

Вінницький національний технічний університет
(повне найменування вищого навчального закладу)
Факультет Інформаційних електронних систем
(повне найменування інституту, назва факультету (відділення))
Кафедра Інформаційних радіоелектронних технологій і систем
(повна назва кафедри (предметної, циклової комісії))

ПОЯСНЮВАЛЬНА ЗАПИСКА

до бакалаврської дипломної роботи на тему:

«РОЗРОБКА СИСТЕМИ АВТОМАТИЧНОГО РЕГУЛЮВАННЯ ПІДСИЛЕННЯ ШИРОКОСМУГОВОГО ЦИФРОВОГО ПРИЙМАЧА»

Виконав: студент 2 (4)-го курсу, групи ТКР-22мс,
спеціальності 172 – Телекомунікації та радіотехніка
(шифр і назва напрямку підготовки, спеціальності)

Чубатюк Ю.С.

(прізвище та ініціали)

Керівник: к.т.н., доц. каф. ІРТС

Савицький А.Ю

(прізвище та ініціали)

« 13 » 08 2024 р.

Рецензент: д.т.н. доц., проф. каф. ІКСТ

Михайлевський Д.В.

(прізвище та ініціали)

« 13 » 08 2024 р.

Допущено до захисту

Завідувач кафедри ІРТС

д.т.н., проф. Осадчук О.В.

(прізвище та ініціали)

« 14 » 08 2024 р.

Вінниця ВНТУ - 2024 рік

Вінницький національний технічний університет
Факультет Інформаційних електронних систем
Кафедра Інформаційних радіоелектронних технологій і систем
Рівень вищої освіти перший (бакалаврський)
Галузь знань – 17 Електроніка та телекомунікації
Спеціальності 172 – Телекомунікації та радіотехніка
Освітньо-професійна програма – Радіотехніка

ЗАТВЕРДЖУЮ
Завідувач кафедри ІРТС
Олександр ОСАДЧУК
15.03. 2024 року

ЗАВДАННЯ НА БАКАЛАВРСЬКУ ДИПЛОМНУ РОБОТУ СТУДЕНТУ

Чубатюку Юрію Сергійовичу
(прізвище, ім'я, по батькові)

- Тема роботи «Розробка системи автоматичного регулювання підсилення широкосмугового цифрового приймача»
керівник роботи Савицький Антон Юрійович, к.т.н., доц. каф. ІРТС
(прізвище, ім'я, по батькові, науковий ступінь, вчене звання)
затверджені наказом вищого навчального закладу від “11”03 2024 року № 80
- Строк подання студентом роботи 07 06 2024 року.
- Вихідні дані до роботи: напруга живлення 12 В, вихідна потужність 4 Вт, опір навантаження 8 Ом, рівень вхідного сигналу 150 мВ, коефіцієнт підсилення 20дБ, коефіцієнт спотворень 7%, струм споживання 0,2 А.
- Зміст текстової частини: Вступ. Обґрунтування теми бакалаврської дипломної роботи. Вибір та обґрунтування схеми пристрою. Електричні розрахунки. Розробка технологічного процесу налаштування пристрою. Основні несправності та методи їх усунення. Основні несправності та методи їх усунення. Розробка програмного забезпечення. Охорона праці. Висновки. Список використаних джерел. Додатки.
- Перелік ілюстративного матеріалу (з точним зазначенням обов'язкових креслень): Блок схема ATMEGA8. Схема електрична принципова. Схема підключення USB за допомогою UART. Алгоритм роботи програми. Меню програми Microchip Studio. Принципова схема стабілізатора напруги. Результати моделювання кварцового генератора в програмі Proteus. Схема блоку реле. ВХ транзистора IRLML2502. Налаштування ф'юзів ATMEGA8 для правильної роботи програми. Вікно програми, та налаштування мови. Контекстне меню програми.

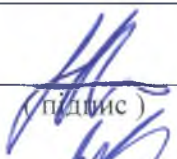
6. Консультанти розділів роботи

Розділ	Прізвище, ініціали та посада консультанта	Підпис, дата	
		завдання видав	виконання прийняв
Основна частина	к.т.н., доцент каф. ІРТС Савицький А.Ю.		
Охорона праці	професор кафедри БЖДПБ, професор, д.п.н., Дембіцька С.В.		

7. Дата видачі завдання 12.03. 202 4 року

КАЛЕНДАРНИЙ ПЛАН

№ з/п	Назва етапів бакалаврської дипломної роботи	Строк виконання етапів роботи	Примітка
1.	Вибір, узгодження та затвердження теми БДР	14.02.2024-28.02.2024	
2.	Огляд та аналіз літературних джерел.	01.03.2024-10.03.2024	
3.	Затвердження теми. Розробка завдання на БДР.	11.03.2024-31.03.2024	
4.	Попередня розробка основних розділів. Аналіз вирішення поставленої задачі. Розробка структурної схеми та технічних рішень.	01.04.2024-06.05.2024	
5.	Математичне моделювання та електричні розрахунки. Експериментальне дослідження.	07.05.2024-18.05.2024	
6.	Розробка графічної частини БДР	19.05.2024-22.05.2024	
7.	Охорона праці (ОП)	23.05.2024-28.05.2024	
8.	Оформлення пояснювальної записки та графічної частини.	29.05.2024-05.06.2024	
9.	Нормоконтроль	06.06.2024-07.06.2024	
10.	Попередній захист БДР, доопрацювання, рецензування БДР	08.06.2024-10.06.2024	
11.	Захист БДР ЕК	11.06.2024-17.06.2024	

Студент  Чубатюк Ю.С.Керівник роботи  Савицький А.Ю.

АНОТАЦІЯ

УДК 621.316

Чубатюк Ю.С. Розробка системи автоматичного регулювання підсилення широкосмугового цифрового приймача. – Вінниця: ВНТУ, 2024. – 91 с. Українською мовою. Бібліогр.: 20 с. ; Рис.: 22; Таб.: 11

Дипломний проект передбачає розробку системи автоматичного регулювання підсилення для отримання вихідної потужності 4 Вт, за умови рівню вхідного сигналу 150 мВ з коефіцієнтом підсилення 20дБ та коефіцієнтом спотворень не більше 7%.

У загальній частині дипломної бакалаврської роботи розглянуто особливості плати ATMEGA8 та принцип роботи схеми, особливість її виконання.

У розрахунково-конструкторській частини розроблено структурну схему розетки, розрахунок надійності.

У технологічній частині розроблений технологічний процес регулювання схеми пристрою та наводяться основні несправності та методи їх усунення.

Також розглянуті заходи по охороні праці і техніки безпеки. В графічній частині представлено такі розділи: схема електрична принципова, складальне креслення.

Ключові слова: широкосмуговий приймач, регулювання підсилення, автоматизація процесу, підсилення, цифровий.

ABSTRACT

Yu.S. Chubatyuk Development of a system of automatic gain adjustment of a broadband digital receiver. – Vinnytsia: VNTU, 2024. – 91 p. In ukrainian. Bibliography: 20 p. ; Fig.: 22; Tab.: 11

The diploma project involves the development of an automatic gain control system to obtain an output power of 4 W, provided that the input signal level is 150 mV with a gain factor of 20 dB and a distortion factor of no more than 7%.

In the general part of the diploma bachelor's work, the features of the ATMEGA8 board and the principle of operation of the scheme, the peculiarities of its implementation are considered.

In the calculation and design part, a structural diagram of the outlet, reliability calculation was developed.

In the technological part, the technological process of regulating the device circuit is developed, and the main malfunctions and methods of their elimination are given.

Measures for occupational health and safety were also considered. The following sections are presented in the graphic part: basic electrical diagram, assembly drawing.

Keywords: broadband receiver, gain control, process automation, gain, digital.

ЗМІСТ

ВСТУП.....	7
1 ОБГРУНТУВАННЯ ТЕМИ БАКАЛАВРСЬКОЇ ДИПЛОМНОЇ РОБОТИ.....	11
1.1 Вибір та обґрунтування схеми пристрою.....	11
1.2 Електричні розрахунки.....	25
1.3 Розрахунок надійності.....	32
2 ТЕХНОЛОГІЧНА ЧАСТИНА	36
2.1 Розробка технологічного процесу настройки пристрою.....	36
2.2 Основні несправності та методи їх усунення.....	39
3 РОЗРОБКА ПРОГРАМНОГО ЗАБЕЗПЕЧЕННЯ.....	40
4 ОХОРОНА ПРАЦІ.....	44
4.1 Технічні рішення щодо безпечного виконання роботи.....	44
4.2 Технічні рішення з гігієни праці та виробничої санітарії.....	47
4.3 Пожежна безпека.....	52
ВИСНОВОК.....	55
СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ.....	56
Додаток А (обов'язковий) Ілюстративний матеріал.....	59
Додаток Б (обов'язковий) Протокол перевірки роботи	64
Додаток В (довідниковий) Лістинг програми.....	66

ВСТУП

Актуальність теми. Сучасне суспільство повністю тримається на електроенергії тому зарядні пристрої стали невід'ємною частиною нашого життя. Прикро коли твоє зарядне гарно працює, але батарея вже занадто виснажена. Ноутбуки використовуються як переносні комп'ютери, але деколи, коли вони стоять стаціонарно, як звичайний ПК, то ресурс батареї втрачається значно швидше, що згодом призводить до того, що батарея тримає заряд півгодини, але зовсім небажано.

Розробка системи автоматичного регулювання підсилення широкосмугового цифрового приймача.

Починаючи з того, що таке зарядний пристрій – це пристрій створений для заряду акумулятора (зберігає енергію). Потрібно спочатку розповісти що таке акумулятор, його принцип дії та основні різновиди, потім повернувшись до самого зарядного пристрою.

Акумулятор - це пристрій, який виробляє електрони за допомогою електрохімічних реакцій і містить позитивні (+) та негативні (-) термінали. Акумулятор складається з однієї або декількох електрохімічних комірок, які перетворюють накопичену хімічну енергію безпосередньо в електричну енергію. Коли зовнішнє навантаження підключається до акумулятора, електрони переходять з негативного на позитивний термінал, створюючи електричний струм. Цей струм може живити двигун, лампочку, годинник, комп'ютер, мобільний телефон та інші електронні пристрої чи обладнання. Швидкість потоку акумулятора визначається внутрішнім опором акумулятора та зовнішнім навантаженням.

Акумуляторна батарея може бути виконана із декількох електрохімічних комірок, об'єднаних в один електричний ланцюг. Ці комірки сполучені електрично і конструктивно для отримання необхідних значень струму і напруги. Використовується, зокрема, як джерело енергії для живлення тягових електродвигунів акумуляторних електровозів. Основні параметри, які

характеризують такий акумулятор, — електрорушійна сила, напруга, внутрішній опір, струм та ємність.

В акумуляторах глибокого заряду-розряду (поїздів, човнів, автотранспорту), автомобільних акумуляторах (забезпечують постійну подачу струму протягом тривалого періоду часу) енергію виробляють елементи — група свинцевих пластин покритих окисом свинцю і кислотою

Незважаючи на те, що щільність енергії становить лише близько однієї третини аналогічного показника літєвої батареї, специфічною властивістю нікель-водневого акумулятора є тривалість терміну експлуатації: осередки витримують більш ніж 20 000 циклів розряду при 85% ефективності.

Зарядний пристрій — пристрій для заряду електричних акумуляторів енергією зовнішнього джерела, зазвичай від мережі змінного струму напругою 100–240 Вольт. Включає в себе перетворювач напруги (трансформатор, імпульсний блок живлення), випрямляч, стабілізатор напруги, пристрій контролю сили струму або процесу заряду, амперметр або світлодіодні індикатори. Характеристики зарядних пристроїв залежать від типу акумуляторів, робочої напруги, номінальної ємності. Зарядні пристрої можуть бути вбудованими та зовнішніми.

Іноді в зарядних пристроях використовуються імпульсні блоки живлення, а при стаціонарному використанні ноутбука з таким типом живлення, цей блок деколи працює в режимі холостого ходу, що згодом починає негативно впливати на нього. Представлений пристрій створений щоб вмикати та вимикати зарядне аби створити повні цикли заряду/розряду у вибраних межах, що запобігає швидкому погіршенню тривалості роботи акумулятора без підзарядки.

Модулем керування системи автоматичного регулювання підсилення є мікросхема компанії AVR ATMEGA8. Його взаємозв'язок з комп'ютером виконаний за допомогою кабелю USB та інтерфейс до нього зроблений програмно.

Система автоматичного регулювання підсилення широкосмугового цифрового приймача – це, насамперед, блок керування цього пристрою, роль якого забезпечити підсилення приймача.

Пристрій не впливає на саму роботу приймача, а лише регулює її. Саме через такий принцип роботи зберігається циклічність процесу зарядки, що при стаціонарному використанні ноутбука втрачається, це впливає працездатність батареї та її ресурс.

Дивлячись на те, що прогрес не стоїть на місці, деякі сучасні зарядні пристрої мають змогу зупинятись на певному рівні заряду та не перезаряджати його. Наприклад адаптер живлення «Lenovo ThinkPad Slim 135W AC Adapter Slim tip». Даний адаптер має функцію призупиняти заряджання батареї ноутбука на рівні 95-97%. Особливістю саме проектованого пристрою є виконання циклів зарядки, розрядки, що не передбачено зарядним пристроєм з прикладу.

Ноутбуком користується велика кількість людей, через його малі габарити, та гарні показники. Обираючи ноутбук люди шукають більш компактний пристрій, аніж потужний. Ноутбуки є як бюджетного варіанту, так і більш потужні – це означає, що вони будуть імпонувати будь-кому.

Переваги системи автоматичного регулювання підсилення широкосмугового цифрового приймача в точності та рівномірності підсилення, що буде автоматично вирівнюватися.

Враховуючи вище приведений матеріал можна вважати, що система автоматичного регулювання підсилення широкосмугового цифрового приймача доцільна.

