали та комплектуючі.	3257,2
2. Основна заробітна плата.	1093,5
3. Додаткова заробітна плата.	109,3
4. Відрахування на соціальні заходи.	436,6
5. Витрати на утримання і експлуатацію устаткування.	1640,25
6. Загальновиробничі витрати.	1640,25
Виробнича собівартість	8177,1
7. Адміністративні витрати.	1640,25
8. Витрати на збут.	408
Повна собівартість системи	10225,35

8.2 Визначення ціни системи

У ринковій економіці існують різні методи ціноутворення: с/в плюс прибуток, забезпечення фіксованого обсягу прибутку, залежно від рівня попиту та ін. Розрахунок оптової ціни системи проведемо за схемою «собівартість плюс прибуток».

$$C_{\text{опт}} = C + \frac{R \cdot C}{100} = 10225,35 + \frac{35 \cdot 10225,35}{100} = 13804,2(\text{грн})$$

де R – рентабельність системи, приймається в розмірі до 35% від його собівартості.

Відпускна ціна системи включає податок на додану вартість:

$$C_{\text{розд}} = C_{\text{онт}} \cdot 1,2 = 16565(\text{грн}).$$

де 20% – ПДВ.

Позитивні сторони даної методики полягають у її простоті, комплексній очевидності такої функції ціни як відшкодування витрат на виробництво і забезпечення прибутковості від створення та реалізації системи. Недолік даної методики полягає в тому, що вона не враховує ринкові фактори ціноутворення і насамперед попит. Однак в умовах ринкової економіки існують ситуації, коли підприємствам доцільно її застосовувати: в умовах відсутності конкуренції (монополії), при обмеженні рентабельності продукції з боку держави, виконанні одноразових замовлень, виготовленні оригінальної продукції.

Необхідно відзначити, що для встановлення реальної ціни яка б відповідала умовам існуючого ринку систем, необхідні відповідні маркетингові дослідження.

8.3 Визначення річних витрат користувача

Річні експлуатаційні витрати V_{pe} включають витрати на електроенергію (живлення пристрою), заробітну плату обслуговуючого персоналу, амортизаційні відрахування, витрати на поточний ремонт та інші витрати.

Витрати на електроенергію (живлення пристрою) визначаються за формулою:

$$V_e = W \cdot C_e \cdot t$$

де W – потужність пристрою, кВт;

C – вартість одного кВт·год, грн/кВт·год (тариф для промислових користувачів);

					ФЗ 8.17100.04.370 ПЗ	Арк
ЗМ.	Лист	№ докум.	Підпис	Дата		67

t – час роботи пристрою у році, год.

$W = 6,5$ кВт; $C=2,65$ грн/кВт·год; $t=200$ год.

$$Ve=6,5 \cdot 2,65 \cdot 200=3445 \text{ (грн)}$$

Річна сума амортизаційних відрахувань визначається за відповідними нормами амортизації від первісної вартості пристрою, що включає: ціну пристрою; витрати на транспортування й монтаж, які приймаються в розмірі 5–15% від ціни.

Амортизацію електронних пристроїв (установок) доцільно нараховувати за прямолінійним методом:

$$A = F \cdot a$$

де F – первісна вартість пристрою;

a – річна норма амортизації.

$$F = C_{розд} + B_{тм} = 16565 + 1657 = 18222 \text{ (грн)}$$

де $C_{розд}$ – роздрібна (договірна) ціна пристрою;

$B_{тм}$ – витрати на транспортування й монтаж пристрою (10%).

Річна норма амортизації обчислюється прямолінійним методом відповідно до строку корисного використання пристрою:

$$a = 1/T_{вик} = 0,2$$

де $T_{вик}$ – строк корисного використання пристрою, років. Відповідно до нового Податкового кодексу України допускається не менше 2 років. $T_{вик} = 5$ років.

Отже амортизація

$$A = 0,2 \cdot 18222 = 3644,4 \text{ (грн)}$$

					ФЗ 8.17100.04.370 ПЗ	Арк
ЗМ.	Лист	№ докум.	Підпис	Дата		68

Витрати на поточний ремонт та інші витрати включають в себе вартість електронних елементів, які виходять зі строю протягом року та вартість демонтажних та монтажних робіт. Витрати на ремонт приймаються рівними 5% від капітальних витрат з урахуванням прогнозування відмов:

$$B_{np} = F \cdot k_{np} / 100 = 693.1 \text{ (грн)}$$

де k_{np} – відсоток витрат на поточний ремонт.

Таблиця 8.4 – Річні експлуатаційні витрати користувача

Статті експлуатаційних витрат	грн.
Вартість електроенергії	3445
Амортизаційні відрахування	3644.4
Витрати на поточний ремонт	693.1
Всього (B_{pe})	7782.5

8.4 Розрахунок повної ціни користувача

Ціна користувача $C_{кор}$ включає в себе капітальні вкладення F та сумарні річні витрати на експлуатацію пристроїв B_{pe} протягом строку використання пристрою:

$$C_{кор} = F + B_{pe} \cdot T_{вик} = 18222 + 7782.5 \cdot 5 = 57134.5 \text{ грн.}$$

8.5 Оцінка конкурентоспроможності

Таблиця 8.5 – Порівняння техніко-економічних показників електронної системи-аналога та проектної електронної системи

Показник	Одиниця вимірювання	проектний пристрій (система)
Потужність пристрою	Вт	6500
Середній наробіток на відмову	год	5000
Ймовірність безвідмовної роботи	%	80
Маса	кг	35
Виробнича собівартість	грн	8177,1
Повна собівартість	грн	10225,35
Відпускна ціна	грн	16565
Річні експлуатаційні витрати користувача	грн	7782,5
Повна ціна користувача	грн	57134,5

Провівши розрахунки і порівнявши їх результати з аналогічними системами можна зробити висновок про техніко-економічну доцільність розроблення і впровадження у виробництво пристрою з урахуванням розрахованих параметрів. Повну собівартість можна знизити приблизно вдвічі, якщо вилучити пускову функцію з пристрою, а також організувати постачання електронних компонентів за оптовими цінами, або цінами виробника.

ВИСНОВОК

В даному проекті була розроблена та виготовлена схема пристрою на базі мікроконтролера сімейства ATmega, що поєднує в собі функції контактного зварювання металів та заряду кислотних автомобільних акумуляторних батарей.

В процесі проектування було виконано:

- ознайомлення з основними принципами та варіантами щодо формування подібних схем;
- розроблено структурну та функціональну схеми майбутнього виробу;
- на основі функціональної схеми проведено делальний розрахунок та опис складових загальної принципової схеми;
- сформовано загальну принципову схему, за якою було виконано розміщення елементів, розроблено трафарети друкованих плат та створено їх тривимірну модель;
- уточнено та остаточно сформовано алгоритм роботи та взаємодії складових пристрою;
- на основі алгоритму написано основні блоки програмного коду керуючої програми;
- було виготовлено в одному екземплярі та налагоджено роботу схемотехніки пристрою.

В процесі роботи на проектом було отримано додаткові знання в області проектування пристроїв на базі мікроконтролерів, принципів роботи з графічним дисплеєм, фазовим методом керування потужністю актино-індуктивного навантаження та здобуто у персональне користування пристрій, що виконує необхідні функції.

					ФЗ 8.17100.04.370 ПЗ	Арк
Зм.	Лист	№ докум.	Підпис	Дата		71

СПИСОК ЛІТЕРАТУРИ

1. Atmega32A/L Datasheet, 2503Q–AVR–02/11/
2. PCF8814 65 × 96 pixels matrix LCD driver., Objective specification 2003 Mar 13., Philips Semiconductors. Datasheet.
3. CodeVisionAVR 3.26 User Manual, rev61/03.2016., HP InfoTech S.L.R
4. AVR042: AVR Hardware Design Considerations., - Atmel-2521M-AVR. APPLICATION NOTE 09/14.
5. AVR120: Characterization and Calibration of the ADC on an AVR., Rev. 2559D-AVR-02/06. APPLICATION NOTE.
6. AVR130: Setup and Use the AVR® Timers., Rev. 2505A–AVR–02/02. APPLICATION NOTE.
7. AVR134: Real Time Clock (RTC) using the Asynchronous Timer., Rev. 1259F-AVR-08/06. APPLICATION NOTE.
8. AVR182: Zero Cross Detector., Rev.2508B–AVR–01/04. APPLICATION NOTE.
9. Кравченко, А.В. 10 практических устройств на AVR-микроконтроллерах. Кн. 4 / А.В. Кравченко. – СПб., 2013. - 320 с.
- 10.Мортон, Д. Микроконтроллеры AVR. Вводный курс / Д. Мортон. - М.: ДМК, 2015. - 272 с.
11. Евстифеев, А.В. Микроконтроллеры AVR семейств Mega. Руководство пользователя / А.В. Евстифеев. - М.: ДМК, 2015. - 588 с.
12. Белов, А.В. Микроконтроллеры AVR: от азов программирования до создания практических устройств / А.В. Белов. - СПб., 2016. - 544 с.
13. Евстифеев А.В. “Микроконтроллеры AVR семейств Tiny и Mega фирмы ATMEL”, 5_е изд., стер. — М.: Издательский дом «Додэка_XXI», 2008. — 560 с. ISBN 978_5_94120_220_1
14. Трамперт В. «Измерение, управление и регулирование с помощью AVR микроконтроллеров» пер. с нем. – к.: МК Прес, 2006. 208с., -ил. ISBN 966-8806-14-X(рус).

					ФЗ 8.17100.04.370 ПЗ	Арк
3М.	Лист	№ докум.	Підпис	Дата		72

15. MOC3052 Random-phase optoisolators triac drivers,. Fairchild semiconductor. Datasheet.
16. PC817 Series High Density Mounting Type Photocoupler, SHARP., Datasheet.
17. DUAL DIFFERENTIAL INPUT OPERATIONAL AMPLIFIERS LM358, SEMICONDUCTOR TECHNICAL DATA, MOTOROLA. Datasheet.
18. DS18B20 Programmable Resolution 1-Wire Digital Thermometer Dallas Maxim Integrated. 19-7487; Rev 4; 1/15 Datasheet.
19. ГОСТ 10317-79 Платы печатные, основные размеры.

					ФЗ 8.17100.04.370 ПЗ	Арк
ЗМ.	Лист	№ докум.	Підпис	Дата		73

Додаток А – основний цикл та формування пунктів меню пристрою

File: spt.c, Date: 13/12/18, Time: 12:17:42

```
1  /*****
2  Project : SPT
3  Version : 1.0
4  Date   : 11/11/2018
5  Author : Gaidar Yaroslav
6  Company :
7  Comments:
8
9  Chip type      : ATmega32A
10 Program type   : Application
11 AVR Core Clock frequency: 8,000000 MHz
12 Memory model   : Small
13 External RAM size : 0
14 Data Stack size : 512
15 *****/
16
17 #include <mega32a.h>
18 #include <nokia_lcd.h>
19 #include <stdio.h>
20 #include <delay.h>
21
22
23
24 //keyboard var
25 #define push 3
26 #define ff 1
27 #define rev 2
28 #define start 4
29 #define esc 5
30 char keyboard_state; //распознавание комбинации кнопок
31 char double_ff; //ускоренная перемотка энкодером
32 unsigned int buztime; //пищалка
33
34
35 //pointer var
36 #define up 1
37 #define down 2
38 #define off 3
39 #define on 4
40
41 //t sensors var
42 char ds_1;
43 char ds_2;
44 char ds_3;
45 char ds_4;
46
47 //charge var
48 bit charging_process; //разрешение заряда
49 char progress_time; //линия прогресса зарядки
50 char progress_line; //линия прогресса зарядки
51
52 //full charge var
53 //bit pulse charge; //режим импульсного заряда
54 float I1fch_set;
55 float I2fch_set;
56 float Ufch_set=14.7;
57 float I1fch_fact;
58 float Ufch_fact;
59 char Cfch_fact;
60 char Cfch_set=55;
61 unsigned long time1; //время каждого этапа зарядки
62 unsigned long time2;
63 unsigned long time3;
64 unsigned long time4;
65 unsigned long time5;
66 unsigned long total_time; //общее время
67
68 //fast charge var
69 float Ifast_ch_set;
70 float Ufast_ch_set;
```