Метою роботи є розробка пристрою, що буде відтворювати цикли зарядки/розрядки батареї ноутбука, отримуючи інформацію про рівень заряду безпосередньо від самого пристрою, що заряджається.

Об'єктом дослідження є процес знайдення та втілення системи автоматичного регулювання підсилення широкосмугового цифрового приймача, навіть після тривалого використання його буде надавати стабільне та високе підсилення.

Предметом дослідження є автоматизація процесу регулювання підсилення широкопasmового цифрового приймача.

Для досягнення поставленої мети в бакалаврській дипломній роботі розв'язуються наступні задачі:

- створення загальної структурної системи приладу;
- вибір мікроконтролера та створення схеми;
- побудування алгоритму роботи програми;
- написання програми;
- аналіз, розрахунок та підбір компонентів;
- експериментальне дослідження працездатності написаного програмного забезпечення.

Методи дослідження ґрунтуються на використанні: математичних розрахунків компонентів, а також проведення запуску програмного забезпечення з метою виявлення проблем з кодом та загального розуміння працездатності додатку.

Наукова новизна одержаних результатів отримання подальший розвиток у програмуванні мікроконтролерів, вдосконалення широкопasmових цифрових приймачів.

Практичне значення одержаних результатів економія ресурсу батареї за для досягнення біль довготривалого часу роботи без під'єднання до мережі.

Структура і обсяг роботи. Бакалаврська дипломна робота складається зі вступу, 4 розділів, висновків, додатків та списку використаних джерел.

1 ОБГРУНТУВАННЯ ТЕМИ БАКАЛАВРСЬКОЇ ДИПЛОМНОЇ РОБОТИ

1.1 Вибір та обґрунтування схеми пристрою

У даному розділі аналізується поставлене завдання та відповідно до критеріїв обирається мікроконтролер для його реалізації. Проводиться розробка структурно-функціональної схеми пристрою та коротко надається опис її роботи. Далі описуються додаткові елементи, які використовуються в даній схемі та наводяться їх характеристики.

1.1.1 Вибір та обґрунтування структурної схеми

Індивідуальне завдання передбачає виконання системи автоматичного регулювання підсилення, що повинна створювати на виході підсилення потужності 4 Вт, за умови рівню вхідного сигналу 150 мВ з коефіцієнтом підсилення 20дБ та коефіцієнтом спотворень не більше 7%.

Якщо потрібно виконувати складні дії, то для цього потрібен мікроконтролер, вибір пав на ATMEGA8 через його універсальність та практичність. Також потрібний кабель з'єднання з мережею та кабель USB.

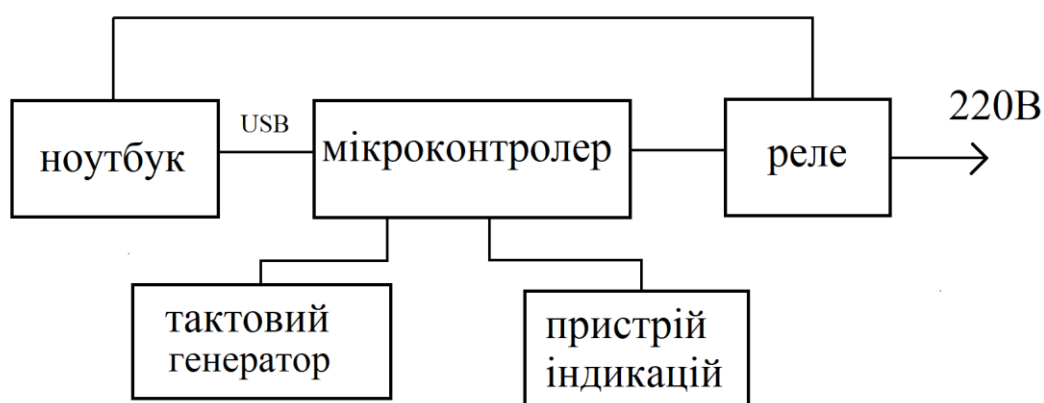


Рисунок 1.1 – Структурна схема

На рисунку 1.1 зображено структурну схему приладу. Основним та єдиним елементом керування є мікроконтролер який здійснює приймання та

передавання сигналу від комп'ютера до реле, що вмикає та вимикає зарядний пристрій.

1.1.2 Вибір мікроконтролера

Блок керування реалізований на ATMEGA8. Програма потребує Flash пам'ять для реалізації інтерфейсу USB у мікроконтролері, через це була обрана мікросхема ATMEGA8 так, як вона має 8Кбай Flash пам'яті. Його данні приведені в таблиці 1.1.

Таблиця 1.1 – Технічні характеристики

Технічні характеристики	Параметри
Бітність процесора	8біт
Flash пам'ять	8Кбайт
EEPROM	256байт
SRAM	1024байт
Час збереження даних	20років при тепературі 80°C
8бітний таймер/секундомір	2
12бітний таймер/секундомір	1
ШИМ канали	3
Вхід АЦП	8-каналів 10біт
Робоча частота	0-16МГц
Живлення	2,7 - 5,5V
Робоча температура	-40 - +80°C

AVR – компанія восьмибітних мікроконтролерів фірми Atmel. Саме ця компанія створила ATMEGA8. Мікроконтролери AVR мають гарвардську

архітектуру (програма та дані розташовані в різних адресних просторах) та систему команд, близьку до ідеології RISC. Процесори AVR мають 32 8-бітові реєстри загального призначення.

Аналого-цифрові перетворювачі – це пристрої, які приймають вхідні аналогові сигнали та генерують відповідні до них цифрові сигнали, які придатні для обробки мікропроцесорами та іншими цифровими пристроями.

Блок схема цього мікроконтролера на рисунку 1.2.

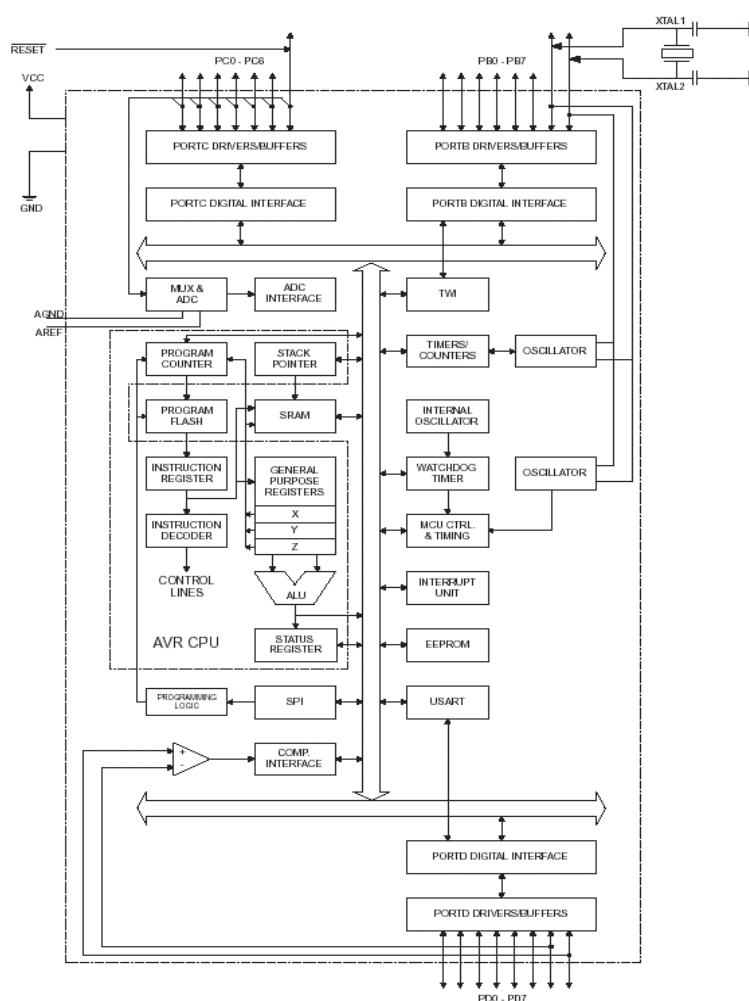


Рисунок 1.2 – Блок схема ATMEGA8

Принципово не виключена можливість безпосереднього перетворення різних фізичних величин в цифрову форму, однак це завдання вдається розв'язати, але досить рідко через складність таких перетворювачів. Тому зараз найраціональнішим вважається спосіб перетворення різних за фізичною природою величин спочатку в функціонально пов'язані з ними електричні, а потім

уже за допомогою перетворювачів напруга - код – в цифрові. Саме ці перетворювачі і мають на увазі, коли говорять про аналого – цифрові перетворювачі.

Робота з периферійними пристроями здійснюється через адресний простір даних. Для зручності з першими 64 адресами периферійних пристроїв можна працювати за допомогою «скорочених» команд IN/OUT доступу до простору вводу-виводу (I/O — Input/Output). Роботу з окремими бітами периферійних пристроїв у просторі I/O забезпечують команди SBI, CBI, SBIS, SBIC. Втім, ці команди працюють лише з першими 32-ма адресами простору. Зовнішній вигляд згори мікропроцесора ATMEGA8 у MLF корпусі зображена на рисунку 1.3.

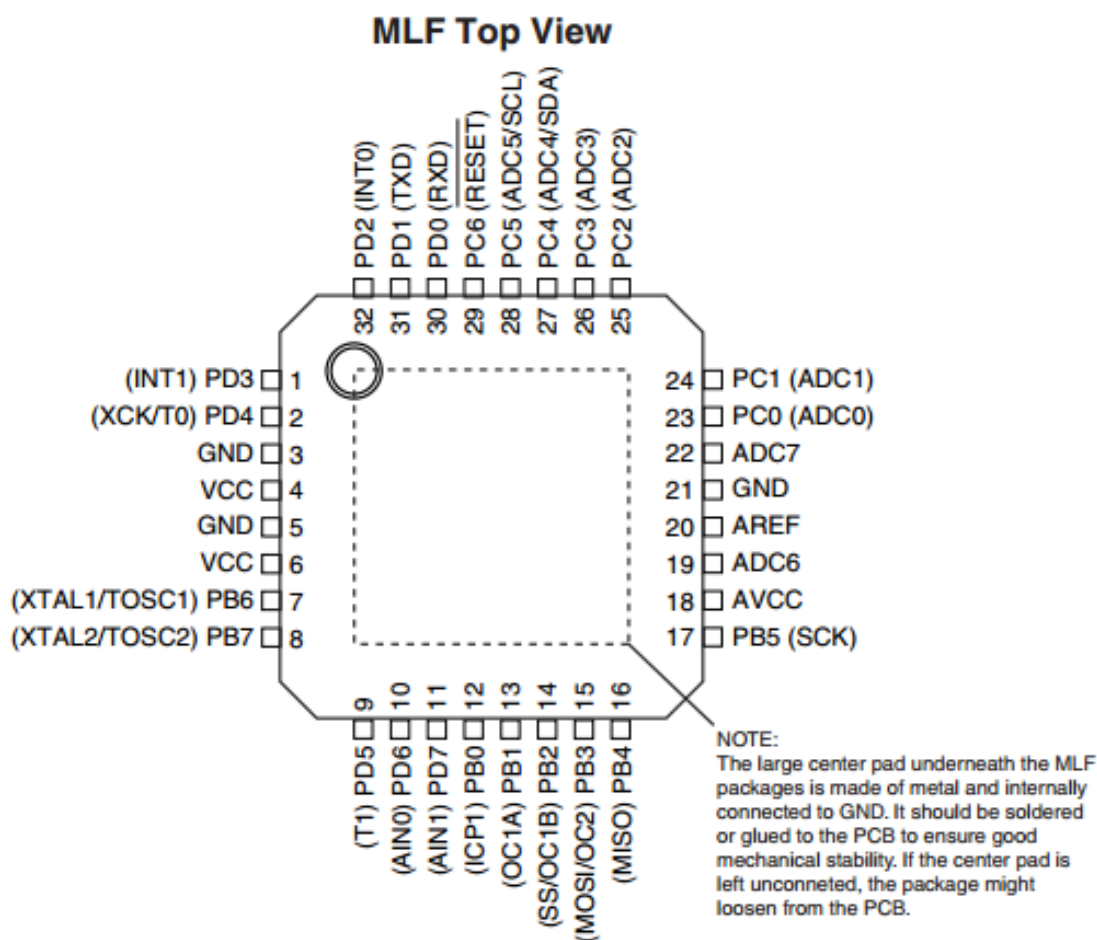


Рисунок 1.3 – Призначення виводів ATMEGA8 в корпусі типу MLF

1.1.3 Розробка схеми пристрою

На рисунку 1.4 зображено схему електричну принципову приладу. Живлення на мікросхему ATMEGA8 подається через порт USB, але не на пряму, а через стабілізатор напруги AMS1117-3.3 також для запобігання паразитних напруг було використано по 2 конденсатора з обох боків. Один електролітичний для замкнення загальних завад, а другий керамічний для замкнення завад першого конденсатора. Живлення підключене на 4-ий та 6-ий (VCC) вивід. Було обрано саме цей стабілізатор через те, що він має мале падіння напруги, а саме 1,1В. Він був обраний із серії ASM1117, на вихідну напругу 3,3В з гарною точністю до 0,05 В похибки. Цей вольтаж вибраний з урахуванням узгодження напруги живлення та напруги сигнальних виводів USB D-, D+(3,3В).

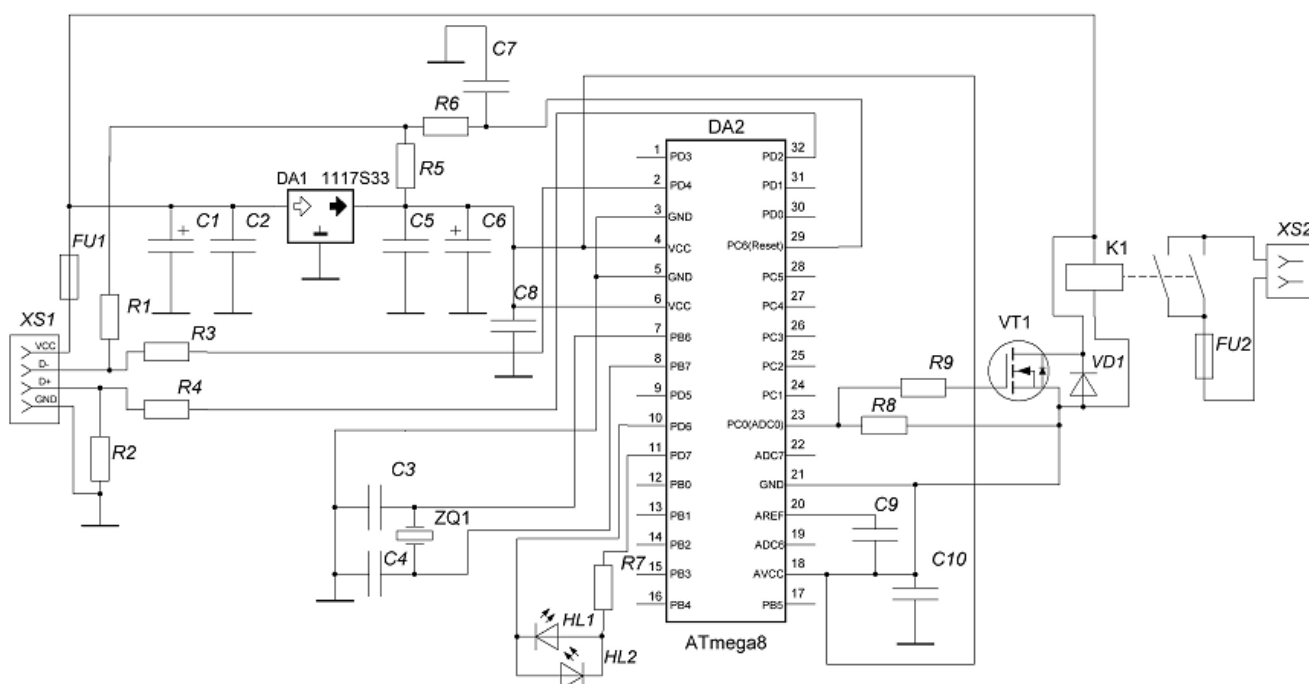


Рисунок 1.4 – Схема електрична принципова

Сигнальні виводи під'єднані до 2-ого(PD4) та 32-ого(PD1) виводів. за допомогою них відтворюється обмін інформацією, та програма розуміє цей пристрій за допомогою HID дескриптора який бачить його як нестандартний пристрій. Встановлений режим низькошвидкісної передачі по USB. А резистори R3 та R4 обрані саме на 68Ом обмежують амплітуду сигналу з виходів D- D+ до

рівня 3,3В. А високошвидкісний резистор R2 на 1МОм забезпечує розрядку кабелю USB.

Вивід 29(Reset) з'єднаний з виводом D- USB, за допомогою R5 та стабілізатора DA1 на ньому вирівнюється амплітуда сигналів до рівня 3,3В, щоб не було моменту при якому подається сигнал Reset, але його МК вважає шумом, через малу амплітуду сигналу.

Тактовий генератор зроблений з частотою 12МГц можливо було взяти ще на 15МГц, але вибір пав на інший варіант. 16МГц та інші не підходять через те, що бібліотека та драйвера працюють лише при 12 або 15МГц. Генератор підключений до виводів 7-8(XTAL1, XTAL2), заземлений на виводи 3, 5(GND) та формує тактові імпульси для МК.

До виводів 9-10(PD6, PD7) під'єднані світлодіоди, які мають роль індикації 3 блимає зелений – старт роботи; горить зелений – зарядний пристрій працює; горить червоний – зарядний пристрій не працює, а також повідомлення, що сталася помилка.

Вивід PC0 працює як звичайний вивід порту C. До цього виводу підключений затвор транзистора VT1, а до витоку через фільтруючі конденсатори підключені: опорна напруга мікросхеми, живлення аналого-цифрового перетворювача, відповідно виводи: 20(AREF), 18(AVCC) та все це заземлено на вивід 21(GND). До стоку транзистора VD1 під'єднаний анод діода, який з'єднаний з реле, це потрібно, щоб при вимкненні подачі струму на котушку реле енергія з неї приймалася діодом та не шкодила елементам схеми. До реле під'єднаний запобіжник на 1 А, його роль забезпечити захист схеми від стрибків струму у мережі.

Електромагнітне реле – це електромеханічний пристрій, що реагує на зміну параметра системи розмиканням або замиканням контактів, основним завданням якого є виконання комутаційних операцій в електричних ланцюгах. Спрацьовування може виконуватися під впливом таких факторів, як електричний струм, тиск або рівень рідини, світлова енергія. Для цього було обране реле

845HN-2A-C-05VDC. Він пропускає 8А та 240В, що підходить для майже будь-якого ноутбука.

1.1.4 Розробка інтерфейсу USB

Інтерфейс USB (Universal Serial Bus - Універсальний Послідовний Інтерфейс) призначений для підключення периферійних пристроїв до персонального комп'ютера. Дозволяє проводити обмін інформацією з периферійними пристроями на трьох швидкостях (специфікація *USB 2.0*):

- низька швидкість (*Low Speed - LS*) - 1,5 Мбит/с;
- повна швидкість (*Full Speed - FS*) - 12 Мбит/с;
- Висока швидкість (*High Speed - HS*) - 480 Мбит/с

За допомогою мікроконтролера, у якого інтерфейс USB реалізований апаратно. У цьому випадку необхідно знати, як працює USB і відповідно з цим написати програму для мікроконтролера. Крім того, якщо операційна система не підтримує стандартні класи USB, то необхідно написати драйвер для комп'ютера. Основний недолік даного способу - обмежена доступність таких мікроконтролерів і їх більш висока вартість в порівнянні з звичайними мікроконтролерами, які використовуються для зв'язку через "RS232".

Якщо брати декілька основних інтерфейсів USB, наприклад для плат Arduino, які використовують мікроконтролери компанії AVR, тоді це будуть UART, SPI, I2C. Більш детально можна розглянути UART. Це перший, розповсюджений та більш простий інтерфейс, з нього і починається вивчення інтерфейсів. Його використовують в будь якому програмуванні так як за допомогою UART інтерфейсу відбувається зв'язок плати з комп'ютером для прошивки. Завдяки спеціальному вбудованому адаптору він перетворюється в більш зрозумілий для комп'ютера порт USB. UART – Universal Asynchronous Receiver-Transmitter, що в переводі означає універсальний асинхронний приймач-передавач, також є його синхронний варіант USART Схеми підключення якого зображена на рисунку 1.5.

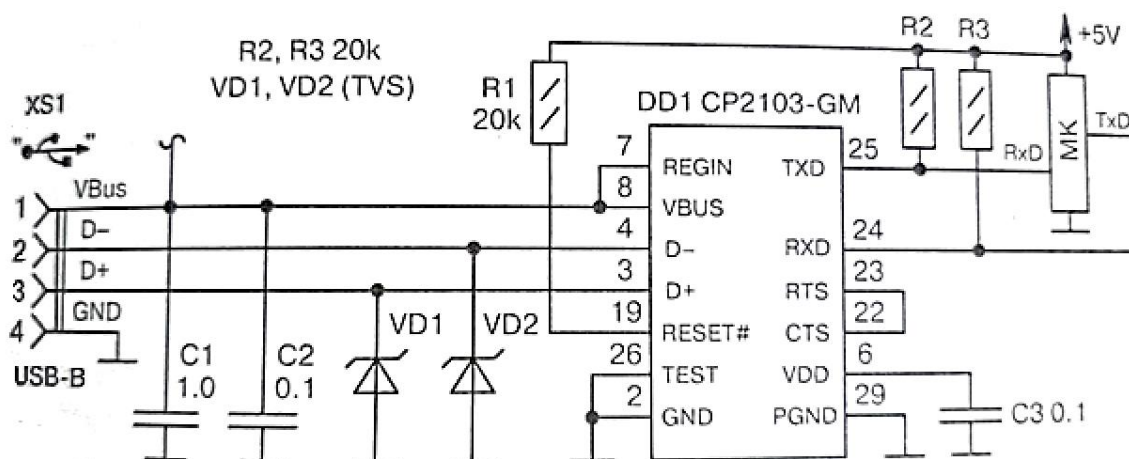


Рисунок 1.5 – Схема підключення USB за допомогою UART

Використання універсального перетворювача інтерфейсів: USB і будь-якого іншого. В якості іншого інтерфейсу зазвичай використовується RS232, 8-розрядна шина даних або шина TWI. У цьому випадку розробка спеціальної прошивки не буде потрібно, немає потреби знати, як працює USB і немає необхідності написання драйвера, тому що виробник перетворювача пропонує свій драйвер. Недолік - більш висока вартість завершеної системи, а також підвищені габарити готового виробу.

Розглянемо варіант 1 з використанням недорогого мікроконтролера і програмної емуляції USB-протоколу в мікроконтролері. Основна проблема такого підходу полягає в складності досягнення високої швидкості. Шина USB досить швидкодіюча: режим низької швидкості - 1,5 Мбіт / сек, режим повної швидкості - 12 Мбіт / сек, режим високої швидкості - 480 Мбіт / сек. Мікроконтролери AVR повністю відповідають вимогам для інтернету низької режиму USB.

Фізичний інтерфейс USB складається з чотирьох провідників: 2 для живлення зовнішнього пристрою (VCC і GND) і 2 сигнальних провідника (DATA+ і DATA-). Через провідники живлення передається постійна напруга приблизно 5В з навантажувальною здатністю максимум 500 мА. Мікроконтролер AVR живиться через виводи Vcc і GND. Сигнальні провідники називаються DATA+, DATA- і управляють зв'язком між головним (комп'ютер) і пристроєм. Сигнали в цих провідниках є двонаправленими. Рівні напруги - диференціальні: коли DATA+ має високий рівень, тоді DATA- знаходиться на низькому рівні. Однак, є деякі

випадки, коли DATA + і DATA- мають один і той же рівень, наприклад, при EOP (кінець пакета).

Таким чином, програма, що відповідає за реалізацію протоколу USB, повинна контролювати дані сигнали або керувати ними.

Принцип детекції підключення та відключення USB-пристрою заснований на контролі опору лінії USB. У низькошвидкісних USB-пристроїв необхідний підтягуючий резистор між сигналом DATA- та Vcc. У повношвидкісних пристроїв даний резистор підключається до DATA+. Визначаючи, на якій лінії підключений підтягуючий резистор, головний комп'ютер визначає який новий пристрій підключено до лінії USB.

Після визначення нового пристрою, головний починає зв'язок відповідно з фізичним протоколом USB. Протокол USB заснований на синхронній передачі даних. Синхронізація передавача і приймача необхідна для здійснення зв'язку. Синхронізація виконується шляхом передачі невеликого заголовка "зразкова синхронізація", який передуює передачі даних. Даний заголовок являє собою прямокутні імпульси (101010), за ними передаються два 0, а потім дані (рис. 1.6).

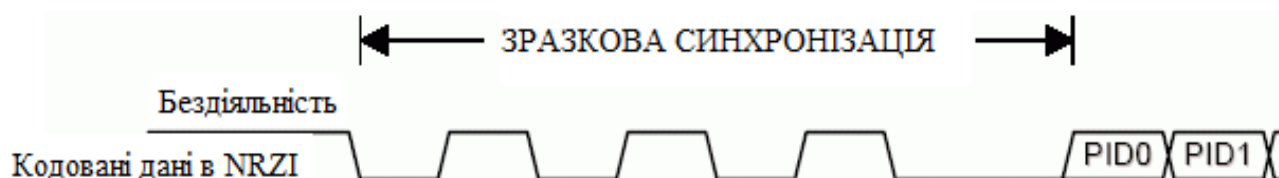


Рисунок 1.6 - Зразкова синхронізація

Прийом даних повинен задовольняти вимогу можливості синхронізувати приймач і передавач в будь-який момент часу. Таким чином, не дозволяється передавати безперервний потік нулів або одиниць по лініях даних. Протокол USB гарантує синхронізацію за рахунок заповнення бітами. Це означає, що, після 6 безперервних одиниць або нулів на лініях даних, вставляється одна одиночна зміна (один біт). Сигнали по лініях USB передаються в коді NRZI. У коді NRZI кожен «0» видається шляхом зрушення поточного рівня сигналу, а кожна «1» шляхом утримання поточного рівня (рис. 1.7). На рівні бітового заповнення це

означає, що кожен нульовий біт вставляється в потік логічних даних після 6 безперервних логічних 1.

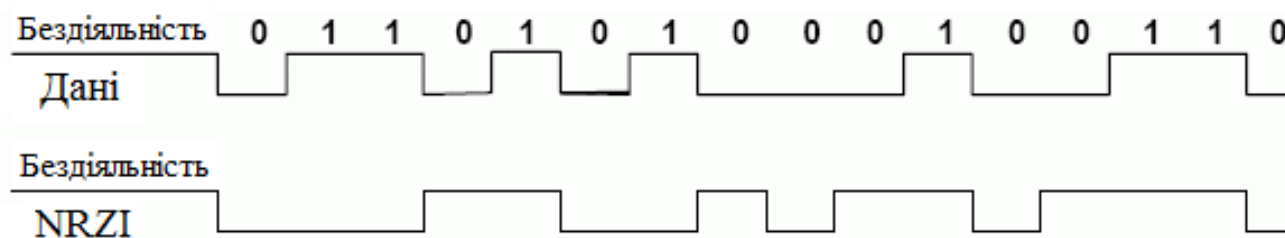


Рисунок 1.7 – Кодування даних в кодї NRZI

Режим передачі управління повинен бути присутнім у кожного USB-пристрої, тому що він використовується для конфігурації при підключенні пристрою, коли необхідно отримати інформацію про пристрій, встановлений адресу пристрою та інше. Кожна передача управління складається з декількох стадій: стадія установки, стадія даних і стадія статусу.