									ФЗ 8.17100.04.370 ПЗ	Арк
ЗМ.	Лист	№ докум.	Підпис	Дата						74

File: spt.c, Date: 13/12/18, Time: 12:17:42

```
71 float Ifast_ch_fact;
72 float Ufast_ch_fact;
73 float Qfast_ch_fact;
74
75 //weld var
76 bit welding; //флаг сварочного процесса
77 bit progressive; //нарастание длительности импульсов до установленного
78 bit manual; //ручной режим
79 char quantity=1; //количество последовательных импульсов
80 char welding_current=50; // % фазовое управление мощностью
81 float welding_time=0.1; //время сварочного импульса
82 unsigned int temp_time;
83
84 //pointers/menu var
85 char buf [5]; //для выв. переменных
86 char i_1;
87 char i_2;
88 char pi; //стартовая позиция указателя меню
89 char level_1; //текущее подменю
90 unsigned int blink_time=0x00FF; //мигание курсора
91 bit set_value; //разрешение изменения переменных
92 bit show_var_enable; //разрешение вывода переменных на дисплей
93 char change_var; //текущая изменяемая переменная
94 char show_var=0x05; //время обновления переменных на дисплее
95
96 char mode;
97 #define mode_fast_charge 1
98 #define mode_charge 2
99 #define mode_test 3
100 #define mode_weld 4
101 #define mode_power_supply 5
102
103
104 //главный экран
105 void main_screen(void)
106 {i_1=2; i_2=6; //строки указателя от и до
107 pi=2; nlcd_Printf(">",0,pi);
108 nlcd_Printf(" ВИВІР РЕЖИМУ ",0,0);
109 nlcd_Line (0,12,95,12);
110 nlcd_Printf("Швидкий заряд",2,2);
111 nlcd_Printf("Заряд-IUI АКБ",2,3);
112 nlcd_Printf("Тест АКБ",2,4);
113 nlcd_Printf("Контактне звар",2,5);
114 nlcd_Printf("Блок живлення",2,6);
115 nlcd_Line (0,61,95,61);
116 }
117
118 //функция движения курсора вверх
119 void pointer(char i_1, char i_2, char i3)
120 {char i;
121
122 switch(i3)
123 {show_var_enable=0;
124 case up:
125 for(i=i_1;i<=i_2;++i)
126 {nlcd_Printf(" ",0,i);}
127 if(--pi<i_1){pi=i_1;}
128 if(pi>7){pi=0;}
129 nlcd_Printf(">",0,pi); break;
130 case down:
131 for(i=i_1;i<=i_2;++i)
132 {nlcd_Printf(" ",0,i);}
133 if(++pi>i_2){pi=i_2;}
134 nlcd_Printf(">",0,pi); break;
135 case on: nlcd_Printf(">",0,pi); break;
136 case off:
137 for(i=i_1;i<=i_2;++i)
138 {nlcd_Printf(" ",0,i);} break;
139 }
140 if(mode!=0){show_var_enable=1;};
```

					ФЗ 8.17100.04.370 ПЗ	Арк
ЗМ.	Лист	№ докум.	Підпис	Дата		75

File: spt.c, Date: 13/12/18, Time: 12:17:42

```
141     }
142
143     //экран полного заряда АКБ
144     void charge(void)
145     {show_var_enable=0;
146      mode=mode_charge;
147      nlcd_Clear();
148      i_1=0; i_2=3;
149      pi=0; nlcd_Printf(">",0,pi);
150      nlcd_Printf("C: ---Ah, ---:",1,0);
151      nlcd_Printf("I: ----A, ---:",1,1);
152      nlcd_Printf("U: ----V, ---:",1,2);
153      nlcd_Printf("I: ----A, ---:",1,3);
154      nlcd_Printf("#####",0,4);
155      nlcd_Printf("Iд: ---A; ---:",0,5);
156      nlcd_Printf("Uд: ----V; ---:",0,6);
157      nlcd_Printf("Cв: ---Ah; t:--C",0,7);
158      show_var_enable=1;
159      while(level_1!=0)
160      {
161          if(keyboard_state==start)
162          {keyboard_state=0;
163           if(charging_process==0)
164           {charging_process=1;
165            pointer(i_1,i_2,off);
166            while(charging_process==1)
167            {
168                if(keyboard_state==start)
169                {keyboard_state=0;
170                 pointer(i_1,i_2,on);
171                 charging_process=0;
172                }
173            }
174          }
175          if(keyboard_state==rev)
176          {keyboard_state=0;
177           pointer(i_1,i_2,up);}
178          if(keyboard_state==ff)
179          {keyboard_state=0;
180           pointer(i_1,i_2,down);}
181          if(keyboard_state==esc)
182          {keyboard_state=0;
183           show_var_enable=0;
184           mode=0;
185           level_1=0;
186           nlcd_Clear();
187           main_screen();}
188          if(keyboard_state==push)
189          {keyboard_state=0;
190           set_value=1;
191           switch(change_var=pi)
192           {case 0: while(set_value==1)
193              {if(keyboard_state==rev)
194                 {keyboard_state=0;
195                  if(double_ff==0)
196                  {Cfch_set--;}
197                  else {Cfch_set=Cfch_set-10;}
198                  double_ff=0x0f;
199                  I1fch_set=(float)Cfch_set/8;
200                  I2fch_set=(float)Cfch_set/40;
201                  }
202              if(keyboard_state==ff)
203              {keyboard_state=0;
204               if(double_ff==0)
205               {Cfch_set++;}
206               else {Cfch_set=Cfch_set+10;}
207               double_ff=0x0f;
208               I1fch_set=(float)Cfch_set/8;
209               I2fch_set=(float)Cfch_set/40;
210              }
211           }
```

File: spt.c, Date: 13/12/18, Time: 12:17:42

```
212         if(keyboard_state==push)
213         {keyboard_state=0;
214           set_value=0;
215           pointer(i_1,i_2,on);}
216         } break;
217     case 1: while(set_value==1)
218     {if(keyboard_state==rev)
219     {keyboard_state=0;
220       if(double_ff==0)
221       {Ilfch_set=Ilfch_set-0.1;}
222       else {Ilfch_set=Ilfch_set-0.5;}
223       double_ff=0x0f;
224     }
225     if(keyboard_state==ff)
226     {keyboard_state=0;
227       if(double_ff==0)
228       {Ilfch_set=Ilfch_set+0.1;}
229       else {Ilfch_set=Ilfch_set+0.5;}
230       double_ff=0x0f;
231     }
232     if(keyboard_state==push)
233     {keyboard_state=0;
234       set_value=0;
235       pointer(i_1,i_2,on);}
236     } break;
237     case 2: while(set_value==1)
238     {if(keyboard_state==rev)
239     {keyboard_state=0;
240       if(double_ff==0)
241       {Ufch_set=Ufch_set-0.1;}
242       else {Ufch_set=Ufch_set-0.5;}
243       double_ff=0x0f;
244     }
245     if(keyboard_state==ff)
246     {keyboard_state=0;
247       if(double_ff==0)
248       {Ufch_set=Ufch_set+0.1;}
249       else {Ufch_set=Ufch_set+0.5;}
250       double_ff=0x0f;
251     }
252     if(keyboard_state==push)
253     {keyboard_state=0;
254       set_value=0;
255       pointer(i_1,i_2,on);}
256     } break;
257     case 3: while(set_value==1)
258     {if(keyboard_state==rev)
259     {keyboard_state=0;
260       if(double_ff==0)
261       {I2fch_set=I2fch_set-0.1;}
262       else {I2fch_set=I2fch_set-0.5;}
263       double_ff=0x0f;
264     }
265     if(keyboard_state==ff)
266     {keyboard_state=0;
267       if(double_ff==0)
268       {I2fch_set=I2fch_set+0.1;}
269       else {I2fch_set=I2fch_set+0.5;}
270       double_ff=0x0f;
271     }
272     if(keyboard_state==push)
273     {keyboard_state=0;
274       set_value=0;
275       pointer(i_1,i_2,on);}
276     } break;
277     }
278     }
279     }
280     }
281     show_var_enable=0;
282 }
```

ЗМ.	Лист	№ докум.	Підпис	Дата

ФЗ 8.17100.04.370 ПЗ

Арк

77

```

283
284 //экран быстрого заряда
285 void fast_charge(void)
286 {mode=mode_fast_charge;
287   show_var_enable=0;
288   nlcd_Clear();
289   i_1=2; i_2=3;
290   pi=2; nlcd_Printf(">",0,pi);
291   nlcd_Printf("ШВИДКИЙ ЗАРЯД-IU",0,0);
292   nlcd_Line (0,12,95,12);
293   nlcd_Printf("I: ----A, --:--",1,2);
294   nlcd_Printf("U: ----V, --:--",1,3);
295   nlcd_Printf("#####",0,4);
296   nlcd_Printf("Id: ---A; --:--",0,5);
297   nlcd_Printf("Uд: ----V; ",0,6);
298   nlcd_Printf("Cв: ---Ah; t:--C",0,7);
299
300   show_var_enable=1;
301   while(level_1!=0)
302   {
303     if(keyboard_state==start)
304     {keyboard_state=0;
305       if(charging_process==0)
306       {charging_process=1;
307         pointer(i_1,i_2,off);
308         while(PINC.4==0){;}
309         while(charging_process==1)
310         {if(keyboard_state==start)
311           {keyboard_state=0;
312             charging_process=0;
313             pointer(i_1,i_2,on);
314           }
315         }
316       }
317     }
318     if(keyboard_state==rev)
319     {keyboard_state=0;
320       pointer(i_1,i_2,up);}
321     if(keyboard_state==ff)
322     {keyboard_state=0;
323       pointer(i_1,i_2,down);}
324     if(keyboard_state==esc)
325     {keyboard_state=0;
326       show_var_enable=0;
327       mode=0;
328       level_1=0;
329       nlcd_Clear();
330       main_screen();}
331     if(keyboard_state==push)
332     {keyboard_state=0;
333       set_value=1;
334       switch(change_var=pi)
335       {case 2: while(set_value==1)
336         {if(keyboard_state==rev)
337           {keyboard_state=0;
338             if(double_ff==0)
339             {Ifast_ch_set=Ifast_ch_set-0.1;}
340             else {Ifast_ch_set=Ifast_ch_set-0.5;}
341             double_ff=0x0f;
342             OCR0=Ifast_ch_set*20;
343           }
344           if(keyboard_state==ff)
345           {keyboard_state=0;
346             if(double_ff==0)
347             {Ifast_ch_set=Ifast_ch_set+0.1;}
348             else {Ifast_ch_set=Ifast_ch_set+0.5;}
349             double_ff=0x0f;
350             OCR0=Ifast_ch_set*20;
351           }
352           if(keyboard_state==push)
353           {keyboard_state=0;

```

File: spt.c, Date: 13/12/18, Time: 12:17:42

```
354         set_value=0;
355         pointer(i_1,i_2,on);}
356     } break;
357     case 3: while(set_value==1)
358         {if(keyboard_state==rev)
359             {keyboard_state=0;
360              if(double_ff==0)
361                  {Ufast_ch_set=Ufast_ch_set-0.1;}
362              else {Ufast_ch_set=Ufast_ch_set-0.5;}
363              double_ff=0x0f;
364              }
365             if(keyboard_state==ff)
366                 {keyboard_state=0;
367                  if(double_ff==0)
368                      {Ufast_ch_set=Ufast_ch_set+0.1;}
369                  else {Ufast_ch_set=Ufast_ch_set+0.5;}
370                  double_ff=0x0f;
371                  }
372             if(keyboard_state==push)
373                 {keyboard_state=0;
374                  set_value=0;
375                  pointer(i_1,i_2,on);}
376             } break;
377         }
378     }
379     }
380     show_var_enable=0;
381 }
382
383 //экран тест АКБ
384 void test(void)
385 {nlcd_Clear();
386  i_1=2; i_2=5;
387  pi=2; nlcd_Printf(">",0,pi);
388  nlcd_Printf("ТЕСТ/ВІДНОВЛЕННЯ",0,0);
389  nlcd_Line (0,12,95,12);
390  nlcd_Printf("Швидкий тест",1,2);
391  nlcd_Printf("Повний цикл",1,3);
392  nlcd_Printf("Відновлення",1,4);
393  nlcd_Printf("Грозряда: ---А",1,5);
394  nlcd_Printf("т акб:    --С",1,6);
395  nlcd_Printf("#####",0,7);}
396
397 //экран режима контактной сварки
398 void weld(void)
399 {mode=mode_weld;
400  show_var_enable=0;
401  nlcd_Clear();
402  i_1=2; i_2=4;
403  pi=2; nlcd_Printf(">",0,pi);
404  nlcd_Printf("КОНТАКТНЕ З-ННЯ.",0,0);
405  nlcd_Line (0,12,95,12);
406  nlcd_Printf("Імпульс, с: ---",1,2);
407  nlcd_Printf("Кільк.імп:  --",1,3);
408  nlcd_Printf("Потужність:---%",1,4);
409  nlcd_Line (0,44,95,44);
410  nlcd_Printf("Тр1:  Тр2:  Р-р:",0,6);
411  nlcd_Printf("--С;  --С;  --С",0,7);
412
413  show_var_enable=1;
414  while(level_1!=0)
415      {
416          if(keyboard_state==rev)
417              {keyboard_state=0;
418               pointer(i_1,i_2,up);}
419          if(keyboard_state==ff)
420              {keyboard_state=0;
421               pointer(i_1,i_2,down);}
422          if(keyboard_state==esc)
423              {keyboard_state=0;
424               show_var_enable=0;
```


File: spt.c, Date: 13/12/18, Time: 12:17:42

```
425         mode=0;
426         level_1=0;
427         nlcd_clear();
428         main_screen();}
429     if(keyboard_state==push)
430     {keyboard_state=0;
431       set_value=1;
432       switch(change_var=pi)
433       {case 2: while(set_value==1)
434         {if(keyboard_state==ff)
435           {keyboard_state=0;
436             if(double_ff==0|welding_time>=0.9)
437               {if(welding_time>0.99){welding_time=1; manual=1;}
438                 else{welding_time=welding_time+0.01;}
439               }
440             else {welding_time=welding_time+0.1;}
441             double_ff=0x0f;
442           }
443         if(keyboard_state==rev)
444           {keyboard_state=0;
445             manual=0;
446             if(double_ff==0|welding_time<0.1)
447               {if(welding_time<0.01){welding_time=0;}
448                 else{welding_time=welding_time-0.01;}}
449             else {welding_time=welding_time-0.1;}
450             double_ff=0x0f;
451           }
452         if(keyboard_state==push)
453           {keyboard_state=0;
454             set_value=0;
455             pointer(i_1,i_2,on);}
456         } break;
457       case 3: while(set_value==1)
458         {if(keyboard_state==rev)
459           {keyboard_state=0;
460             quantity--;
461           }
462         if(keyboard_state==ff)
463           {keyboard_state=0;
464             quantity++;
465           }
466         if(keyboard_state==push)
467           {keyboard_state=0;
468             set_value=0;
469             pointer(i_1,i_2,on);}
470         } break;
471       case 4: while(set_value==1)
472         {if(keyboard_state==rev)
473           {keyboard_state=0;
474             if(double_ff==0)
475               {welding_current--;}
476             else {welding_current=welding_current-10;}
477             double_ff=0x0f;
478           }
479         if(keyboard_state==ff)
480           {keyboard_state=0;
481             if(double_ff==0)
482               {welding_current++;}
483             else {welding_current=welding_current+10;}
484             double_ff=0x0f;
485           }
486         if(keyboard_state==push)
487           {keyboard_state=0;
488             set_value=0;
489             pointer(i_1,i_2,on);}
490         } break;
491       }
492     }
493   }
494   show_var_enable=0;
495 }
```

ЗМ.	Лист	№ докум.	Підпис	Дата
-----	------	----------	--------	------

ФЗ 8.17100.04.370 ПЗ

Арк

80

File: spt.c, Date: 13/12/18, Time: 12:17:42

```
496
497
498 //экран блока питания
499 void power_supply(void)
500 {nlcd_Clear();
501  i_1=2; i_2=4;
502  pi=2; nlcd_Printf(">",0,pi);
503  nlcd_Printf(" БЛОК ЖИВЛЕНИЯ",0,0);
504  nlcd_Line (0,12,95,12);
505  nlcd_Printf(" Устаб.: ----V",1,2);
506  nlcd_Printf(" Тстаб.: ----A",1,3);
507  nlcd_Printf("Обмеження ----A",1,4);
508  nlcd_Printf("Уфакт.: ----V",1,5);
509  nlcd_Printf("Ифакт.: ----A",1,6);
510  nlcd_Printf("Рфакт.: ----W",1,7);}
511
512
513 // External Interrupt 0 service routine
514 // прерывание по клавиатуре
515 interrupt [EXT_INT0] void ext_int0_isr(void)
516 {
517  switch(PINC&0x3D) //маска, сброс 8,7,2 бита
518  {case 0x3C: {keyboard_state=push; buztime=0x005f; break;} //нажатие энкодера
519   case 0x3D: {keyboard_state=ff; buztime=0x000f; break;} //шаг энкодера вперед
520   case 0x39: {keyboard_state=rev; buztime=0x000f; break;} //шаг энкодера назад
521   case 0x2D: {keyboard_state=start; buztime=0x005f; break;} //левая кнопка
522   case 0x35: {keyboard_state=esc; buztime=0x005f; break;} //правая кнопка
523   case 0x1D: {welding=1; //сварочная кнопка
524              OCR1A=welding_current*100;
525              temp_time=welding_time*255;
526              break;}
527  }
528 }
529
530 // External Interrupt 1 service routine
531 // прерывание по переходу 0 сети
532 interrupt [EXT_INT1] void ext_int1_isr(void)
533 {TCNT1=0;
534  PORTD.5=0;
535 }
536
537 // Timer 0 overflow interrupt service routine
538 interrupt [TIM0_OVF] void timer0_ovf_isr(void)
539 {
540  if(buztime!=0) //пищалка
541  {PORTD.0=1; buztime--;}
542  else{PORTD.0=0;}
543 }
544
545 //Timer1 output compare A interrupt service routine
546 interrupt [TIM1_COMPA] void timer1_compa_isr(void)
547 {if(welding==1)
548  {PORTD.5=1;
549  }
550  // else{PORTD.5=0;}
551 }
552
553
554 // Timer2 output compare interrupt service routine
555 interrupt [TIM2_COMP] void timer2_comp_isr(void)
556 {
557  if(show_var!=0) //Вывод переменных на экран
558  {show_var--;}
559  else{show_var=0x05;
560      if(show_var_enable==1&welding==0)
561      {switch(mode)
562       {case mode_fast_charge:sprintf(buf,"%4.1f",Ifast_ch_set);
563        nlcd_Print(buf,4,2);
564        sprintf(buf,"%4.1f",Ufast_ch_set);
565        nlcd_Print(buf,4,3);
566        break;
```

File: spt.c, Date: 13/12/18, Time: 12:17:42

```
567         case mode_charge:      sprintf(buf, "%3u", Cfch_set);
568                               nlcd_Print(buf, 4, 0);
569                               sprintf(buf, "%4.1f", I1fch_set);
570                               nlcd_Print(buf, 4, 1);
571                               sprintf(buf, "%4.1f", Ufch_set);
572                               nlcd_Print(buf, 4, 2);
573                               sprintf(buf, "%4.1f", I2fch_set);
574                               nlcd_Print(buf, 4, 3);
575                               break;
576     case mode_test: break;
577     case mode_weld:      if(manual==1){nlcd_Printf("PYUH", 12, 2);
578                               nlcd_Printf("---", 13, 3);}
579                               else{sprintf(buf, "%4.2f", welding_time);
580                               nlcd_Print(buf, 12, 2);
581                               sprintf(buf, "%3u", quantity);
582                               nlcd_Print(buf, 13, 3);}
583                               sprintf(buf, "%3u", welding_current);
584                               nlcd_Print(buf, 12, 4);
585                               break;
586     case mode_power_supply: break;
587     default: break;
588     }
589 }}
590
591
592 if(set_value!=0)
593 {GICR=0x80;
594   if(--blink_time==0x007f)
595     {pointer(i_1, i_2, off);}
596   if(blink_time==0){pointer(i_1, i_2, on);
597     blink_time=0x00ff;}
598   GICR=0xC0;
599 }
600 else
601 {if(charging_process==1)
602   { if(progress_time---=0)
603     {progress_time=0x1f;
604     if(progress_line<16)
605       {nlcd_Printf("#", progress_line++, 4);}
606     else{progress_line=0; nlcd_Printf("$$$$$$$$$$$$$$$$", progress_line, 4);}
607     }
608   }
609 }
610
611 if(double_ff!=0)           //При быстром вращении энкодера
612   {double_ff--;}         //значение изменяется на порядок быстрее
613
614 if(temp_time!=0)
615   {temp_time--;}
616 else{welding=0;}
617 }
618
619
620 void main(void)
621 {
622   // Declare your local variables here
623
624   // Input/Output Ports initialization
625   // Port A initialization
626   // Function: Bit7=In Bit6=In Bit5=In Bit4=In Bit3=In Bit2=In Bit1=In Bit0=In
627   DDRA=(0<<DDA7) | (0<<DDA6) | (0<<DDA5) | (0<<DDA4) | (0<<DDA3) | (0<<DDA2) | (0<<DDA1) | (0<<DDA0);
628   // State: Bit7=P Bit6=P Bit5=P Bit4=P Bit3=P Bit2=P Bit1=P Bit0=P
629   PORTA=(1<<PORTA7) | (1<<PORTA6) | (1<<PORTA5) | (1<<PORTA4) | (1<<PORTA3) | (1<<PORTA2) | (1<<PORTA1) | (1<<PORTA0);
630
631   // Port B initialization
632   // Function: Bit7=Out Bit6=Out Bit5=Out Bit4=Out Bit3=Out Bit2=Out Bit1=Out Bit0=Out
633   DDRB=(1<<DDB7) | (1<<DDB6) | (1<<DDB5) | (1<<DDB4) | (1<<DDB3) | (1<<DDB2) | (1<<DDB1) | (1<<DDB0);
634   // State: Bit7=0 Bit6=0 Bit5=0 Bit4=0 Bit3=0 Bit2=0 Bit1=0 Bit0=0
```

File: spt.c, Date: 13/12/18, Time: 12:17:42

```
635 PORTB=(0<<PORTB7) | (0<<PORTB6) | (0<<PORTB5) | (0<<PORTB4) | (0<<PORTB3) | (0<<PORTB2) | (
    0<<PORTB1) | (0<<PORTB0);
636
637 // Port C initialization
638 // Function: Bit7=In Bit6=In Bit5=In Bit4=In Bit3=In Bit2=In Bit1=In Bit0=In
639 DDRC=(0<<DDC7) | (0<<DDC6) | (0<<DDC5) | (0<<DDC4) | (0<<DDC3) | (0<<DDC2) | (0<<DDC1) | (0
    <<DDC0);
640 // State: Bit7=P Bit6=P Bit5=P Bit4=P Bit3=P Bit2=P Bit1=P Bit0=P
641 PORTC=(1<<PORTC7) | (1<<PORTC6) | (1<<PORTC5) | (1<<PORTC4) | (1<<PORTC3) | (1<<PORTC2) | (
    1<<PORTC1) | (1<<PORTC0);
642
643 // Port D initialization
644 // Function: Bit7=Out Bit6=Out Bit5=Out Bit4=Out Bit3=In Bit2=In Bit1=Out Bit0=Out
645 DDRD=(1<<DDD7) | (1<<DDD6) | (1<<DDD5) | (1<<DDD4) | (0<<DDD3) | (0<<DDD2) | (1<<DDD1) | (1
    <<DDD0);
646 // State: Bit7=0 Bit6=0 Bit5=0 Bit4=0 Bit3=P Bit2=P Bit1=0 Bit0=0
647 PORTD=(0<<PORTD7) | (0<<PORTD6) | (0<<PORTD5) | (0<<PORTD4) | (1<<PORTD3) | (1<<PORTD2) | (
    0<<PORTD1) | (0<<PORTD0);
648
649 // Timer/Counter 0 initialization
650 // Clock source: System Clock
651 // Clock value: 8000,000 kHz
652 // Mode: Fast PWM top=0xFF
653 // OCO output: Non-Inverted PWM
654 // Timer Period: 0,032 ms
655 // Output Pulse(s):
656 // OCO Period: 0,032 ms Width: 0 us
657 TCCR0=(1<<WGM00) | (1<<COM01) | (0<<COM00) | (1<<WGM01) | (0<<CS02) | (0<<CS01) | (1<<CS00)
    ;
658 TCNT0=0x00;
659 OCR0=0x00;
660
661 // Timer/Counter 1 initialization
662 // Clock source: System Clock
663 // Clock value: 1000,000 kHz
664 // Mode: CTC top=OCR1A
665 // OCL1A output: Clear on compare match
666 // OCL1B output: Disconnected
667 // Noise Canceler: Off
668 // Input Capture on Falling Edge
669 // Timer Period: 1 us
670 // Output Pulse(s):
671 // OCL1A Period: 1 us
672 // Timer1 Overflow Interrupt: Off
673 // Input Capture Interrupt: Off
674 // Compare A Match Interrupt: Off
675 // Compare B Match Interrupt: Off
676 TCCR1A=(0<<COM1A1) | (0<<COM1A0) | (0<<COM1B1) | (0<<COM1B0) | (0<<FOC1A) | (0<<FOC1B) | (0
    <<WGM11) | (0<<WGM10);
677 TCCR1B=(0<<ICNC1) | (0<<ICES1) | (0<<WGM13) | (1<<WGM12) | (0<<CS12) | (1<<CS11) | (0<<CS10
    );
678 TCNT1H=0x00;
679 TCNT1L=0x00;
680 ICR1H=0x00;
681 ICR1L=0x00;
682 OCR1AH=0x00;
683 OCR1AL=0x00;
684 OCR1BH=0x00;
685 OCR1BL=0x00;
686
687 // Timer/Counter 2 initialization
688 // Clock source: TOSC1 pin
689 // Clock value: PCK2/128
690 // Mode: CTC top=OCR2A
691 // OC2 output: Disconnected
692 ASSR=1<<AS2;
693 TCCR2=(0<<PWM2) | (0<<COM21) | (0<<COM20) | (1<<CTC2) | (1<<CS22) | (0<<CS21) | (1<<CS20);
694 TCNT2=0x00;
695 OCR2=0x00;
696
697 // Timer(s)/Counter(s) Interrupt(s) initialization
```