Дані по шині USB передаються пакетами, по кілька байт у кожному. Розмір пакета визначається кожним пристроєм, але його граничний розмір обмежений. Для низькошвидкісних пристроїв максимальний розмір пакета дорівнює 8 байтам. Даний 8-байтний пакет разом з початковим і кінцевим полем повинні бути прийняті в буфер пристрої за одну USB-передачу. У апаратно-реалізованих USB-приймачах різні частини передачі автоматично дешифруються і пристрій повідомляється, коли всі повідомлення призначені окремому пристрою. При програмній реалізації USB-повідомлення дешифрується програмно - після прийому в буфер всього повідомлення. Внаслідок цього виникають вимоги та обмеження: пристрій повинен мати буфер для зберігання всього USB-повідомлення, мати інший буфер для USB-передачі (підготовка даних для передачі), а також виконувати адміністрування з дешифруванням і перевіркою повідомлень. Крім того, необхідна програма для виконання швидкого і точного синхронного прийому з фізичних ліній в буфер і передачі з буфера на лінії.

По-перше, за допомогою адреси аналізується, що прийнятий пакет відноситься до даного пристрою. Адреса передається при кожній USB - транзакції

і, отже, пристрій знає, що передані дані відносяться саме до нього. Дешифрування USB - адреси повинна виконуватися дуже швидко, оскільки, у разі визначення дійсного пакета із заданим адресою, пристрій повинен відповісти комп'ютеру пакетом підтвердження ACK. Таким чином, дешифрування адреси є критичною частиною USB - відповіді.

Якщо комп'ютер запитує відповідь в процесі дешифрування, то пристрій повинен відповісти негайно NAK (немає підтвердження), виходячи з чого комп'ютер зрозуміє про неготовність пристрою. Таким чином, мікроконтролер повинен бути здатний приймати пакети від комп'ютера в процесі дешифрування, визначити чи відноситься транзакція до пристрою, а потім відправити пакет NAK, якщо дешифрування ще перебуває в процесі виконання. У цьому випадку комп'ютер відправить запит знову. Мікроконтролер також дешифрує основну USB - транзакцію і виконує дію.

Основний програмний цикл перевіряє стан прапора дії і визначає, що робити після прийому даних. Крім того, в основному циклі перевіряється, скинутий чи USB-інтерфейс (обидві лінії даних знаходяться в низькому рівні протягом тривалого часу) і, у разі визначення такого стану, виконується переініціалізація пристрою. Якщо прапор дії активний, то викликається відповідна дія: дешифрування коду NRZI в пакеті, видалення порозрядного заповнення та підготовка відповіді в буфері передавача (з порозрядним заповненням та NRZI - кодуванням). Після цього активізується прапор для інформування про готовність відповіді до відправки.

1.1.5 Розробка алгоритму роботи та програмного забезпечення

Алгоритм, що зображений на рисунку 1.8, є основною всього програмного забезпечення.

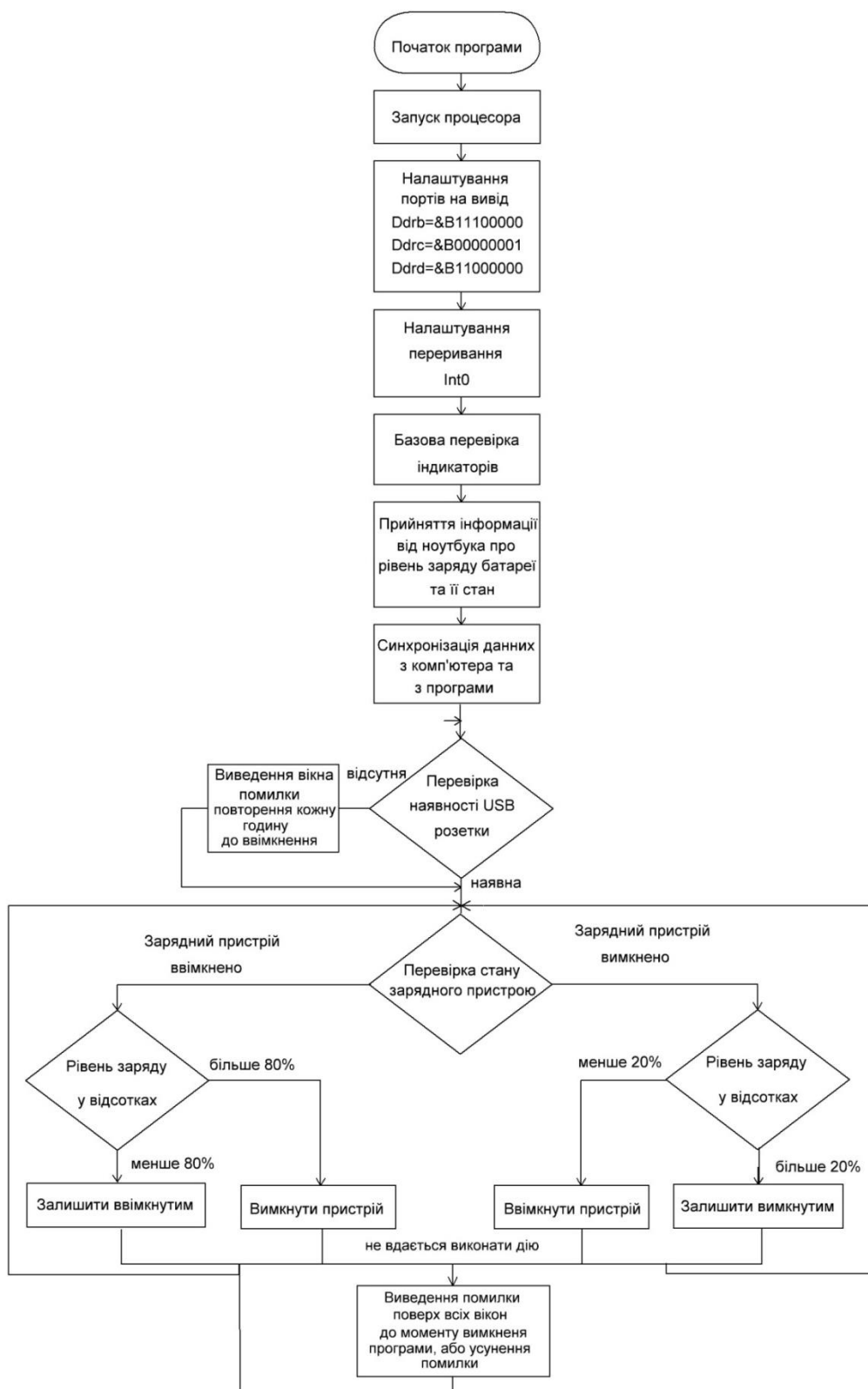


Рисунок 1.8 – Алгоритм роботи програми

З алгоритмом легше зрозуміти які дії виконує мікроконтролер та показує їх поетапно. Він дозволяє точніше зрозуміти роботу програми та спрощує час на пошук несправностей.

Програма починає працювати відразу після підключення пристрою. До цього просто виводиться помилка: «USB пристрій не знайдено», яка зникає після підключення пристрою. Встановлення межі зарядки ноутбука відбувається в програмі, яка знаходиться у вільному доступі за посиланням URL: <https://inlnk.ru/poZYgB>, її код додано до додатку Б. Для показу алгоритму будуть обрані межі від 20% до 80%. Вікно програми зображене на рисунку 1.9.

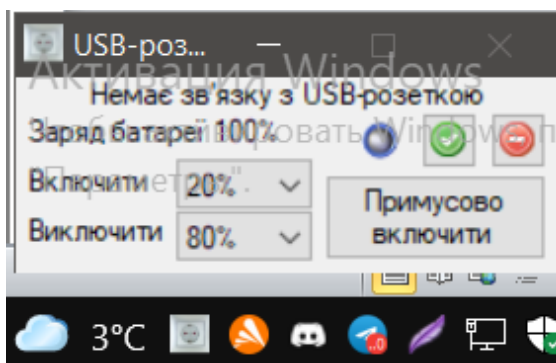


Рисунок 1.9 – Вікно програми

Для розробки програмного забезпечення була вибрана програма Microchip Studio для програмування ATMEGA8. Спочатку налаштуємо Microchip Studio на роботу з мікросхемою ATMEGA8., для цього потрібно зробити наступні кроки:

На рисунку 1.10, у меню програми Microchip Studio потрібно вибрати вкладку File->New->New Project...

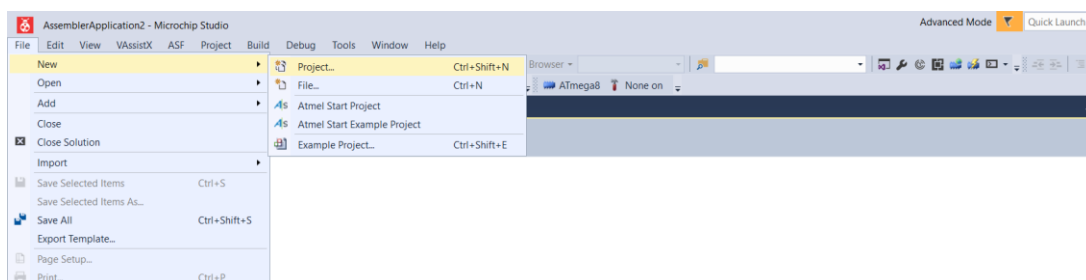


Рисунок 1.10 – Меню програми Microchip Studio

Потім відкриється перше діалогове вікно, зображений на рисунку 1.11, в якому потрібно вибрати мову Assembler->натиснути ОК. Після цих дій з'явиться друге діалогове вікно, зображений на рисунку 1.12, в якому потрібно вибрати потрібний МК. Вибирається мікропроцесор "ATmega8".

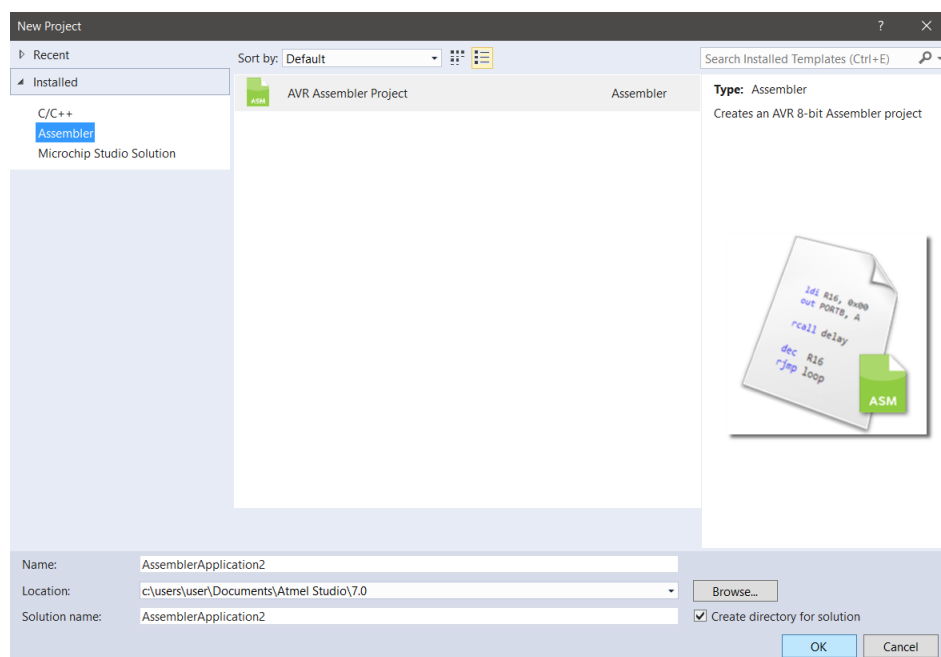


Рисунок 1.11 – Діалогове вікно для створення нового проєкту

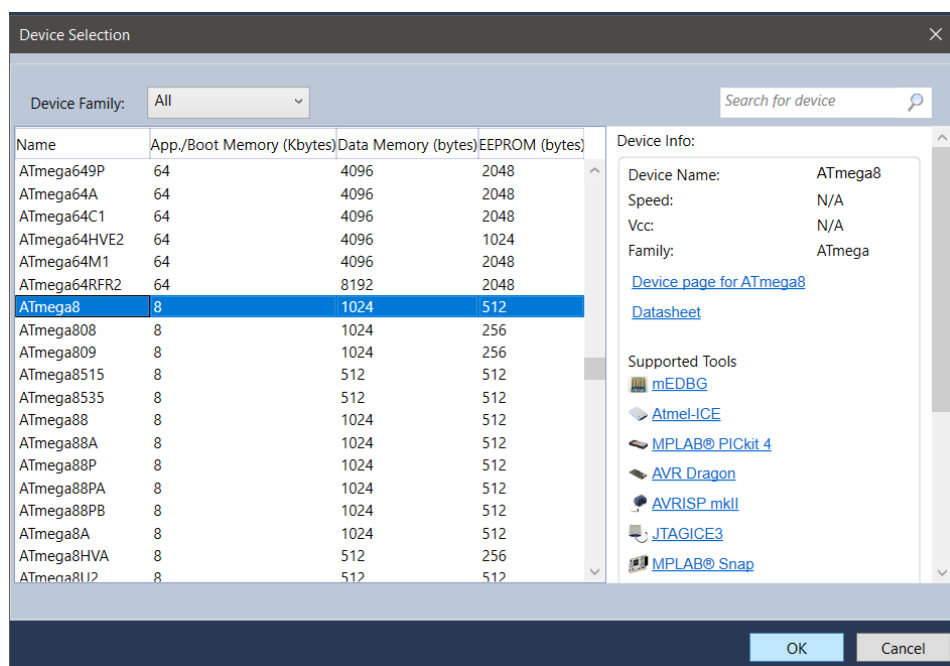


Рисунок 1.12 – Діалогове вікно вибору мікроконтролера

Після встановлення програми була встановлена бібліотека Droopy Library та HID_Lib. Бібліотека HID_Lib створена для роботи з USB HID пристроями.

1.2 Електричні розрахунки

Електричний розрахунок стабілізатора напруги AMS1117-3.3

На рисунку 1.13 зображена принципова схема стабілізатора напруги.

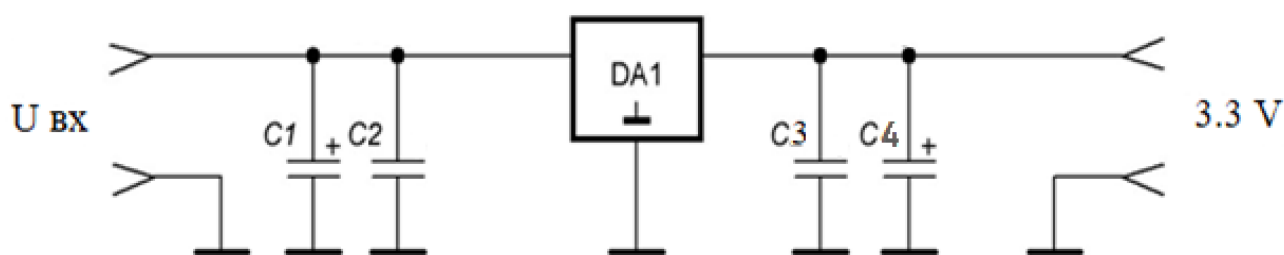


Рисунок 1.13 – схема стабілізатора напруги

Вихідні дані для розрахунку:

- 1) номінальна напруга на виході стабілізатора $U_{\text{вих}} = 3,3\text{В}$;
- 2) номінальний струм навантаження: $I_{\text{вих}} = 20\text{мА}$;
- 3) допустиме відхилення напруги на вході стабілізатора від номінального в сторону збільшення і зменшення $a_{\text{вх}} = b_{\text{вх}} = 5\%$;
- 4) допустиме відхилення напруги на виході стабілізатора $a_{\text{вих}} = b_{\text{вих}} = 0,5\%$;
- 5) допустиме відхилення струму навантаження від номінального в сторону збільшення і зменшення $c = d = 50\%$.

Вибираємо тип мікросхеми кремнієвого стабілізатора. Вибір проводиться на основі заданого значення $U_{\text{вих}}$.

Для забезпечення $U_{\text{вих}} = 3,3\text{В} \pm 0,5\%$ можна вибрати мікросхему стабілізатора типу AMS1117-3.3 з параметрами:

- 1) номінальна напруга стабілізації $U_{\text{ст}} = 3,29 \div 3,35\text{ В}$;
- 2) номінальний струм стабілізації $I_{\text{ст}} = 15\text{ мА}$;
- 3) максимальний струм стабілізації $I_{\text{ст макс. доп.}} = 1,5\text{ А}$;
- 4) динамічний опір $R_{\text{д}} = 6\text{ Ом}$.

Знаходимо максимальне значення коефіцієнта стабілізації.

Підставивши у формулу (1.1), отримаємо

$$K_{CT(u) \text{ макс}} = \frac{U_{\text{вих}} \left(1 - \frac{b_{\text{вих}}}{100}\right)}{\left[I_{\text{вих}} \left(1 + \frac{c}{100}\right) + I_{CT}\right] \cdot R_d}; \quad (1.1)$$

$$K_{CT(u) \text{ макс}} = \frac{3,3 \cdot \left(1 - \frac{5}{100}\right)}{\left[20 \left(1 + \frac{50}{100}\right) + 5\right] \cdot 10^{-3} \cdot 7} = 9,79$$

Величина $K_{CT(u) \text{ макс}}$ повинна бути більше необхідного коефіцієнту стабілізації $K_{CT(u)}$, але не менше ніж в $1,3 \div 1,5$ раз.

$$K_{CT(u) \text{ макс}} \geq (1,3 \div 1,5) K_{CT(u)}$$

$$K_{CT(u) \text{ макс}} = 9,79 > (1,3 \div 1,5) \cdot 10 = 13 \div 15.$$

Визначаємо необхідне значення напруги на вході стабілізатора:

Підставивши у формулу (1.2), отримаємо

$$U_{\text{BX}} = \frac{U_{\text{вих}}}{\left(1 - \frac{b_{\text{BX}}}{100}\right) \cdot \left(1 - \frac{K_{CT(u)}}{K_{CT(u) \text{ макс}}}\right)}; \quad (1.2)$$

$$U_{\text{BX}} = \frac{3,3}{\left(1 - \frac{5}{100}\right) \cdot \left(1 - \frac{10}{9,79}\right)} = 9,20 \text{ (В)};$$

Максимальний струм стабілізатора, розраховуємо за формулою (1.3):

$$I_{CT. \text{ макс}} = I_{CT} + I_{\text{вих}} \cdot \left(1 + \frac{C}{100}\right) - I_{\text{вих}} \left(1 - \frac{d}{100}\right) + \left[I_{\text{вих}} \left(1 + \frac{C}{100}\right) + I_{CT}\right] \cdot \frac{\left(\frac{a_{\text{BX}} + e_{\text{BX}}}{100}\right) \cdot K_{CT(u) \text{ макс}}}{\left(1 - \frac{e_{\text{BX}}}{100}\right) \cdot K_{CT(u)}} \quad (1.3);$$

Підставивши у формулу (1.3), отримаємо:

$$I_{\text{ст макс}} = 5 + 20 \cdot \left(1 + \frac{50}{100}\right) - 20 \cdot \left(1 - \frac{50}{100}\right) + \left[20 \cdot \left(1 + \frac{50}{100}\right) + 5\right] \cdot \frac{\left(\frac{5+5}{100}\right) \cdot 9,79}{\left(1 - \frac{5}{100}\right) \cdot 10} = 28,60 \text{ mA.}$$

Необхідно, щоб знайдене значення струму $I_{\text{ст. макс}}$ не перевищувало $I_{\text{ст. макс доп}}$ для вибраного типу стабілізатора:

$$I_{\text{ст.макс}} \leq I_{\text{ст. макс доп}};$$

$$I_{\text{ст.макс}} = 44,14 < I_{\text{ст. макс доп}} = 1,5 \text{ A.}$$

Визначаємо ємність конденсаторів C_1 , C_6 приймаємо їх як: C_1 , C_2 .

Приймаємо $C_1 = C_2 = 47 \text{ мкФ}$. Вибираємо конденсатор C_1 , C_2 типу *D5-H11-10B-47 мкФ±10%*, через те що живлення відбувається за допомогою кабелю USB, що на виході має 5В а ці конденсатори підходять для придушення його завад.

За допомогою конденсатора, забезпечується стійка робота мікросхеми. Величина C_2 , C_3 вибирається такою $C_2 = C_3 \approx 0,1 \text{ мкФ}$. Вони призначені для фільтрації ВЧ складової, яку не спроможний забезпечити C_1 та C_2 . Вибираємо тип *CBV-21-10B-0,1 мкФ±10%*.

При струмі навантаження не більше 50мА мікросхему ATmega8 необхідно встановити на радіатор.

1.2.1 Електричний розрахунок кварцового генератора

Задаючий генератор електронного частотоміра виконаний на мікроконтролері PIC16F628. Він має в своєму складі генератор, розрахований на роботу з зовнішнім кварцовим резонатором. Схема задаючого генератора зображена на рисунку 1.14.

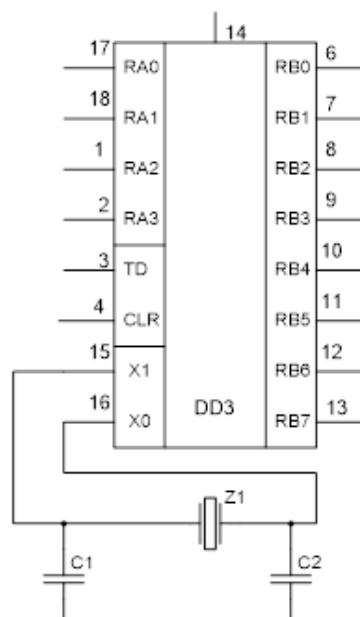


Рисунок 1.14 –Схема задаючого генератора

У схемі використовується кварц типу HC-49S з частотою генерації 12 МГц.

Параметри інших часозадаючих елементів розраховуються за формулою (1,4):

$$F = \frac{1}{2\pi \sqrt{\frac{L_{к\theta} \cdot C1 \cdot C2}{C1 + C2}}} \quad (1.4)$$

За допомогою довідника визначаємо індуктивність кварцового резонатора, яка складає 0,106 мГн.

Відношення ємностей конденсаторів C1 : C2, для стійкості генерації, повинно знаходитись в межах 2...4.

Тоді, якщо C2 дорівнюватиме 3,3 пФ, то C1 дорівнює:

$$C1 = \frac{C2}{F^2 \cdot 4\pi^2 \cdot L_{KK} \cdot C2 - 1} \quad (1.5)$$

За формулою (1.5) знайдемо C1

$$C1 = \frac{3,3 \cdot 10^{-12}}{(12 \cdot 10^6)^2 \cdot 4 \cdot 3,14^2 \cdot 0,106 \cdot 10^{-3} \cdot 3,3 \cdot 10^{-12} - 1} \approx 3,3 \text{ пФ}$$

Конденсатор C1 являється рівним C2, а отже рівним 3,3пФ

Розрахована схема була досліджена в програмі Proteus (логічний елемент НЕ входить до складу мікросхеми PIC16F628) і результати її випробувань наведені на рисунку 1.15.

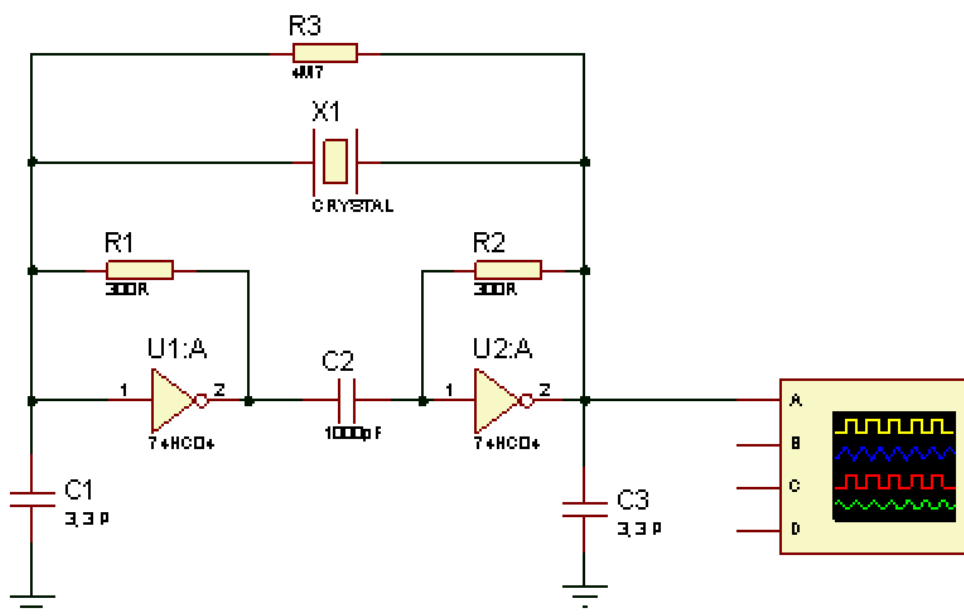


Рисунок 1.15 – Результати моделювання кварцового генератора

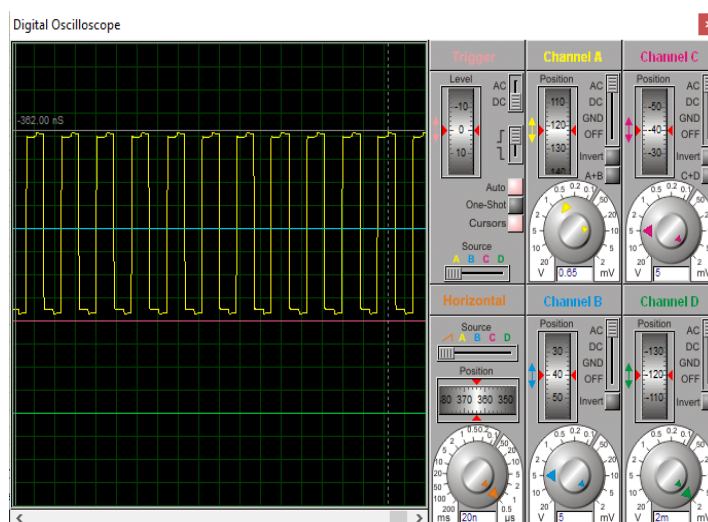


Рисунок 1.16 – Результати моделювання сигналу що виробляє генератор

На екрані осцилографа видно, що період коливань дорівнює $T=80\text{ ns}$, тобто частота коливань складає $f_{ГЕН.}=1/T=1/80\cdot 10^{-9}=12\text{ МГц}$. Амплітуда коливань складає 2,5 В, чого цілком достатньо для нормальної роботи частотоміра.

1.2.2 Розрахунок транзисторного ключа

У даному розділі проводиться розрахунок транзисторного ключа для керування реле, включеного за схемою наведеною на рисунку 1.17.

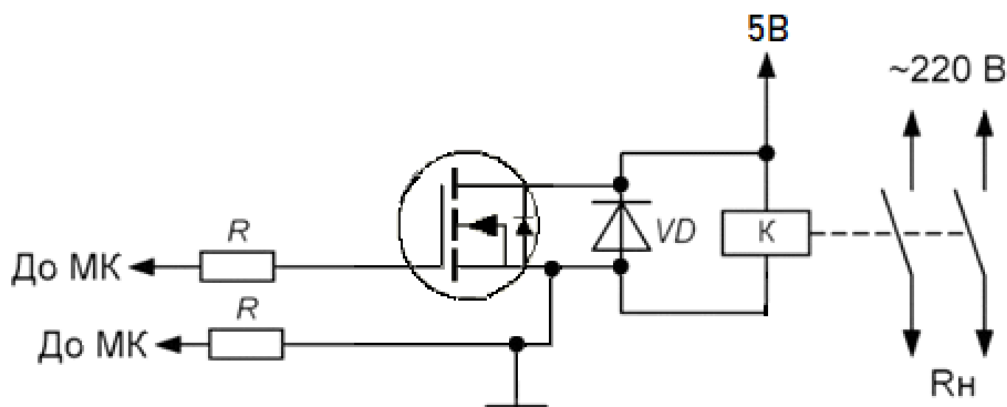


Рисунок 1.17 – Схема блоку реле

Реле в свою чергу буде пропускати через себе напругу 220В. Для зарядного пристрою це значення підходить, вони споживають не більше 300Вт. З таких даних розраховується, що зарядний буде споживати не більше ніж 8 А.