						ФЗ 8.17100.04.370 ПЗ	Арк
ЗМ.	Лист	№ докум.	Підпис	Дата			83

File: spt.c, Date: 13/12/18, Time: 12:17:42

```
698 TIMSK=(1<<OCIE2) | (0<<TOIE2) | (0<<TICIE1) | (1<<OCIE1A) | (0<<OCIE1B) | (0<<TOIE1) | (0<<
699 OCIE0) | (1<<TOIE0);
700
701 // External Interrupt(s) initialization
702 // INT0: On
703 // INT0 Mode: Falling Edge
704 // INT1: On
705 // INT1 Mode: Rising Edge
706 // INT2: Off
707 GICR|=(1<<INT1) | (1<<INT0) | (0<<INT2);
708 MCUCR=(1<<ISC11) | (1<<ISC10) | (1<<ISC01) | (0<<ISC00);
709 MCUCSR=(0<<ISC2);
710 GIFR=(1<<INTF1) | (1<<INTF0) | (0<<INTF2);
711
712
713 // USART initialization
714 // USART disabled
715 UCSRB=(0<<RXCE) | (0<<TXCE) | (0<<UDRIE) | (0<<RXEN) | (0<<TXEN) | (0<<UCSZ2) | (0<<RXB8)
716 | (0<<TXB8);
717
718 // Analog Comparator initialization
719 // Analog Comparator: Off
720 // The Analog Comparator's positive input is
721 // connected to the AIN0 pin
722 // The Analog Comparator's negative input is
723 // connected to the AIN1 pin
724 ACSR=(1<<ACD) | (0<<ACBG) | (0<<ACO) | (0<<ACI) | (0<<ACIE) | (0<<ACIC) | (0<<ACIS1) | (0<<
725 ACIS0);
726
727 //// ADC initialization
728 //// ADC Clock frequency: 1000,000 kHz
729 //// ADC Voltage Reference: AREF pin
730 //// ADC Auto Trigger Source: Free Running
731 //// Only the 8 most significant bits of
732 //// the AD conversion result are used
733 //ADMUX=FIRST_ADC_INPUT | ADC_VREF_TYPE;
734 //ADCSRA=(1<<ADEN) | (1<<ADSC) | (1<<ADATE) | (0<<ADIF) | (1<<ADIE) | (0<<ADPS2) | (1<<
735 ADPS1) | (1<<ADPS0);
736 //SFIOR=(0<<ADTS2) | (0<<ADTS1) | (0<<ADTS0);
737
738 // SPI initialization
739 // SPI disabled
740 SPCR=(0<<SPIE) | (0<<SPE) | (0<<DORD) | (0<<MSTR) | (0<<CPOL) | (0<<CPHA) | (0<<SPR1) | (0<
741 <SPR0);
742
743 // TWI initialization
744 // TWI disabled
745 TWCR=(0<<TWEA) | (0<<TWSTA) | (0<<TWSTO) | (0<<TWEN) | (0<<TWIE);
746
747 // Global enable interrupts
748 #asm("sei")
749
750 nlcd_Init();
751 nlcd_Clear();
752 main_screen();
753
754 I1fch_set=(float)Cfch_set/8;
755 I2fch_set=(float)Cfch_set/40;
756
757 OCR1A=welding_current*100;
758 TCNT1=0x0000;
759 // temp_time=welding_time;
760
761 while (1)
762 {
763 if(keyboard_state==rev)
764 {keyboard_state=0;
765 pointer(i_1,i_2,up);}
766 }
```

ЗМ.	Лист	№ докум.	Підпис	Дата
-----	------	----------	--------	------

ФЗ 8.17100.04.370 ПЗ

Арк

84

File: spt.c, Date: 13/12/18, Time: 12:17:42

```
764     if(keyboard_state==ff)
765     {keyboard_state=0;
766       pointer(i_1,i_2,down);}
767
768     if(keyboard_state==push)
769     {keyboard_state=0;
770       level_1=pi;
771       switch(level_1)
772       {case 2: fast_charge();break;
773        case 3: charge();break;
774        case 4: test();break;
775        case 5: weld();break;
776        case 6: power_supply();break;}
777     }
778   }
779 }
780
```

					ФЗ 8.17100.04.370 ПЗ	Арк
						85
Зм.	Лист	№ докум.	Підпис	Дата		

Додаток Б – перелік графічних функцій обробки та виводу зображення

File: C:\cvavr\inc\nokia_lcd.h, Date: 04/12/18, Time: 20:09

```
1 //*****
2 // File.....: nokia_lcd.h
3 //*****
4 // Notice: Все управляющие контакты LCD-контроллера должны быть подключены к
5 // одному и тому же порту на микроконтроллере
6 //*****
7 #ifndef _NOKIA_LCD_H_
8 #define _NOKIA_LCD_H_
9 #include <mega32a.h>
10 #include <delay.h>
11 //*****
12 // Настройка библиотеки
13
14 // Порт, к которому подключен LCD-контроллер NOKIA 1100
15 #define PORT_LCD PORTB
16 #define PIN_LCD PINB
17 #define DDR_LCD DDRB
18
19 // Номера выводов порта, к которым подключены выводы LCD-контроллера
20 #define SCLK_LCD_PIN 7
21 #define SDA_LCD_PIN 5
22 #define CS_LCD_PIN 6
23 #define RST_LCD_PIN 4
24
25 //*****
26 // Макросы и определения
27
28 #define SCLK_LCD_SET PORT_LCD |= (1<<SCLK_LCD_PIN)
29 #define SDA_LCD_SET PORT_LCD |= (1<<SDA_LCD_PIN)
30 #define CS_LCD_SET PORT_LCD |= (1<<CS_LCD_PIN)
31 #define RST_LCD_SET PORT_LCD |= (1<<RST_LCD_PIN)
32
33 #define SCLK_LCD_RESET PORT_LCD &= ~(1<<SCLK_LCD_PIN)
34 #define SDA_LCD_RESET PORT_LCD &= ~(1<<SDA_LCD_PIN)
35 #define CS_LCD_RESET PORT_LCD &= ~(1<<CS_LCD_PIN)
36 #define RST_LCD_RESET PORT_LCD &= ~(1<<RST_LCD_PIN)
37
38 #define CMD_LCD_MODE 0
39 #define DATA_LCD_MODE 1
40
41 #define INV_MODE_ON 0
42 #define INV_MODE_OFF 1
43
44 #define PIXEL_ON 0
45 #define PIXEL_OFF 1
46 #define PIXEL_INV 2
47
48
49 //*****
50 // Прототипы функций
51 #pragma used+
52 void nlcd_Init(void);
53 void nlcd_Clear(void);
54 void nlcd_SendByte(char mode,unsigned char c);
55 void nlcd_Putc(unsigned char c);
56 void nlcd_Print(char * message, char x, char y);
57 void nlcd_Printf(unsigned flash char * message, char x, char y);
58 void nlcd_GotoXY(char x,char y);
59 void nlcd_GotoXY_pix(char x,char y);
60 void nlcd_Inverse(unsigned char mode);
61
62 void nlcd_Line(unsigned char x1,unsigned char y1, unsigned char x2,unsigned char y2);
63 void nlcd_Pixel(unsigned char x,unsigned char y, unsigned char pixel_mode);
64
65 #pragma used-
66 #pragma library nokia_lcd.lib
67 #endif /* _NOKIA1100_LCD_LIB_H_ */
68
```

									ФЗ 8.17100.04.370 ПЗ	Арк
ЗМ.	Лист	№ докум.	Підпис	Дата						86

Додаток В – функції роботи з графічним дисплеєм

File: C:\cvavr\LIB\nokia_lcd.lib, Date: 04/12/18, Time: 20:

```
1 //*****
2 // File.....: nokia_lcd.lib
3 //*****
4 // Notice: Все управляющие контакты LCD-контроллера должны быть подключены к
5 // одному и тому же порту на микроконтроллере
6 //*****
7 // изменено для квар column_ptr[i] = pgm_read_byte(&(S_box[ column_ptr[i] ]));->column_ptr[
8   i]=S_box[column_ptr[i]];
9 #include <nokia_lcd.h>
10 #include <nokia_lcd_fnt.h> // Подключаем шрифт (будет размещен в программной памяти)
11 #include <stdlib.h>
12
13 // Макросы для работы с битами
14 #define ClearBit(reg, bit)    reg &= (~(1<<(bit)))
15 #define SetBit(reg, bit)     reg |= (1<<(bit))
16 #define InvBit(reg, bit)     reg ^= 1<<bit
17
18 //*****
19 // Инициализация контроллера
20 void nlcd_Init(void)
21 {
22     // Инициализируем порт на вывод для работы с LCD-контроллером
23     DDR_LCD |= (1<<SCLK_LCD_PIN) | (1<<SDA_LCD_PIN) | (1<<CS_LCD_PIN) | (1<<RST_LCD_PIN);
24
25     CS_LCD_RESET;
26     RST_LCD_RESET;
27
28     delay_ms(10); // выждем не менее 5мс для установки генератора(менее 5 мс
29 // может не работать)
30
31     RST_LCD_SET;
32
33     nlcd_SendByte(CMD_LCD_MODE,0x20); // Запись в регистр Vop
34     nlcd_SendByte(CMD_LCD_MODE,0x92);
35     nlcd_SendByte(CMD_LCD_MODE,0xA4); // all on/normal display
36     nlcd_SendByte(CMD_LCD_MODE,0x2F); // Power control set(charge pump on/off)
37     nlcd_SendByte(CMD_LCD_MODE,0x40); // set start row address = 0
38     nlcd_SendByte(CMD_LCD_MODE,0xB0); // установить Y-адрес = 0
39     nlcd_SendByte(CMD_LCD_MODE,0x10); // установить X-адрес, старшие 3 бита
40     nlcd_SendByte(CMD_LCD_MODE,0x0); // установить X-адрес, младшие 4 бита
41     //nlcd_SendByte(CMD_LCD_MODE,0xC8); // mirror Y axis (about X axis)
42     //nlcd_SendByte(CMD_LCD_MODE,0xA1); // Инvertировать экран по горизонтали
43     nlcd_SendByte(CMD_LCD_MODE,0xAC); // set initial row (R0) of the display
44     nlcd_SendByte(CMD_LCD_MODE,0x07);
45     //nlcd_SendByte(CMD_LCD_MODE,0xF9); //
46     nlcd_SendByte(CMD_LCD_MODE,0xAF); // экран вкл/выкл
47
48     nlcd_Clear(); // clear LCD
49
50     nlcd_SendByte(CMD_LCD_MODE,0xA7); // инvertированный экран
51     //delay_ms(300);//было по умолчанию (500) изменил для протейса-долго и тормоз// 1/2 Sec
52     delay
53     nlcd_SendByte(CMD_LCD_MODE,0xA6); // нормальный экран (не инvertированный)
54     //delay_ms(100);//(1000)-тоже самое здесь для протейса // 1 Sec
55     delay
56 }
57
58 //*****
59 // Очистка экрана
60 void nlcd_Clear(void)
61 {
62     int i=0;
63     nlcd_SendByte(CMD_LCD_MODE,0x40); // Y = 0
64     nlcd_SendByte(CMD_LCD_MODE,0xB0);
65     nlcd_SendByte(CMD_LCD_MODE,0x10); // X = 0
66     nlcd_SendByte(CMD_LCD_MODE,0x00);
67     //nlcd_SendByte(CMD_LCD_MODE,0xAE); // disable display;
68
69     for (i=0;i<864;i++) {nlcd_SendByte(DATA_LCD_MODE,0x00);};
70 }
```

									ФЗ 8.17100.04.370 ПЗ	Арк
ЗМ.	Лист	№ докум.	Підпис	Дата						87

File: C:\cvavr\LIB\nokia_lcd.lib, Date: 04/12/18, Time: 20:

```
67 //nlcd_SendByte(CMD_LCD_MODE,0x07);
68 //nlcd_SendByte(CMD_LCD_MODE,0xAF); // enable display;
69 }
70
71
72 //*****
73 // Передача байта (команды или данных) на LCD-контроллер
74 // mode: CMD_LCD_MODE - передаем команду
75 // DATA_LCD_MODE - передаем данные
76 // c: значение передаваемого байта
77 void nlcd_SendByte(char mode,unsigned char c)
78 { unsigned char i=0;
79   CS_LCD_RESET;
80   SCLK_LCD_RESET;
81
82   if(mode) SDA_LCD_SET;
83   else SDA_LCD_RESET;
84
85   SCLK_LCD_SET;
86
87   for(i=0;i<8;i++)
88   {
89     SCLK_LCD_RESET;
90
91     if(c & 0x80) SDA_LCD_SET;
92     else SDA_LCD_RESET;
93
94     SCLK_LCD_SET;
95     c <<= 1;
96     // delay_ms(3);
97   }
98   CS_LCD_SET;
99 }
100
101
102
103 //*****
104 // Устанавливает курсор в необходимое положение. Отсчет начинается в верхнем
105 // левом углу. По горизонтали 16 знакомест, по вертикали - 8
106 // x: 0..15
107 // y: 0..7
108 void nlcd_GotoXY(char x,char y)
109 {
110   x=x*6; // Переходим от координаты в пикселях к координатам в знакоместах
111   nlcd_SendByte(CMD_LCD_MODE,(0xB0|(y&0x0F))); // установка адреса по Y: 0100 yyyu -
112   nlcd_SendByte(CMD_LCD_MODE,(0x00|(x&0x0F))); // установка адреса по X: 0000 xxxx -
113   // биты (x3 x2 x1 x0)
114   nlcd_SendByte(CMD_LCD_MODE,(0x10|((x>>4)&0x07))); // установка адреса по X: 0010 0xxx -
115   // биты (x6 x5 x4)
116 }
117
118
119 //*****
120 // Вывод символа на LCD-контроллер в текущее место
121 // c: код символа
122 void nlcd_Putc(unsigned char c)
123 { unsigned char i = 0;
124   for ( i = 0; i < 5; i++ )
125   { if(c<122){nlcd_SendByte(DATA_LCD_MODE,nlcd_Font[c-32][i]<<1);}
126     else{nlcd_SendByte(DATA_LCD_MODE,nlcd_Font[c-101][i]<<1);}
127   }
128   nlcd_SendByte(DATA_LCD_MODE,0x00); // Зазор между символами по горизонтали в 1 пиксель
129 }
130
131
132 //*****
133 // Вывод строки символов на LCD-контроллер в текущее место. Если строка выходит
134 // за экран в текущей строке, то остаток переносится на следующую строку.
```

					ФЗ 8.17100.04.370 ПЗ	Арк
ЗМ.	Лист	№ докум.	Підпис	Дата		88

File: C:\cvavr\LIB\nokia_lcd.lib, Date: 04/12/18, Time: 20:

```
135 // message: указатель на строку символов. 0x00 - признак конца строки.
136 void nlcd_Print(char * message, char x, char y)
137 {
138     nlcd_GotoXY(x,y);
139     while (*message) // Конец строки обозначен нулем
140         {nlcd_Putc(*message++);}
141 }
142
143 //*****
144 // Вывод строки символов на LCD-контроллер в текущее место из программной памяти.
145 // Если строка выходит за экран в текущей строке, то остаток переносится на следующую
146 // строку.
147 // message: указатель на строку символов в программной памяти. 0x00 - признак конца строки
148 .
149 void nlcd_Printf(unsigned flash char * message, char x, char y)
150 {
151     unsigned char data;
152     nlcd_GotoXY(x,y);
153     data=*message;
154     while (data)
155     {
156         nlcd_Putc(data);
157         message++;
158         data=*message;
159     }
160 }
161 //*****
162 // Устанавливает курсор в пикселях. Отсчет начинается в верхнем
163 // левом углу. По горизонтали 96 пикселей, по вертикали - 65
164 // x: 0..95
165 // y: 0..64
166 void nlcd_GotoXY_pix(char x,char y)
167 {
168     nlcd_SendByte(CMD_LCD_MODE,(0xB0|(y&0x0F))); // установка адреса по Y: 0100 yyyу
169     nlcd_SendByte(CMD_LCD_MODE,(0x00|(x&0x0F))); // установка адреса по X: 0000 xxxx -
170 // биты (x3 x2 x1 x0)
171     nlcd_SendByte(CMD_LCD_MODE,(0x10|((x>>4)&0x07))); // установка адреса по X: 0010 0xxx -
172 // биты (x6 x5 x4)
173 }
174 //*****
175 // Устанавливает режим инверсии всего экрана. Данные на экране не изменяются, только
176 // инвертируются
177 // mode: INV_MODE_ON или INV_MODE_OFF
178 void nlcd_Inverse(unsigned char mode)
179 {
180     if (mode) nlcd_SendByte(CMD_LCD_MODE,0xA6);
181     else nlcd_SendByte(CMD_LCD_MODE,0xA7);
182 }
183 //*****
184 // Вывод точки на LCD-экран NOKIA 1100
185 // x: 0..95 координата по горизонтали (отсчет от верхнего левого угла)
186 // y: 0..67 координата по вертикали
187 // pixel_mode: PIXEL_ON - для включения пиксела
188 // PIXEL_OFF - для выключения пиксела
189 void nlcd_Pixel(char x, char y, unsigned char pixel_mode){
190     //Объявляем переменные
191     char bite;
192     char page;
193     char data_p=0;
194     //Выход если точка лежит вне экрана
195     if((x>95)|(y>67)) return;
196     //page-номер страницы
197     page=y/8;
```

					ФЗ 8.17100.04.370 ПЗ	Арк
3М.	Лист	№ докум.	Підпис	Дата		89

```

200 //bite-Номер байта который нам предстоит вывести
201 bite=y%8;
202
203 switch(pixel_mode)
204 {
205     case PIXEL_ON:
206         SetBit(data_p, y%8);           // Включаем пиксел
207         break;
208     case PIXEL_OFF:
209         ClearBit(data_p, y%8);        // Выключаем пиксел
210         break;
211 }
212
213 nlcd_GotoXY_pix(x,page);
214 nlcd_SendByte(DATA_LCD_MODE,data_p);
215 }
216
217
218
219 //*****
220 // Вывод линии на LCD-экран NOKIA 1100
221 // x1, x2: 0..95 координата по горизонтали (отсчет от верхнего левого угла)
222 // y1, y2: 0..67 координата по вертикали
223
224 void nlcd_Line (unsigned char x1,unsigned char y1, unsigned char x2,unsigned char y2)
225 {
226     int dy, dx;
227     signed char addx = 1, addy = 1;
228     signed int P, diff;
229
230     unsigned char i = 0;
231
232     dx = abs((signed char) (x2 - x1));
233     dy = abs((signed char) (y2 - y1));
234
235     if(x1 > x2) addx = -1;
236     if(y1 > y2) addy = -1;
237
238     if(dx >= dy)
239     {
240         dy *= 2;
241         P = dy - dx;
242
243         diff = P - dx;
244
245         for(; i<=dx; ++i)
246         { nlcd_Pixel(x1, y1, PIXEL_ON);
247
248             if(P < 0)
249             {
250                 P += dy;
251                 x1 += addx;
252             }
253             else
254             {
255                 P += diff;
256                 x1 += addx;
257                 y1 += addy;
258             }
259         }
260     }
261     else
262     {
263         dx *= 2;
264         P = dx - dy;
265         diff = P - dy;
266
267         for(; i<=dy; ++i)
268         { nlcd_Pixel(x1, y1, PIXEL_ON);
269
270             if(P < 0)

```

File: C:\cvavr\LIB\nokia_lcd.lib, Date: 04/12/18, Time: 20:

```
271         {
272             P += dx;
273             y1 += addy;
274         }
275         else
276         {
277             P += diff;
278             x1 += addx;
279             y1 += addy;
280         }
281     }
282 }
283 }
```

					ФЗ 8.17100.04.370 ПЗ	Арк
ЗМ.	Лист	№ докум.	Підпис	Дата		91

Додаток Г - кодування символів 5x8 згідно стандарту ASCII

File: C:\cvavr\INC\nokia_lcd_fnt.h, Date: 04/12/18, Time: 2

```
1 //*****
2 // File.....: nokia_lcd_fnt.h
3 //*****
4 // Notice:
5 //*****
6 #ifndef _NOKIA1100_LCD_FONT_H_
7 #define _NOKIA1100_LCD_FONT_H_
8
9 //*****
10 // Формат символов: 5x8. Установленный бит соответствует включенному пикселю
11 const unsigned char nlcd_Font[][5]={
12     { 0x00, 0x00, 0x00, 0x00, 0x00 }, // 0 0 32
13     { 0x00, 0x00, 0x2f, 0x00, 0x00 }, // ! 1 1 33
14     { 0x00, 0x07, 0x00, 0x07, 0x00 }, // " 2 2 34
15     { 0x7f, 0x7f, 0x7f, 0x7f, 0x7f }, // # 3 3 залитый квадрат
16     { 0x7f, 0x41, 0x41, 0x41, 0x7f }, // $ 4 4 пустой квадрат
17     { 0xc4, 0xc8, 0x10, 0x26, 0x46 }, // % 5 5
18     { 0x36, 0x49, 0x55, 0x22, 0x50 }, // & 6 6
19     { 0x00, 0x07, 0x05, 0x07, 0x00 }, // ' 7 7
20     { 0x00, 0x1c, 0x22, 0x41, 0x00 }, // ( 8 8
21     { 0x00, 0x41, 0x22, 0x1c, 0x00 }, // ) 9 9
22     { 0x14, 0x08, 0x3e, 0x08, 0x14 }, // * 10 10
23     { 0x08, 0x08, 0x3e, 0x08, 0x08 }, // + 11 11 43
24     { 0x00, 0x00, 0x50, 0x30, 0x00 }, // , 12 12
25     { 0x10, 0x10, 0x10, 0x10, 0x10 }, // - 13 13
26     { 0x00, 0x60, 0x60, 0x00, 0x00 }, // . 14 14
27     { 0x20, 0x10, 0x08, 0x04, 0x02 }, // / 15 15
28     { 0x3e, 0x51, 0x49, 0x45, 0x3e }, // 0 16 16
29     { 0x00, 0x42, 0x7f, 0x40, 0x00 }, // 1 17
30     { 0x42, 0x61, 0x51, 0x49, 0x46 }, // 2 18
31     { 0x21, 0x41, 0x45, 0x4b, 0x31 }, // 3 19
32     { 0x18, 0x14, 0x12, 0x7f, 0x10 }, // 4 20
33     { 0x27, 0x45, 0x45, 0x45, 0x39 }, // 5 21
34     { 0x3c, 0x4a, 0x49, 0x49, 0x30 }, // 6 22
35     { 0x01, 0x71, 0x09, 0x05, 0x03 }, // 7 23
36     { 0x36, 0x49, 0x49, 0x49, 0x36 }, // 8 24
37     { 0x06, 0x49, 0x49, 0x29, 0x1e }, // 9 25
38     { 0x00, 0x36, 0x36, 0x00, 0x00 }, // : 26
39     { 0x00, 0x56, 0x36, 0x00, 0x00 }, // ; 27
40     { 0x08, 0x14, 0x22, 0x41, 0x00 }, // < 28
41     { 0x14, 0x14, 0x14, 0x14, 0x14 }, // = 29
42     { 0x00, 0x7f, 0x3e, 0x1c, 0x08 }, // > 30
43     { 0x02, 0x01, 0x51, 0x09, 0x06 }, // ? 31
44     { 0x32, 0x49, 0x59, 0x51, 0x3e }, // @ 32
45
46     { 0x7e, 0x11, 0x11, 0x11, 0x7e }, // A 33
47     { 0x7f, 0x49, 0x49, 0x49, 0x36 }, // B 34
48     { 0x3e, 0x41, 0x41, 0x41, 0x22 }, // C 35
49     { 0x7f, 0x41, 0x41, 0x22, 0x1c }, // D 36
50     { 0x7f, 0x49, 0x49, 0x49, 0x41 }, // E 37
51     { 0x7f, 0x09, 0x09, 0x09, 0x01 }, // F 38
52     { 0x3e, 0x41, 0x49, 0x49, 0x7a }, // G 39
53     { 0x7f, 0x08, 0x08, 0x08, 0x7f }, // H 40
54     { 0x00, 0x41, 0x7f, 0x41, 0x00 }, // I
55     { 0x20, 0x40, 0x41, 0x3f, 0x01 }, // J
56     { 0x7f, 0x08, 0x14, 0x22, 0x41 }, // K
57     { 0x7f, 0x40, 0x40, 0x40, 0x40 }, // L
58     { 0x7f, 0x02, 0x0c, 0x02, 0x7f }, // M
59     { 0x7f, 0x04, 0x08, 0x10, 0x7f }, // N
60     { 0x3e, 0x41, 0x41, 0x41, 0x3e }, // O
61     { 0x7f, 0x09, 0x09, 0x09, 0x06 }, // P
62     { 0x3e, 0x41, 0x51, 0x21, 0x5e }, // Q
63     { 0x7f, 0x09, 0x19, 0x29, 0x46 }, // R 50
64     { 0x46, 0x49, 0x49, 0x49, 0x31 }, // S
65     { 0x01, 0x01, 0x7f, 0x01, 0x01 }, // T
66     { 0x3f, 0x40, 0x40, 0x40, 0x3f }, // U
67     { 0x1f, 0x20, 0x40, 0x20, 0x1f }, // V
68     { 0x3f, 0x40, 0x38, 0x40, 0x3f }, // W
69     { 0x63, 0x14, 0x08, 0x14, 0x63 }, // X
```