З вихідних даних про споживання зарядного, було вибрано реле 845HN-2A-C-05VDC. У нашому випадку R це опір в ланцюзі затвору, а K це опір нашого навантаження. Розрахуємо ключ для реле 845HN-2A-C-05VDC.

Ось його основні параметри:

- комутація – 8A 5VDC, 8A 240VAC;
- управління – 5 В;
- опір котушки – 36 Ом;
- номінальна потужність котушки – 530 мВт.

Для управління використовується транзистор IRLML2502, його параметри:

- максимальна напруга стік-витік – 20 В;
- потужність розсіювання – 1,25 Вт;
- напруга логічного 0 – 0,4В;
- напруга логічного 1 – 3,3В.

Максимальна напруга стік-витік 20 В для нашого реле необхідно 5 В, що задовольняє вимоги. Далі, максимальний струм 4,2 А, для реле необхідно (1.6):

$$I = U / R = 5/36 = 0,079 \text{ А.} = 79\text{мА.} \quad (1.6)$$

Тепер подивимося що у нас з розсіюваною потужністю (1.7):

$$P = 0,4 * 0,079 = 0.0096\text{W} = 9,6 \text{ мВт.} \quad (1.7)$$

Транзистор IRLML2502 може витримати до 530 мВт, що теж задовольняє вимоги.

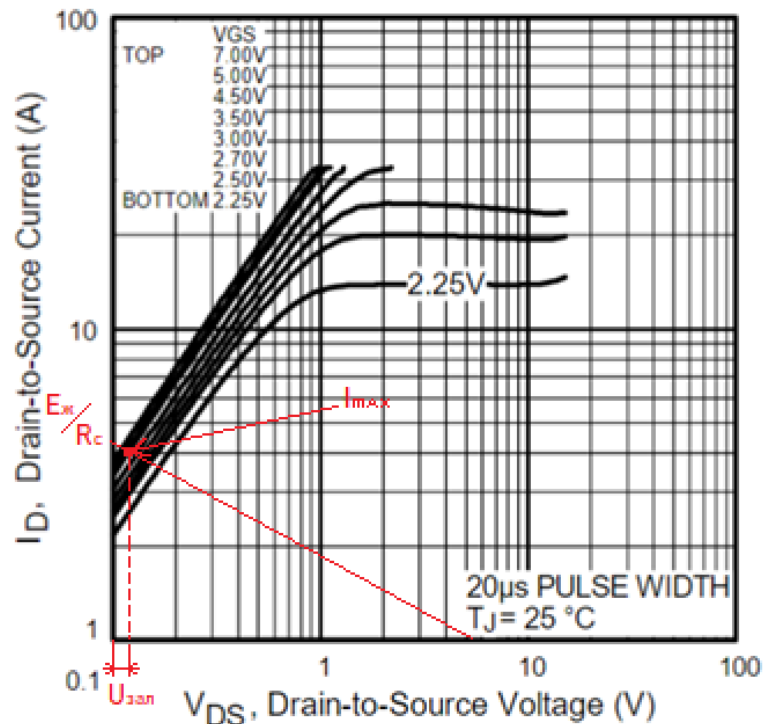


Рисунок 1.18 – ВАХ транзистора IRLML2502

По ВАХ транзистора знаходимо, що при $U_{\text{зал}} = U_0 = 0,2\text{В}$; $I_{\text{мах}} = 3\text{А}$.
Струм через навантаження дорівнює (1.8):

$$I_{RC} = I_{\text{мах}} = \frac{E_C - U_{\text{зал}}}{R_C} = \frac{E_C}{R_C} = \frac{5}{36} = 0,079\text{А} = 79\text{мА} \quad (1.8)$$

де R_c – опір навантаження (котушки реле);

E_c – напруга на навантаженні;

$U_{зал}$ – напруга логічного 0;

I_{max} – струм навантаження.

Час наростання (фронту) імпульсу вхідної напруги при запиранні транзистора (1.9):

$$t_{\phi} = 2,3 R_c \cdot C_{вих}, \quad (1.9)$$

де $C_{вих}$ – ємність затвор-стік транзистора.

$$t_{\phi} = 2,3 \cdot 36 \cdot 90 \cdot 10^{-12} = 7,45 \cdot 10^{-9} \text{ с} < 1 \text{ мс}$$

Далі потрібно обрати діод із зворотною напругою більше 5 В. для цього обрано діод LL4148 його зворотня напруга 75В, що задовільняє умови завдання.

1.3 Розрахунок надійності

Надійність – це властивості приладів зберігати свої параметри в певних експлуатаційних умовах. Розрахунок надійності заключається у визначенні показників надійності

Імовірність безвідмовної роботи – це вірогідність того, що в певному інтервалі часу або в заданих межах напрацювання не відбудеться жодної відмови.

Розрахунок імовірності безвідмовної роботи визначається за формулою (1.10):

$$P(t) = e^{-\lambda \cdot t}, \quad (1.10)$$

де λ – це сумарна інтенсивність відмов;

t – час, необхідної безвідмовної роботи (10000 ч).

Інтенсивність відмов показує, яка частина елементів, відносно загальної кількості працюючих елементів в середньому виходить з ладу за одиницю часу. Визначається за формулою (1.11):

$$\lambda_0 = \sum \lambda_{ie} N_i, \quad (1.11)$$

де λ_{ie} - інтенсивність відмов елемента i ;

N_i – кількість елементів.

Напрацювання на відмову T_0 – це середній час роботи виробу між суміжними відмовами. $T_0 = 1 / \lambda_0$.

Таблиця 1.2 – Попередній розрахунок надійності

Тип елемента	Середньо групові значення інтенсивності відказів λ_0 , 10^{-6} [1/год]	К-ть елементів	Інтенсивність відмов, λ
1	2	3	4
АТМega8	0,99	1	0,99
Реле	0,4	1	0,4
Транзистор	0,3	1	0,3
Запобіжник	0,2	2	0,4
Кварцевий генератор	0,047	1	0,047
Конденсатор ел.	0,03	8	0,32
Конденсатор кер.	0,3	2	0,6
Мережевий роз'єм	0,35	1	0,35

Продовження таблиці 1.2

1	2	3	4
Стабілітрон	0,07	1	0,07
Резистор	0,006	9	0,054
Світлодіод	0,13	2	0,26
Роз'єм USB	0,78	1	0,78
Пайка з друкованим монтажем	0,0006	95	0,057
Всього			4,628

Система автоматичного регулювання підсилення широкосмугового цифрового приймача типу GB (наземне стаціонарне обладнання), для якого поправочний коефіцієнт $\pi E = 0,5$ тому за формулою (1.12):

$$FR = \lambda (1 + 0,2 \pi E) \quad (1.12)$$

Результати розрахунку:

$$FR = 5,09 \times 10^{-6} \text{ 1/год.},$$

$$MBTF = 196\,463,65 \text{ год.}$$

Побудуємо графік ймовірності безвідмовної роботи системи автоматичного регулювання підсилення широкосмугового цифрового приймача за формулою (1.13):

$$P(t) = e^{-FR \times t}, \quad (1.13)$$

де $t = 0 \dots 100000$ годин

Таблиця 1.3 – Залежність $P(t) = e^{-FR \times t}$

Т, тис. год	0	10	20	30	40	50	60	70	80	90	100
P(t)	1,00	0,95	0,90	0,85	0,81	0,76	0,72	0,769	0,66	0,63	0,61

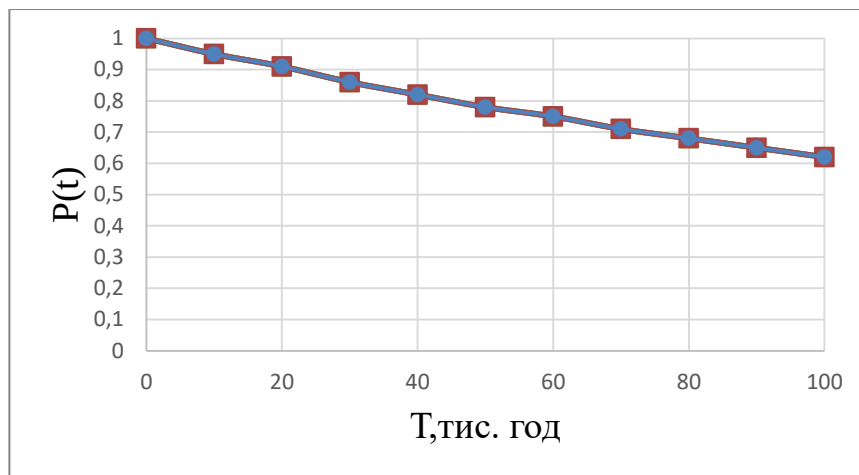


Рисунок 1.19 – Графік ймовірності безвідмовної

З результатів розрахунку, зображених на рисунку 1.19, видно що інтенсивність відмови приладу приблизно $\lambda = 5,09 \cdot 10^{-6}$ 1/год., а час безвідмовної роботи приблизно $T_{\text{ср}} = 196\,463,65$ год.

2 ТЕХНОЛОГІЧНА ЧАСТИНА

2.1 Розробка технологічного процесу настройки пристрою

Технологічний процес регулювання та настройки системи автоматичного регулювання підсилення широкосмугового цифрового приймача наведений в таблиці 2.1. Технологічний процес настройки складається з основних технологічних операцій: перевірка роботи, підключення USB, настройка мікроконтролера, перевірка працездатності.

Таблиця 2.1 – Технологічний процес регулювання

Технологічна операція	Вимірювальні прилади	Примітка
1	2	3
1. Підготовка		
1.1 Перевірка якості монтажу, пайки, травки доріжок, відсутність коротких замикань, правильність встановлення елементів.	Мультиметр, візуально	
1.2 Перевірка наявності механічних пошкоджень.	Візуально	
2. Регулювання		
2.1 Встановлення програми для прошивання мікроконтролера ATmega8.	Ноутбук HP-15-dw3023	
2.2 Підключення мікроконтролера до програматора та з'єднання його з комп'ютером.	Ноутбук HP-15-dw3023, програматор USB ASP	
2.3 Програмування мікроконтролера за допомогою програми Microchip Studio	Ноутбук HP-15-dw3023, програматор USB ASP	
2.4 Настройка ф'юзів мікроконтролера, а саме живлення, синхронізації, FLASH пам'яті.	Ноутбук HP-15-dw3023, програматор USB ASP	Рисунок 2.2

Продовження таблиці 2.1

1	2	3
3. Контрольні		
3.1 Підключити пристрій до ноутбука.	Ноутбук HP-15-dw3023	
3.2 Ввімкнути пристрій та почати перевірку напруг на певних точках (вхід – 5В, вихід стабілізатора 3,3В, вихід 18 AVCC – 3,3В, катод діода VD1 – 5В).	Мультиметр	Напруга на AVCC та VCC не повинна різнитись між собою більш ніж на 0,3В.
3.3 Перевірити наявність помилок на екрані та на індикаторі	Ноутбук HP-15-dw3023,	
4. Заключні		
4.1 Перевірка справності світлодіодів.	Батарейка на 9В	
4.2 Перевірка відповідності заряду ноутбука в програмі, та в самому ноутбуці.	Ноутбук HP-15-dw3023	
4.3 Перевірка можливості зміни мови	Ноутбук HP-15-dw3023	Рисунок 2.1
4.4 Підключити пристрій до мережі, та ноутбук до пристрою	Ноутбук HP-15-dw3023	
4.5 Запустити програму та встановити потрібні налаштування.	Ноутбук HP-15-dw3023	
4.6 Перевірити роботу пристрою при встановлених різних межах зарядки.	Ноутбук HP-15-dw3023	

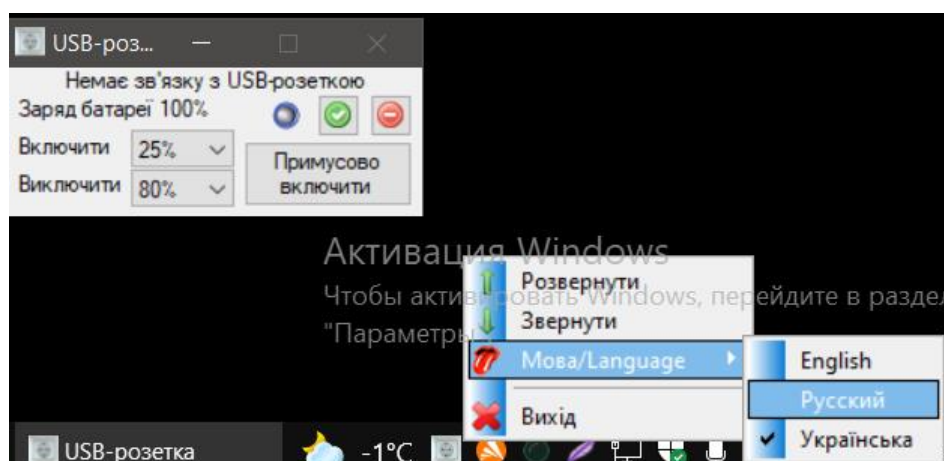


Рисунок 2.1 – Вікно програми

Для мікросхеми ATmega8

Вибір потрібних режимів роботи мікросхеми

Результати (стан FUSE перемикачів) відображається в кінці сторінки

Режими синхронізації та затримки роботи запуску (CKSEL, SUT)

Зовнішній кристал/резонатор високої частоти, Затримка запуску: 16K тактів+64 мс

☐ Ввімкнути систему контролю напруги живлення (BODEN)

Рівень вмикання системи контролю живлення (BODLEVEL)

Вмикається при напрузі Vcc=2,7В

☐ Після скидання почати роботу з селектора загрузки (BOOTRST)

Розмір секції завантажувача Flash пам'яті (BOOTSZ)

Розмір секції = 1024 слова. Адрес початкового запуску=\$0C00

☐ Не видалити EEPROM при виконанні команди "Стерти Кристал" (EESAVE)

☐ Значення FUSE-перемикача CKOPT (залежить від значення CKSEL)

☒ Дозволити послідовне програмування (SPIEN)

☐ Ввімкнути таймер

☐ Вимкнути вхід RESET (Використовувати PC6 як вхід вводу/виводу)

Встановити значення за замовчуванням

	Встановлення чекбоксів в PonyProg	BIN	HEX
Старший байт	☐ RSTDISL ☐ WTDON ☒ SPIEN ☐ CKOPT ☐ EESAVE ☒ BOOTSZ1 ☒ BOOTSZ0 ☐ BOOTRST	11011001	D9
Молодший байт	☐ BODLEVEL ☐ BODEN ☐ SUT1 ☐ SUT0 ☐ CKSEL3 ☐ CKSEL2 ☐ CKSEL1 ☐ CKSEL0	11111111	FF

AVRDUDE -U hfuse:w:0xFF:m -U lfuse:w:0xD9:m

Рисунок 2.2– Налаштування ф’юзів ATMEGA8 для правильної роботи програми

При відсутності під’єданого пристрою виводиться вікно зображене на рисунку 2.3.

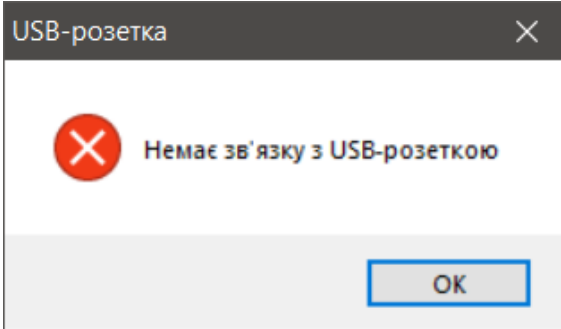


Рисунок 2.3 – Вікно помилки

2.2 Основні несправності та методи їх усунення

При використуванні системи автоматичного регулювання підсилення широкосмугового цифрового приймача для регулювання зряду батареї ноутбука можливі несправності але їх діагностика та причини наведені в таблиці 2.2. Ця таблиця розрахована на сервісні центри та ремонтні майстерні для полегшення ремонту.

Таблиця 2.2 – Основні несправності та методи їх усунення

Характерні несправності	Можливі причини	Способи усунення
Пристрій не працює з'єднання наявне червоний світлодіод	Згорів запобіжник FU2, збій програми, пробій транзистора VT1	Спочатку перезавантажити програму, перевірити працездатність запобіжника FU1, перевірити працездатність транзистора VT1.
Пристрій заряджає але не працює вимкнення та вмикання	Замкнуті контакти реле K1, відсутня напруга 5В на діоді VD1	Потрібно перевірити наявність напруги живлення на мікросхемі DA1, стабілізатор напруги DA1, контакти USB
Нічого не працює	Проблеми з програмою	Переустановити програмне забезпечення
Пристрій заряджає, але нічого більше не працює	Проблеми з кабелем USB, згорів запобіжник FU1, вийшов з ладу стабілізатор DA1	Перевірити підключення до пристрою, перевірити справність кабелю USB, порту USB ноутбука, перевірити справність запобіжника FU1, замінити його при поломці, перевірити напругу на виході стабілізатора напруги DA1 повинна бути 3,3В при подачі напруги на вхід 5В, пройти ту саму процедуру (перевірка, заміна) з стабілізатором напруги DA1.

3 РОЗРОБКА ПРОГРАМНОГО ЗАБЕЗПЕЧЕННЯ

У данному розділі проводиться дослідження роботи системи автоматичного регулювання підсилення широкосмугового цифрового приймача, а саме програмне забезпечення для ноутбука, виведення помилок.

Основне керування здійснюється за допомогою програми, що встановлюється на ноутбук. Також з'являється ярлик на робочому столі та іконка на панелі задач, що дає можливість керувати приладом.

На рисунку 3.1 зображено ярлик, іконку та контекстне меню програми, яке відкривається з панелі задач.

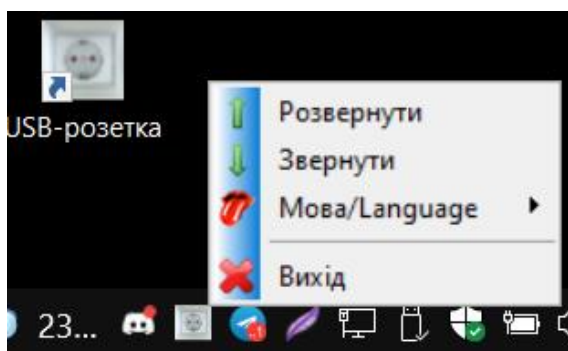


Рисунок 3.1 – Контекстне меню програми

Як видно на рисунку, у програми відкривається контекстне меню з панелі задач, що має 4 функції:

- розгорнення вікна програми (рисунок 3.2);
- згортання вікна програми;
- зміна мови на 3 різні мови (рисунок 3.3, 3.4, 3.5);
- вихід з програми.

Вікно програми з індикатором, що свідчить про помилку роботи приладу та кнопка примусового вимкнення зарядного пристрою.

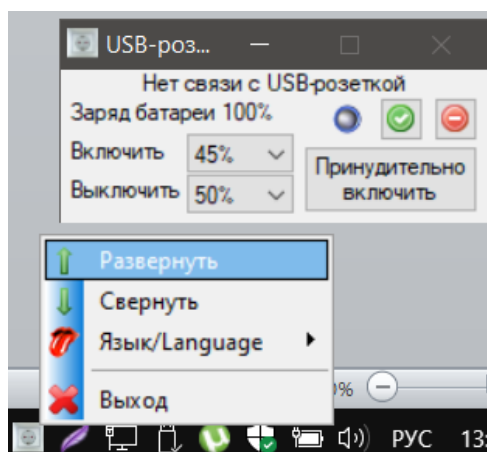


Рисунок 3.2 – Вікно програми

Контекстне меню має можливість зміни мови що також впливає на вікно програми. Можливо використовувати 3 мови: українська, російська, англійська. Рисунки 3.3, 3.4, зображують зміну мови.

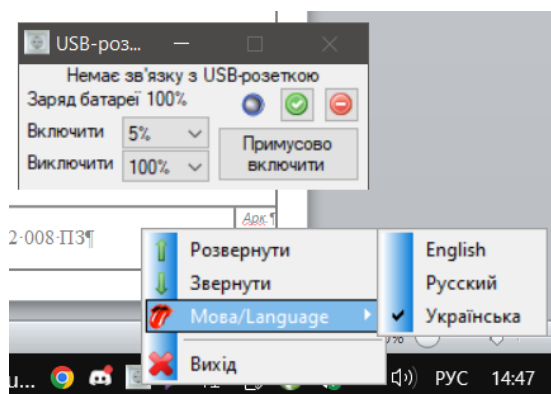


Рисунок 3.3 – Контекстне меню програми з обраною українською мовою

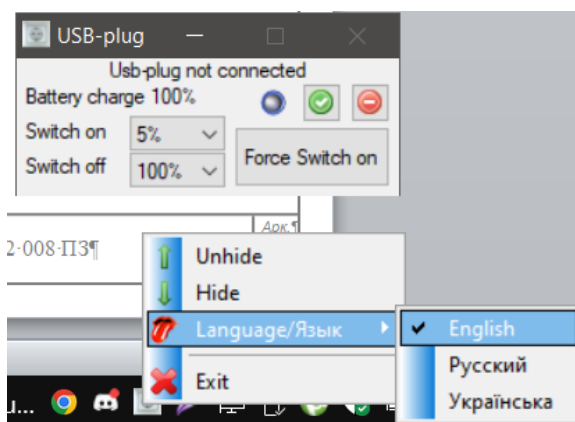


Рисунок 3.4 – Контекстне меню програми з обраною англійською мовою

Так як основне завдання – це зчитування даних про рівень заряду та стан зарядного пристрою, програма відмінно показує рівень заряду батареї ноутбука і стан зарядного пристрою, але при 100% стан вимкнено, хоч зарядне. Він точно такий самий, який виводиться самим комп'ютером. На рисунку 3.5 зображено рівень заряду вказаний самим комп'ютером та рівень заряду вказаний програмою. На рисунку 3.6 зображено відповідність рівня зарядки а також стан зарядки, а саме ввімкнена.

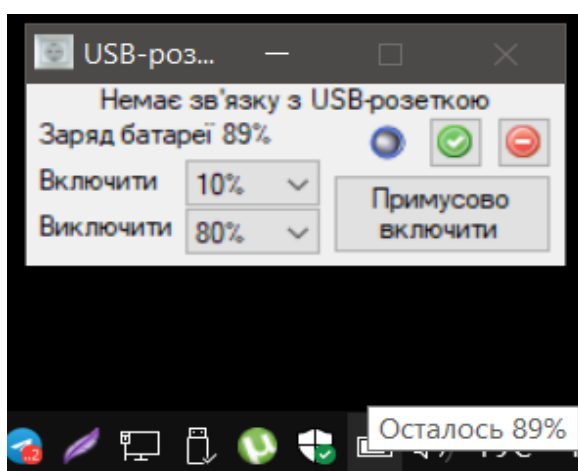


Рисунок 3.5 – Відповідність відображення рівня заряду батареї ноутбука

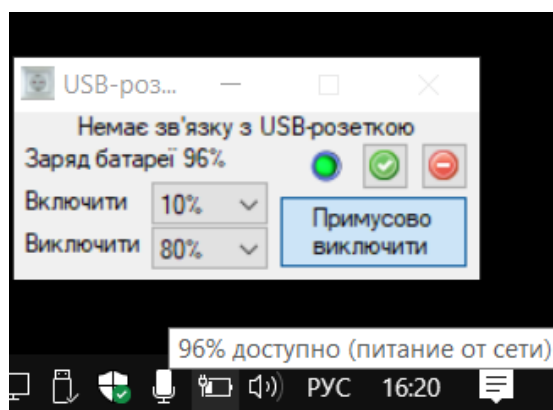


Рисунок 3.6 – Відповідність стану зарядного пристрою

В програмі є вікно помилки, для прикладу можливо показати лише одне, а саме про відсутність підключення. Вікно помилки зображено на рисунку 3.7.

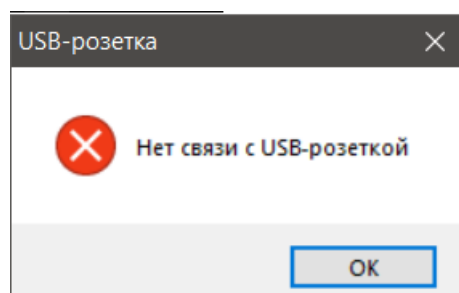


Рисунок 3.7 – Вікно помилки

Згідно проведених досліджень можна зробити висновок, програмне забезпечення для ноутбука правильно працює та задовільняє поставлене завдання

4 ОХОРОНА ПРАЦІ

Під час проектування системи автоматичного регулювання підсилення широкосмугового цифрового приймача на працівника могли мати вплив такі небезпечні та шкідливі виробничі фактори:

1. Фізичні: підвищена запиленість та загазованість повітря робочої зони; підвищена чи понижена температура повітря робочої зони; підвищений рівень шуму на робочому місці; підвищений рівень електромагнітного випромінювання; підвищена чи понижена іонізація повітря; пряма і відбита блискість; підвищення яскравість; недостатня освітленість робочої зони.

2. Психофізіологічні: статичне перевантаження; розумове перевантаження; емоційні перевантаження.

Відповідно до наведених факторів здійснюємо планування щодо безпечного виконання роботи.

4.1 Технічні рішення щодо безпечного виконання роботи

Під робочим місцем проектанта системи автоматичного регулювання підсилення широкосмугового цифрового приймача маєтись на увазі зона його трудової діяльності, оснащення необхідними засобами праці.

Загальною вимогою раціональної організації робочих місць є створення працівникам максимальної зручності для здійснення їх діяльності, вивільнення від нераціональних переміщень та рухів, зниження на цій основі витрат праці на виконувану роботу. Економія часу і сил працівника - головні критерії раціональності організації робочих місць.

Робоче місце і взаємне розташування усіх її елементів має відповідати антропометричним, фізичним і неординарним психологічним вимогам [12]. Важливе значення має також характеру роботи. Зокрема, з організацією робочого місця проектувальника мають бути дотримані такі основні умови:

- оптимальне розміщення устаткування, що до складу робочого місця;

- достатній робочий простір, що дозволяє здійснювати всі необхідні руху, і переміщення;
- необхідне природне, і штучне висвітлення до виконання поставлених завдань;
- рівень акустичного шуму, який не перевищує допустимого значення;
- достатня вентиляція робочого місця;

Головними елементами робочого місця розробника є письмовий стіл і крісло. Основним робочим положенням є положення сидячи. Робоче місце для виконання робіт у положенні сидячи організується відповідно до вимог [12].

Необхідно передбачити при проектуванні можливість різного розміщення документів: збоку від ПК, між монітором і клавіатурою і т.п. Крім того, у випадках, коли ПК має низьку якість зображення, наприклад, помітні миготіння, відстань від очей до екрана роблять більшою (близько 700 мм), ніж відстань від ока до документа (300...450 мм). В загальному випадку, при високій якості зображення на ПК відстань від очей користувача до екрана, документа і клавіатури може бути рівним.