					Ф3 8.17100.04.370 ПЗ	Арк
ЗМ.	Лист	№ докум.	Підпис	Дата		92

```

70      { 0x07, 0x08, 0x70, 0x08, 0x07 }, // Y
71      { 0x61, 0x51, 0x49, 0x45, 0x43 }, // Z
72      { 0x00, 0x7F, 0x41, 0x41, 0x00 }, // [
73      { 0x55, 0x2A, 0x55, 0x2A, 0x55 }, // 60
74      { 0x00, 0x41, 0x41, 0x7F, 0x00 }, // ]
75      { 0x04, 0x02, 0x01, 0x02, 0x04 }, // ^
76      { 0x40, 0x40, 0x40, 0x40, 0x40 }, // _
77      { 0x00, 0x01, 0x02, 0x04, 0x00 }, // `
78      { 0x20, 0x54, 0x54, 0x54, 0x78 }, // a
79      { 0x7F, 0x48, 0x44, 0x44, 0x38 }, // b
80      { 0x38, 0x44, 0x44, 0x44, 0x20 }, // c
81      { 0x38, 0x44, 0x44, 0x48, 0x7F }, // d
82      { 0x38, 0x54, 0x54, 0x54, 0x18 }, // e
83      { 0x08, 0x7E, 0x09, 0x01, 0x02 }, // f 70
84      { 0x0C, 0x52, 0x52, 0x52, 0x3E }, // g
85      { 0x7F, 0x08, 0x04, 0x04, 0x78 }, // h
86      { 0x00, 0x44, 0x7D, 0x40, 0x00 }, // i
87      { 0x20, 0x40, 0x44, 0x3D, 0x00 }, // j
88      { 0x7F, 0x10, 0x28, 0x44, 0x00 }, // k
89      { 0x00, 0x41, 0x7F, 0x40, 0x00 }, // l
90      { 0x7C, 0x04, 0x18, 0x04, 0x78 }, // m
91      { 0x7C, 0x08, 0x04, 0x04, 0x78 }, // n
92      { 0x38, 0x44, 0x44, 0x44, 0x38 }, // o
93      { 0x7C, 0x14, 0x14, 0x14, 0x08 }, // p 80
94      { 0x08, 0x14, 0x14, 0x18, 0x7C }, // q
95      { 0x7C, 0x08, 0x04, 0x04, 0x08 }, // r
96      { 0x48, 0x54, 0x54, 0x54, 0x20 }, // s
97      { 0x04, 0x3F, 0x44, 0x40, 0x20 }, // t
98      { 0x3C, 0x40, 0x40, 0x20, 0x7C }, // u
99      { 0x1C, 0x20, 0x40, 0x20, 0x1C }, // v
100     { 0x3C, 0x40, 0x30, 0x40, 0x3C }, // w
101     { 0x44, 0x28, 0x10, 0x28, 0x44 }, // x
102     { 0x0C, 0x50, 0x50, 0x50, 0x3C }, // y
103     { 0x44, 0x64, 0x54, 0x4C, 0x44 }, // z 90 122
104
105
106     { 0x7e, 0x11, 0x11, 0x11, 0x7e}, // 0xc0 - 'A'. 90 192
107     { 0x7f, 0x49, 0x49, 0x49, 0x31}, // 0xc1 - 'B'.
108     { 0x7f, 0x49, 0x49, 0x49, 0x36}, // 0xc2 - 'B'.
109     { 0x7f, 0x01, 0x01, 0x01, 0x01}, // 0xc3 - 'Г'.
110     { 0x7e, 0x41, 0x41, 0x7f, 0xc0}, // 0xc4 - 'Д'.
111     { 0x7f, 0x49, 0x49, 0x49, 0x41}, // 0xc5 - 'E'.
112     { 0x77, 0x08, 0x7f, 0x08, 0x77}, // 0xc6 - 'Ж'.
113     { 0x22, 0x49, 0x49, 0x49, 0x36}, // 0xc7 - 'З'.
114     { 0x7f, 0x20, 0x10, 0x08, 0x7f}, // 0xc8 - 'И'.
115     { 0x7e, 0x21, 0x11, 0x09, 0x7e}, // 0xc9 - 'Й'.
116     { 0x7f, 0x08, 0x14, 0x22, 0x41}, // 0xca - 'К'.
117     { 0x40, 0x7e, 0x01, 0x01, 0x7f}, // 0xcb - 'Л'.
118     { 0x7f, 0x02, 0x04, 0x02, 0x7f}, // 0xcc - 'М'.
119     { 0x7f, 0x08, 0x08, 0x08, 0x7f}, // 0xcd - 'Н'.
120     { 0x3e, 0x41, 0x41, 0x41, 0x3e}, // 0xce - 'О'.
121     { 0x7f, 0x01, 0x01, 0x01, 0x7f}, // 0xcf - 'П'.
122     { 0x7f, 0x09, 0x09, 0x09, 0x06}, // 0xd0 - 'Р'.
123     { 0x3e, 0x41, 0x41, 0x41, 0x22}, // 0xd1 - 'С'.
124     { 0x01, 0x01, 0x7E, 0x01, 0x01}, // 0xd1 - 'С'.
125     { 0x27, 0x48, 0x48, 0x48, 0x3f}, // 0xd1 - 'С'.
126     { 0x0e, 0x11, 0x7F, 0x11, 0x0e}, // 0xd4 - 'Ф'.
127     { 0x63, 0x14, 0x08, 0x14, 0x63}, // 0xd5 - 'Х'.
128     { 0x7f, 0x40, 0x40, 0x7f, 0xc0}, // 0xd6 - 'Ц'.
129     { 0x07, 0x08, 0x08, 0x08, 0x7f}, // 0xd7 - 'Ч'.
130     { 0x7f, 0x40, 0x7f, 0x40, 0x7f}, // 0xd8 - 'Ш'.
131     { 0x7f, 0x40, 0x7E, 0x40, 0xff}, // 0xd9 - 'Щ'.
132     { 0x03, 0x01, 0x7f, 0x48, 0x30}, // 0xda - 'Ъ'.
133     { 0x7f, 0x48, 0x48, 0x30, 0x7f}, // 0xdb - 'Ы'.
134     { 0x7f, 0x48, 0x48, 0x48, 0x30}, // 0xdc - 'Ь'.
135     { 0x22, 0x41, 0x49, 0x49, 0x3e}, // 0xdd - 'Э'.
136     { 0x7f, 0x08, 0x3e, 0x41, 0x3e}, // 0xde - 'Ю'.
137     { 0x66, 0x19, 0x09, 0x09, 0x7f}, // 0xde - 'Ю'.
138     { 0x20, 0x54, 0x54, 0x54, 0x78}, // 0xe0 - 'а'.

```

File: C:\cvavr\INC\nokia_lcd_fnt.h, Date: 04/12/18, Time: 2

```
139     { 0x3c, 0x4a, 0x4a, 0x4a, 0x31}, // 0xe1 - 'б'.
140     { 0x7c, 0x54, 0x54, 0x54, 0x28}, // 0xe2 - 'в'.
141     { 0x7c, 0x04, 0x04, 0x0c, 0x00}, // 0xe3 - 'г'.
142     { 0xc0, 0x78, 0x44, 0x7c, 0xc0}, // 0xe4 - 'д'.
143     { 0x38, 0x54, 0x54, 0x54, 0x08}, // 0xe5 - 'е'.
144     { 0x6c, 0x10, 0x7c, 0x10, 0x6c}, // 0xe6 - 'ж'.
145     { 0x28, 0x44, 0x54, 0x54, 0x28}, // 0xe7 - 'з'.
146     { 0x7c, 0x20, 0x10, 0x08, 0x7c}, // 0xe8 - 'и'.
147     { 0x7c, 0x20, 0x12, 0x0a, 0x7c}, // 0xe9 - 'й'.
148     { 0x7c, 0x10, 0x28, 0x44, 0x00}, // 0xea - 'к'.
149     { 0x40, 0x38, 0x04, 0x04, 0x7c}, // 0xeb - 'л'.
150     { 0x7c, 0x08, 0x10, 0x08, 0x7c}, // 0xec - 'м'.
151     { 0x7c, 0x10, 0x10, 0x10, 0x7c}, // 0xed - 'н'.
152     { 0x38, 0x44, 0x44, 0x44, 0x38}, // 0xee - 'о'.
153     { 0x7c, 0x04, 0x04, 0x04, 0x7c}, // 0xef - 'п'.
154     { 0x7c, 0x24, 0x24, 0x24, 0x18}, // 0xf0 - 'р'.
155     { 0x38, 0x44, 0x44, 0x44, 0x28}, // 0xf1 - 'с'.
156     { 0x04, 0x04, 0x7c, 0x04, 0x04}, // 0xf2 - 'т'.
157     { 0x9c, 0xa0, 0x60, 0x3c, 0x00}, // 0xf2 - 'т'.
158     { 0x18, 0x24, 0x7c, 0x24, 0x18}, // 0xf4 - 'ф'.
159     { 0x6c, 0x10, 0x10, 0x6c, 0x00}, // 0xf5 - 'х'.
160     { 0x7c, 0x40, 0x40, 0x7c, 0xc0}, // 0xf6 - 'ц'.
161     { 0x0c, 0x10, 0x10, 0x10, 0x7c}, // 0xf7 - 'ч'.
162     { 0x7c, 0x40, 0x7c, 0x40, 0x7c}, // 0xf8 - 'ш'.
163     { 0x7c, 0x40, 0x7c, 0x40, 0xfc}, // 0xf9 - 'ш'.
164     { 0x0c, 0x04, 0x7c, 0x50, 0x20}, // 0xfa - 'ъ'.
165     { 0x00, 0x44, 0x7d, 0x40, 0x00}, // 0xfb - 'ы'.
166     { 0x7c, 0x50, 0x50, 0x50, 0x20}, // 0xfc - 'ь'.
167     { 0x28, 0x44, 0x54, 0x54, 0x38}, // 0xfd - 'э'.
168     { 0x7c, 0x10, 0x38, 0x44, 0x38}, // 0xfe - 'ю'.
169     { 0x48, 0x34, 0x14, 0x14, 0x7c} // 0xff - 'я'.
170 };
171
172 #endif /* _NOKIA1100_LCD_FONT_H_ */
```

					ФЗ 8.17100.04.370 ПЗ	Арк
ЗМ.	Лист	№ докум.	Підпис	Дата		94

Додаток Д – функції роботи з температурними сенсорами ds1820

File: C:\cvavr\LIB\ds1820.lib, Date: 04/12/18, Time: 20:43:

```
1      /*
2         CodeVisionAVR C Compiler
3         Dallas Semiconductor DS1820 1 Wire bus temperature sensor functions
4      */
5
6      #include <delay.h>
7      #include <ds1820.h>
8
9      struct __ds1820_scratch_pad_struct __ds1820_scratch_pad;
10
11     unsigned char ds1820_select(unsigned char *addr)
12     {
13     unsigned char i;
14     if (wl_init()==0) return 0;
15     if (addr)
16     {
17         wl_write(0x55);
18         i=0;
19         do
20             wl_write(*(addr++));
21         while (++i<8);
22     }
23     else wl_write(0xcc);
24     return 1;
25     }
26
27     unsigned char ds1820_read_spd(unsigned char *addr)
28     {
29     unsigned char i;
30     unsigned char *p;
31     if (ds1820_select(addr)==0) return 0;
32     wl_write(0xbe);
33     i=0;
34     p=(char *) &__ds1820_scratch_pad;
35     do
36         *(p++)=wl_read();
37     while (++i<9);
38     return !wl_dow_crc8(&__ds1820_scratch_pad,9);
39     }
40
41     int ds1820_temperature_10(unsigned char *addr)
42     {
43     if (ds1820_select(addr)==0) return -9999;
44     wl_write(0x44);
45     delay_ms(550);
46     if (ds1820_read_spd(addr)==0) return -9999;
47     wl_init();
48     return (((int)__ds1820_scratch_pad.temp_msb<<8) |
49             __ds1820_scratch_pad.temp_lsb)*5;
50     }
51
52     unsigned char ds1820_set_alarm(unsigned char *addr, signed char temp_low, signed char
temp_high)
53     {
54     if (ds1820_select(addr)==0) return 0;
55     wl_write(0x4e);
56     wl_write(temp_high);
57     wl_write(temp_low);
58     if (ds1820_read_spd(addr)==0) return 0;
59     if ((__ds1820_scratch_pad.temp_low!=temp_low) ||
60         (__ds1820_scratch_pad.temp_high!=temp_high)) return 0;
61     if (ds1820_select(addr)==0) return 0;
62     wl_write(0x48);
63     delay_ms(15);
64     return wl_init();
65     }
66
```

						ФЗ 8.17100.04.370 ПЗ	Арк
ЗМ.	Лист	№ докум.	Підпис	Дата			95

Клавіатура

Блок
підготовки
живлення

Температурні
сенсори

Блок
керування
струмом

Мікроконтролер

Графічний
дисплей

Блок
вимірювання
струму

Трансформатор 1

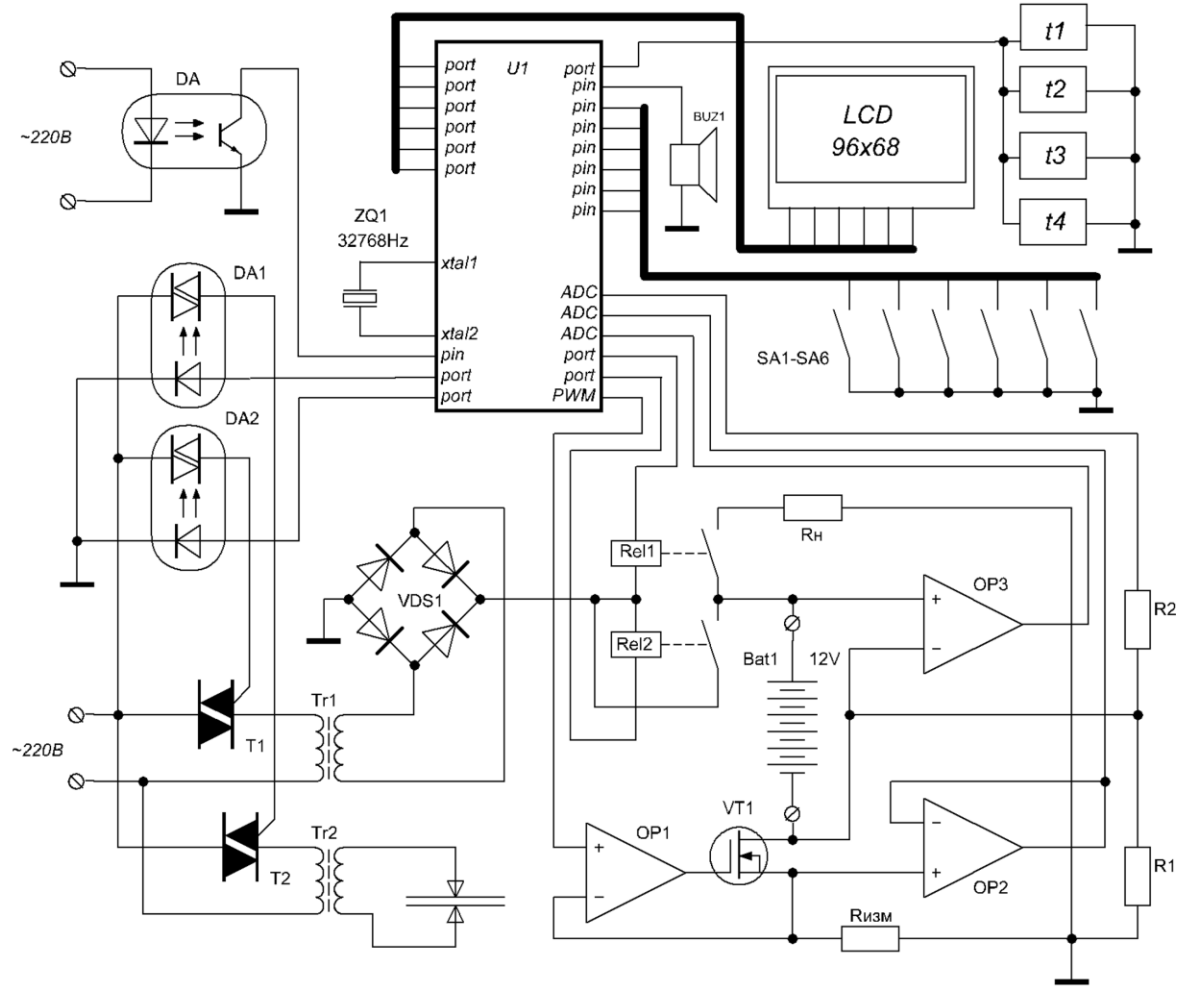
Блок 1 фазового
керування
потужності

Блок
вимірювання
напруги

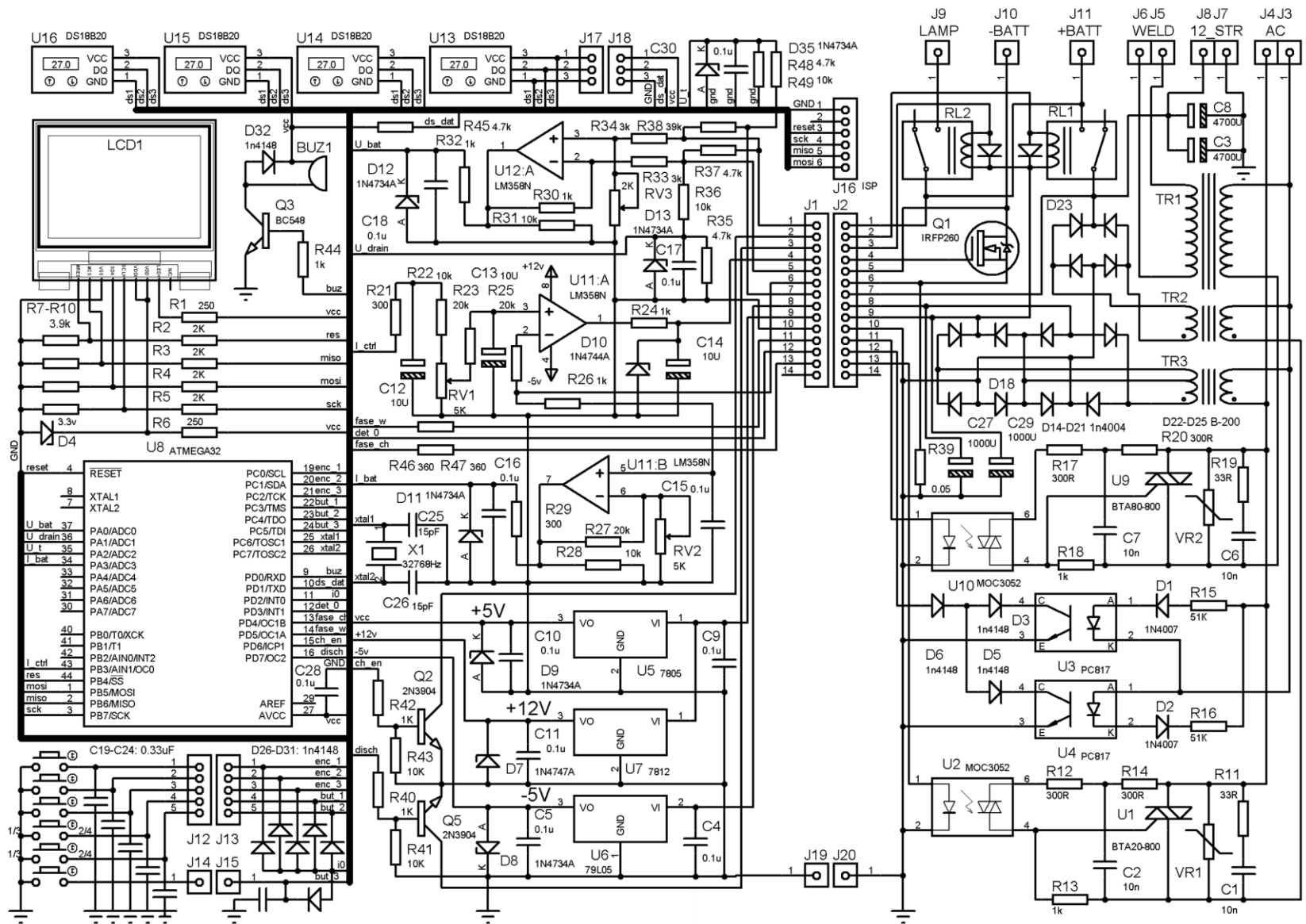
Трансформатор 2

Блок 2 фазового
керування
потужності

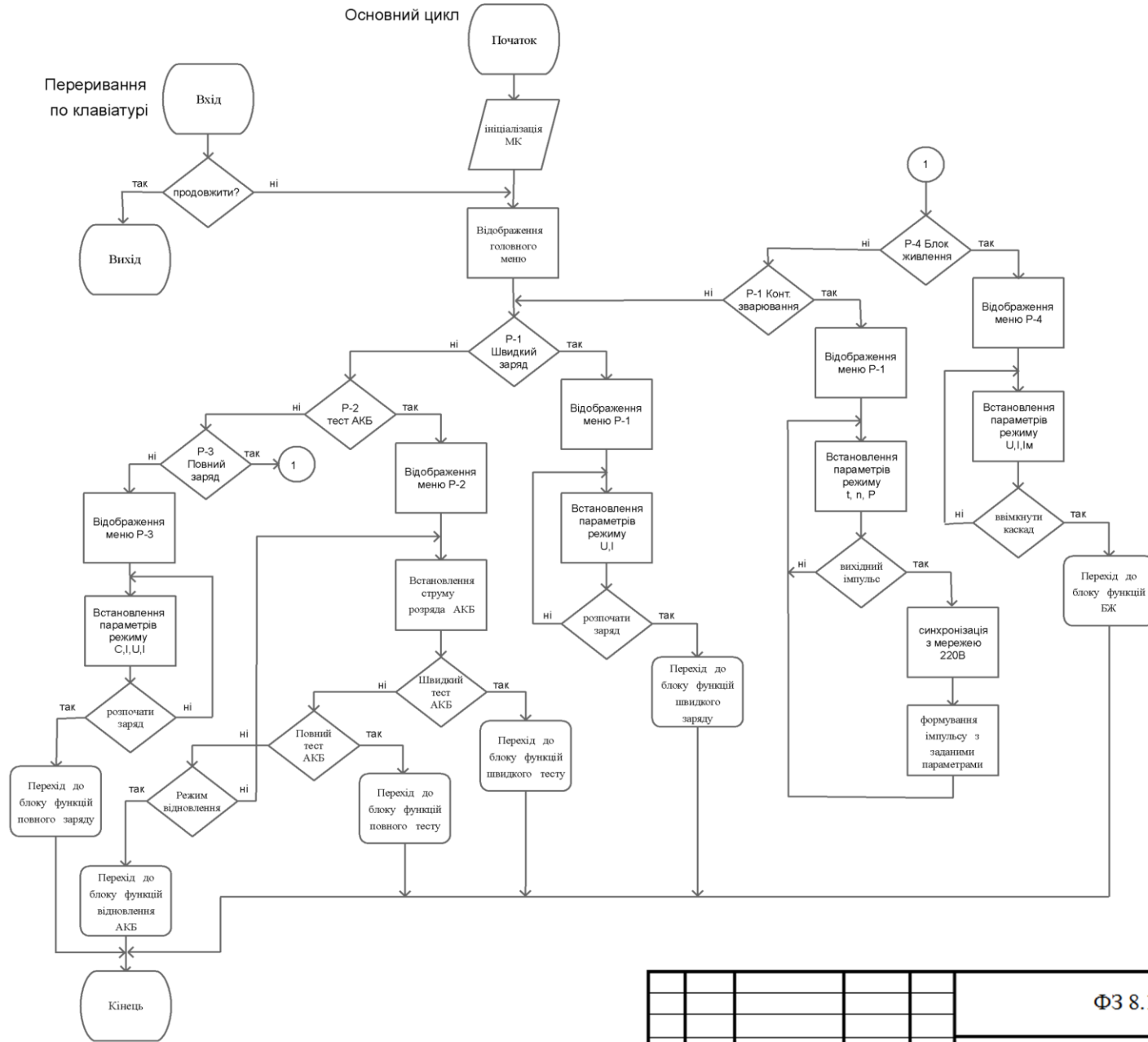
					ФЗ 8.17100.04.370 Е1			
					Система керування електричним струмом та потужністю	Літ.	Маса	Масштаб
Зм.	Арк.	№ докум.	Підпис	Дата				
Розроб.		Гайдар Я.В.						
Перевір.		Горячев О.С.						
Реценз.		Дудник А.Б.						
					Арк.	1	Аркушів	1
Н. Контр.		Гапич В.М.			Структурна схема		СумДУ, КСмЗ-71С	
Затв.		Опанасюк О.С.						



					Ф3 8.17100.04.370 Е2			
Зм.	Арк.	№ докум.	Підпис	Дата	Система керування електричним струмом та потужністю	Літ.	Маса	Масштаб
Розроб.	Гайдар Я.В.							
Перевір.	Горячев О.Є.							
Реценз.	Дудник А.Б.				Арк.	1	Аркушів	1
Н. Контр.	Гапич В.М.				Схема функціональна			СумДУ, КСмз-71С
Затв.	Опанасюк О.С.							

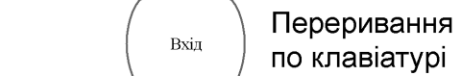
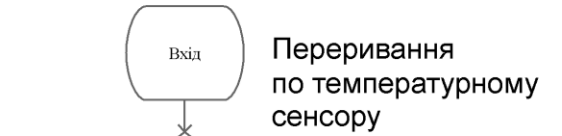
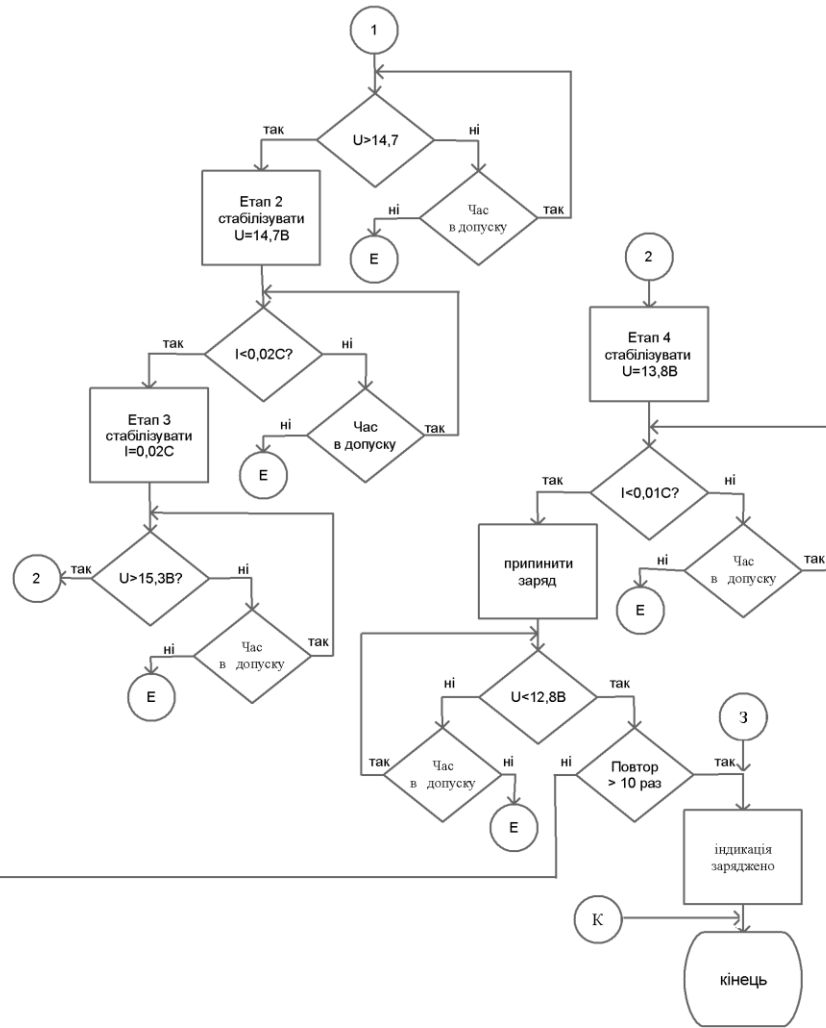
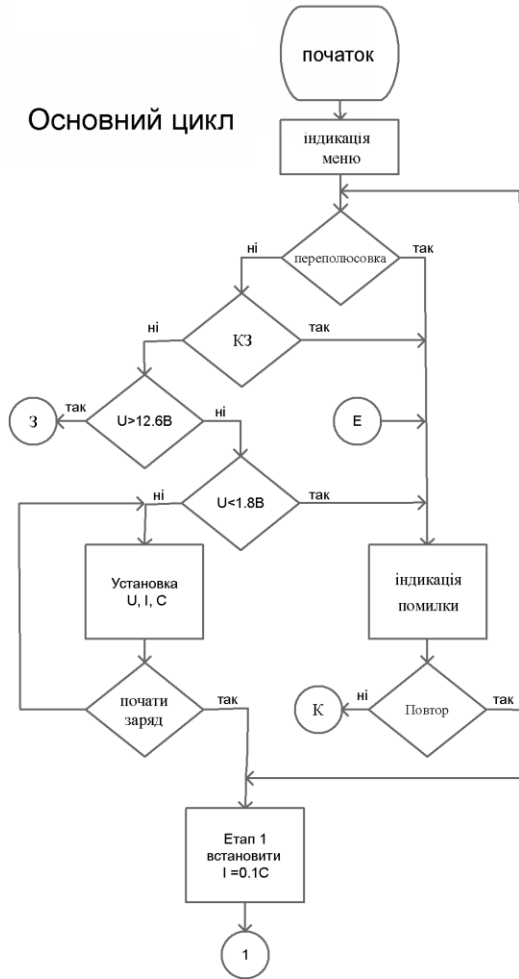


				Ф3 8.17100.04.370 E3				
Зм.	Арк.	№ докум.	Підпис	Дата	Система керування електричним струмом та потужністю	Літ.	Маса	Масштаб
						Арк. 1	Аркушів 1	
Н. Контр.	Гайдар Я.В.				Схема принципова	СумДУ, КСмЗ-71С		
Затв.	Опанасюк О.С.							

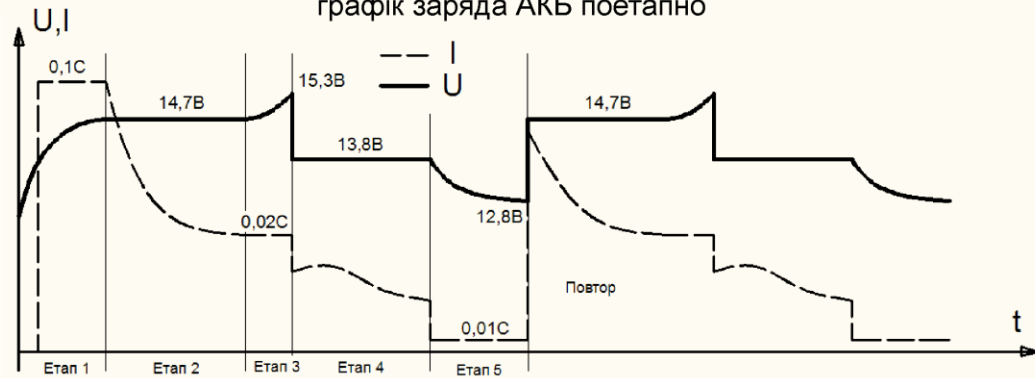


					ФЗ 8.17100.04.370 СА		
					Система керування електричним струмом та потужністю		
					Літ.	Маса	Масштаб
Зм.	Арк.	№ докум.	Підпис	Дата			
Розроб.		Гайдар Я.В.					
Перевір.		Горячев О.Є.					
Реценз.		Дудник А.Б.					
					Арк. 1	Аркушів 3	
					Схема алгоритму		
Н. Контр.		Гапич В.М.			СумДУ, КСмз-71С		
Затв.		Опанасюк О.С.					

Основний цикл

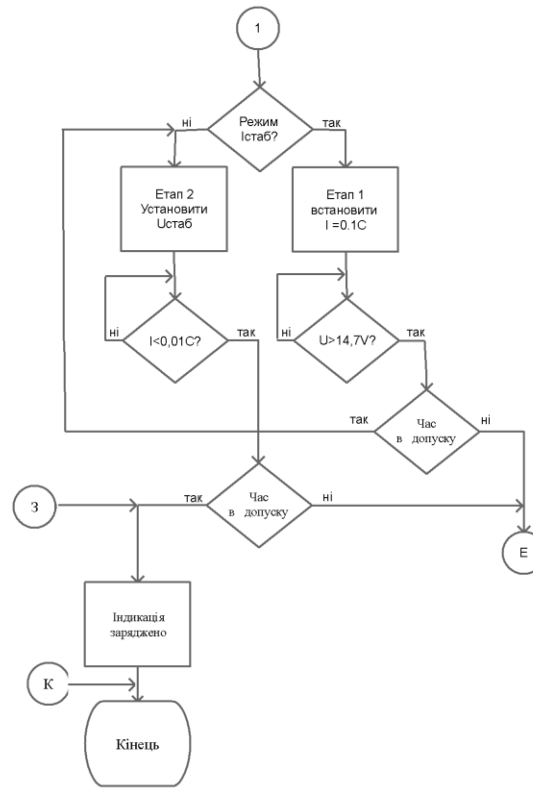
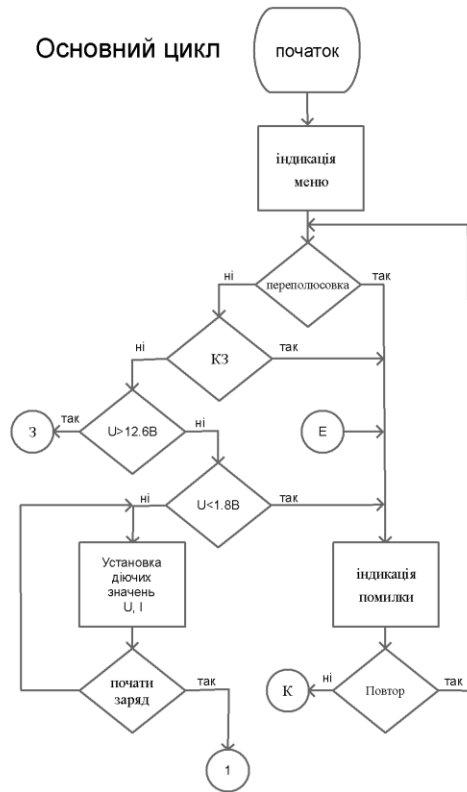


графік заряду АКБ поетапно



					Ф3 8.17100.04.370 СА						
Зм.	Арк.	№ докум.	Підпис	Дата	Система керування електричним струмом та потужністю			Літ.	Маса	Масштаб	
Розроб.	Гайдар Я.В.										
Перевір.	Горячев О.Є.										
Реценз.	Дудник А.Б.				Арк. 2			Аркушів 3			
Н. Контр.	Галич В.М.				Схема алгоритму				СумДУ, КСМЗ-71С		
Затв.	Опанасюк О.С.										

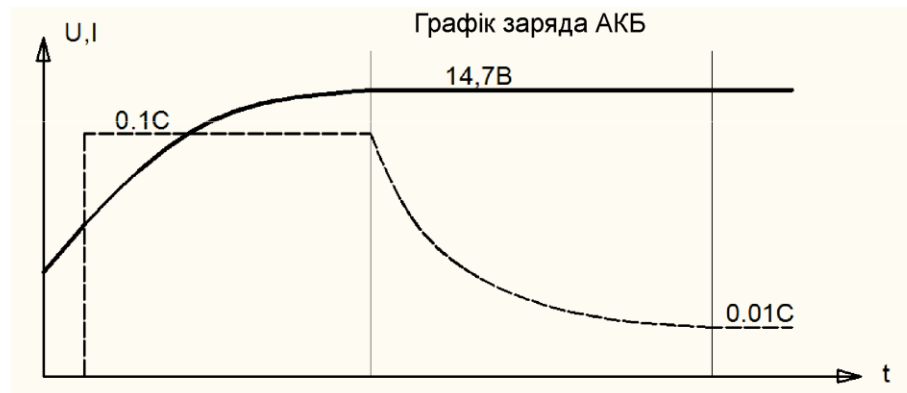
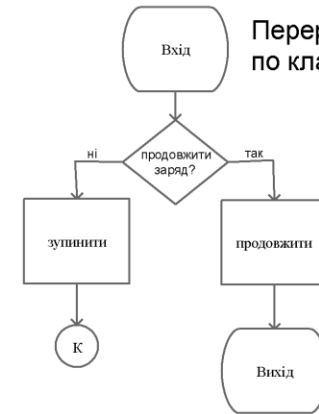
Основний цикл



Переривання по температурному сенсору



Переривання по клавіатурі



					Ф3 8.17100.04.370 СА				
Зм.	Арк.	№ докум.	Підпис	Дата	Система керування електричним струмом та потужністю		Літ.	Маса	Масштаб
Розроб.	Гайдар Я.В.								
Перевір.	Горячев О.Є.								
Реценз.	Дудник А.Б.						Арк. 3	Аркушів 3	
Н. Контр.	Гапич В.М.				Схема алгоритму		СумДУ, КСмз-71С		
Затв.	Опанасюк О.С.								

Позначення	Номинал	Кіл.	Корпус
Конденсатори			
C1-C2,C6-C7	0.01uF	4	CAP90
C3,C8	4700uF-25V	2	CAPPRD750W
C4,C5,C9-C11, C15-C18,C28, C30	0.1uF	11	SMD 1206
C12-C14	10uF-25V	3	CAPPRD200W
C19-C24	0.33uF	6	SMD 1206
C25-C26	15pF	2	SMD 1206
C27,C29	1000uF-25V	2	CAPPRD500W
Резистори			
R1,R6	250	2	SMD 1206
R2-R5	2K	4	SMD 1206
R7-R10	3.9K	4	SMD 1206
R11,R19	33R (3W)	2	RES120
R12,R14,R17,R20	300R (0.6W)	4	RES40
R13,R18	1k (2W)	2	RES90
R15-R16	51K (3W)	2	RES120
R21,R29	300	2	SMD 1206
R22,R28,R31,R36,R49	10k	5	SMD 1206
R23,R25,R27	20k	3	SMD 1206
R24,R26,R30,R32,R44	1k	5	SMD 1206
R33-R34	3k	2	SMD 1206
R35,R37,R45,R48	4.7k	4	SMD 1206
R38	39k	1	SMD 1206

					ФЗ 8.17100.04.370 ПЕ		
Зм.	Арк.	№ докум.	Підпис	Дата			
Розроб.		Гайдар Я.В.			Система керування електричним струмом та потужністю Перелік елементів		
Перевір.		Горячев О.Є.					
Реценз.							
Н. Контр.		Гапич В.М.					
Затв.		Опанасюк О.С.					
					Літ.	Арк.	Аркушів
						1	3
					СумДУ, КСмз-71С		

R39	0.05 (10W)	1	RES180
R40,R42	1K	2	SMD 1206
R41,R43	10K	2	SMD 1206
R46-R47	360	2	SMD 1206
Мікросхеми			
U1	BTA20-800	1	TO254
U2,U10	МОС3052	2	DIL06
U3-U4	PC817	2	DIL04
U5	7805	1	TO220
U6	79L05	1	TO92
U7	7812	1	TO220
U8	ATMEGA32	1	TQFP44
U9	BTA80-800	1	TO254
U11-U12	LM358N	2	DIL08
U13-U16	DS18B20	4	TO92
Транзистори			
Q1	IRFP260	1	TO247
Q2,Q5	2N3904	2	TO92
Q3	BC548	1	TO92
Діоди			
D1-D2	1N4007	2	DO41
D3,D5-D6	1n4148	3	DO35
D4	1N4728A	1	SMB
D7	1N4747A	1	DO41
D8-D9,D11-D13,D35	1N4734A	6	DO41
D10	1N4744A	1	DO41
D14-D21, D33-D34	1N4004	10	DO41
D22-D25,	B-200	4	NULL
D26-D32	1n4148	7	SMB

Дисплей			
LCD1	LCD_nokia1280	1	
Інше			
X1	32768Hz	1	
RL1-RL2	RLY-NTE-46	2	TEXTELL KB REL
But1-But3		3	Button_dil
EN1	encoder	1	
BUZ1	buzzer	1	
L1-L3	100uH	3	INDC4532X175
RV1-RV2	5K	2	5K OHM 1/2 inch SQ
RV3	2K	1	2K OHM 1/2 inch SQ
VR1-VR2	S20K420	2	VAR-S20
TR1	8kVA	1	
TR2	1.2kVA	1	