Велике значення надається характеристикам робочого крісла [13]. При проектуванні крісла виходять з того, що при будь-якому робочому положенні дослідника його поза повинна бути фізіологічно правильно обґрунтованою, тобто положення частин тіла повинно бути оптимальним. Для задоволення вимог фізіології, що випливають з аналізу положення тіла людини в положенні сидячи, конструкція робочого сидіння повинна відповідати таким основним вимогам:

- допускати можливість зміни положення тіла, тобто забезпечити вільне переміщення корпусу і кінцівок тіла один щодо одного;
- допускати регулювання висоти в залежності від росту працюючої людини (в межах від 400 до 550 мм);
- мати злегка увігнуту поверхню, мати невеликий нахил, тому поверхню сидіння рекомендується робити м'якою, передній край закругленим, а кут нахилу спинки робочого крісла - регульованим.

Приміщення, де здійснювалася робота за небезпекою ураження електричним струмом можна віднести до 1 класу, тобто це приміщення без підвищеної небезпеки (сухе, мало заповишене, з нормальною температурою повітря, ізолюваними підлогами і малим числом заземлених приладів) [14].

На робочому місці з усього обладнання металевим є лише корпус системного блоку комп'ютера, в якого крім робочої ізоляції передбачений елемент для заземлення та провід з заземлюючий жилою для приєднання до джерела живлення.

Електротехнічне устаткування: апаратури, кабелі й керівництва, розподільні пристрої всіх видів і напруг по своїх номінальних параметрах задовольняє умовам роботи як при нормальних режимах, так і при коротких замиканнях, перенапругах, перевантаженнях.

Для забезпечення безпеки установлюються наступні технічні рішення:

- Забезпечено недоступність струмопровідних частин (застосована схована проводка, кабель прокладений у спеціальних ринвах).
- Забезпечено ізолювання струмопровідних частин з використанням ізоляції, опір якої не нижче 1кОм/В, передбачені постійний контроль і профілактика ізоляції.
- Напруга освітлювальної мережі приймається 220 В із заземленою нейтраллю.

У випадку виникнення аварійної ситуації працівник зобов'язаний:

- у всіх випадках виявлення пошкодження проводів електричного живлення, несправності заземлення та інших пошкодженнях електрообладнання, виникненні запаху гарі, диму - негайно вимкнути електричне живлення і повідомити про аварійну ситуацію свого безпосереднього керівника й чергового електрика;
- при попаданні людини під електричну напругу негайно звільнити її від дії струму шляхом вимкнення електричного живлення, до прибуття лікаря надати потерпілому долікарську медичну допомогу;

–при будь-яких випадках порушень роботи технічного обладнання або програмного забезпечення негайно викликати представника технічної служби з питань експлуатації обчислювальної техніки;

–у випадку виникнення різі в очах, різкого погіршення зору, виникнення головного болю, больових відчуттів у пальцях та кистях рук, посилення серцебиття - негайно припинити роботу з використанням ПК, повідомити про те, що сталося, свого безпосереднього керівника й звернутися до медичної установи;

–при загорянні обладнання негайно відключити його від електромережі;

–про загорання повідомити свого безпосереднього керівника, оперативного чергового, пожежну службу; ужити заходів щодо ліквідації вогню за допомогою вуглекислотного або порошкового вогнегасника.

4.2 Технічні рішення з гігієни праці та виробничої санітарії

4.2.1 Мікроклімат

Робота, яка виконується розробником, згідно за енерговитратами відноситься до категорії I а (енерговитрати до 139Дж/с) [15]. Допустимі параметри мікроклімату для цієї категорії наведені в табл. 4.1 (відповідно до ДСН 3.3.6.042-99 [16]).

Таблиця 4.1 – Параметри мікроклімату

Період року	Допустимі		
	t , °C	W , %	V , м/с
Теплий	2	5	0
	2-28	5	,1-0,2
Холодний	2	7	0
	1-25	5	,1

Для забезпечення необхідних за нормативами параметрів мікроклімату в приміщенні передбачено система опалення, система кондиціонування та систематичне вологе прибирання.

Вимірювання параметрів мікроклімату проводиться на робочому місці і в робочій зоні на початку, в середині та в кінці робочої зміни. Вимірювання здійснюються не менше 2-х разів на рік (теплий та холодний періоди року) у порядку поточного санітарного нагляду, а також при прийманні до експлуатації нового технологічного устаткування, внесенні технічних змін в конструкцію діючого устаткування, організації нових робочих місць тощо.

4.2.2 Склад повітря робочої зони

У сучасній техніці застосовується безліч речовин, які можуть потрапляти в повітря і становити небезпеку здоров'ю людей. Для визначення небезпечності досліджується вплив цих речовин на організм людини і встановлюють безпечні для людини концентрації та дози, які можуть потрапити різними шляхами в організм людини.

Для створення здорових і безпечних умов праці потрібно мати гігієнічне нормування шкідливих речовин, надійні способи визначення їх концентрацій у повітрі і сучасне технічне та організаційне забезпечення їх знешкодження.

В приміщенні, де здійснюється робота можливими шкідливими речовинами у повітрі є пил та озон. Джерелами цих речовин є офісна техніка. Пил потрапляє у приміщення ззовні. ГДК шкідливих речовин, які знаходяться в досліджуваному приміщенні, наведені в таблиці 4.2

Таблиця 4.2 – ГДК шкідливих речовин у повітрі

Назва речовини	ГДК, мг/м ³	Середньо добова	Клас небезпечності
	Максималь но разова		
Пил нетоксичний	0,5	0,15	4
Озон	0,16	0,03	4

Параметри іонного складу повітря на робочому місці, що обладнане ПК, повинні відповідати допустимим нормам (табл. 4.3).

Таблиця 4.3 – Рівні іонізації повітря приміщень при роботі на ПК

Рівні	Кількість іонів в 1 см ³	
	n+	n-
Мінімально необхідні	400	600
Оптимальні	1500-3000	3000-5000
Максимально необхідні	50000	50000

Для забезпечення комфортних умов використовуються як організаційні методи (раціональна організація проведення робіт залежно від пори року і доби, чергування праці і відпочинку), так і технічні засоби (вентиляція, кондиціонування повітря, опалювальна система).

4.2.3 Виробниче освітлення

Серед чинників зовнішнього середовища, що впливають на організм людини в процесі праці, світло посідає одне з перших місць. Адже відомо, що майже 90 % усієї інформації про довкілля людина одержує через органи зору. Підчас здійснення будь-якої трудової діяльності втомлюваність очей, в основному, залежить від напруженості процесів, що супроводжують зорове сприйняття. До таких процесів належать адаптація, акомодация та конвергенція.

Світло впливає не лише на функцію органів зору, а й на діяльність організму в цілому. У разі поганого освітлення людина швидко втомлюється, працює менш продуктивно, зростає потенційна небезпека помилкових дій і нещасних випадків.

Норми освітленості при штучному освітленні та КПО (для III пояса світлового клімату) при природному та сумісному освітленні, які необхідно забезпечити під час виконання роботи зазначені у таблиці 4.4 (відповідно ДБН В.2.5-28-2018 [17]):

Таблиця 4.4 - Норми освітленості в приміщенні

Характеристика зорової роботи	Найменший розмір об'єкта розрізнення	Розряд зорової роботи	Підрозряд зорової роботи	Контраст об'єкта розрізнення з фоном	Характеристика фона	Освітленість, лк		КПО, e_n , %		
						Штучне освітлення		Природне освітлення		Сумісне освітлення
						Комбіноване	Загальне	Верхнє або верхнє і бокове	Бокове	Верхнє або верхнє і бокове Бокове
Дуже високої точності	Від 0,15 до 0,3	I		Великий	Світлий	000	00		,5	,2 4 ,5

Для забезпечення достатнього освітлення передбачені такі заходи:

- 1) Систематичне очищення скла від бруду – не рідше двох разів на рік.
- 2) Система природного освітлення доповнюється загальним штучним освітленням, що створюється за допомогою люмінесцентних ламп.

4.2.4 Виробничий шум

Для робочих місць в приміщеннях із ПК характерна наявність механічних шумів (виникають внаслідок коливання деталей машин та їхнім взаємним переміщенням), аеродинамічних шумів (виникають в пружних конструкціях у газі або рідині) та шумів електричних машин.

Деякі ПК є потенційними джерелами цілого ряду звукових коливань як чутного, так і ультразвукового діапазону. Найчастіше рівні акустичного випромінювання, що виходить від ПК, охоплюють діапазон частот від 6,3 до 40 кГц.

Санітарні норми виробничого шуму, ультразвуку та інфразвуку відображені в ДСН 3.3.6.037-99 [18]. Для умов виконання роботи допустимі рівні звукового тиску повинні наведені в таблиці 4.5.

Таблиця 4.5 – Допустимі рівні звукового тиску і рівні звуку для постійного широкополосного шуму

Характер робіт	Допустимі рівні звукового тиску (дБ) в стандартизованих октавних смугах зі середньгеометричними частинами (Гц)									Допустимий рівень звуку, дБА
	32	63	125	250	500	1000	2000	4000	8000	
Виробничі приміщення	86	71	61	54	49	45	42	40	38	50

Для забезпечення допустимих параметрів шуму доцільно використовувати комп'ютери з пасивним охолодженням та встановити пластикові вікна, які мають достатню звукоізоляцію.

4.2.5 Виробничі випромінювання

Під час проектування системи автоматичного регулювання підсилення широкосмугового цифрового приймача існує ризик виникнення підвищеного рівня електромагнітного поля. Несприятливу дію на організм людини мають електромагнітні випромінювання радіочастотного діапазону, джерелами яких є телевізійні і радіомовні станції, пристрої стільникового радіозв'язку, апарати високочастотного нагрівання і навіть побутова апаратура.

Електромагнітні поля чинять термічний і морфологічний вплив на організм людини, викликаючи в ньому функціональні зміни.

Допустимі значення параметрів неіонізуючих електромагнітних випромінювань від монітора комп'ютера представлені в табл. 4.6.

Таблиця 4.6 - Допустимі значення параметрів неіонізуючих електромагнітних випромінювань

Найменування параметра	Допустимі значення
Напруженість електричної складової електромагнітного поля на відстані 50см від поверхні відеомонітора	10В / м
Напруженість магнітної складової електромагнітного поля на відстані 50см від поверхні відеомонітора	0,3 А / м
Напруженість електростатичного поля не повинна перевищувати: для дорослих користувачів для дітей дошкільних установ і що вчаться середніх спеціальних і вищих навчальних закладів	20кВ / м 15кВ / м

Для зниження дії цих видів випромінювання рекомендується застосовувати монітори із зниженим рівнем випромінювання (MPR-II, TCO-92, TCO-99, TCO-03), а також дотримувати регламентовані режими праці і відпочинку.

4.3 Пожежна безпека

Метою пожежної безпеки об'єкта є попередження виникнення пожежі на визначеному чинними нормативами рівні, а у випадку виникнення пожежі – обмеження її розповсюдження, своєчасне виявлення, гасіння пожежі, захист людей і матеріальних цінностей.

В приміщенні, де здійснювалося проектування системи автоматичного регулювання підсилення широкосмугового цифрового приймача використовуються тільки негорючі речовини та матеріали у холодному стані, тому за ступенем вибухопожежної та пожежної небезпеки приміщення відноситься до категорії «Д».

За вогнестійкістю приміщення відноситься до другої категорії [19]. Робоча зона розробника відноситься до класу вибухонебезпечності В-IIа, оскільки вибухонебезпечна концентрація пилу і волокон може утворюватися лише внаслідок аварії або несправності.

4.3.1 Технічні рішення системи запобігання пожежі

Система попередження пожежі – це комплекс організаційних заходів та технічних засобів, спрямованих на усунення умов виникнення пожежі.

У дослідженому приміщенні є відповідальний за пожежну безпеку працівник. Меблі та обладнання розміщені таким чином, щоб забезпечувався вільний евакуаційний прохід до дверей виходу з приміщення. Евакуаційні шляхи та виходи постійно утримуються вільними.

Документи, папір та інші горючі матеріали зберігаються на відстані не менше 1 м від електрощитів, електрозборок і електрокабелів, 0,5 м від світильників та 0,25 м від приладів опалення. Засоби протипожежного захисту (пожежні крани, пожежна та охороно - пожежна сигналізація, первинні засоби пожежегасіння тощо), які є у приміщенні утримуються у справному стані.

Можливі причини виникнення пожежі у приміщенні такі:

- несправна електропроводка (іскріння, перегрів провідників, пересихання електроізоляційних матеріалів);
- використання електропобутових пристроїв (електрочайники, обігрівачі);
- попадання вологи на працююче електрообладнання;

Для запобігання виникнення пожежі здійснюються такі заходи:

- щорічне проведення повторних протипожежних інструктажів та занять за програмою пожежно-технічного мінімуму з особами, що відповідальні за пожежну безпеку;
- утримання в справному стані засобів протипожежного захисту;
- своєчасне інформування про несправність пожежної техніки, систем протипожежного захисту, водопостачання тощо.

4.3.2 Технічні рішення системи протипожежного захисту

Система протипожежного захисту – це сукупність організаційних заходів, а також технічних засобів, спрямованих на запобігання впливу на людей небезпечних факторів пожежі та обмеження матеріальних збитків від неї.

У приміщені на випадок виникнення пожежі для обмеження її розповсюдження знаходиться переносний вуглекислотний вогнегасник типу ОУ-5, що відповідає нормам. Підходи до засобів первинного пожежогасіння та відключення електросхем устаткування вільні.

З метою дотримання протипожежного захисту забороняється:

- влаштовувати тимчасові електромережі, застосовувати саморобні плавкі вставки в запобіжниках, прокладати електричні проводи безпосередньо по горючій основі, експлуатувати світильники зі знятими ковпаками (розсіювачами), використовувати саморобні подовжувачі, які не відповідають вимогам Правил улаштування електроустановок;

- пристосовувати вимикачі, штепсельні розетки для підвішування одягу та інших предметів, обгортати електролампи й світильники, заклеювати ділянки електропроводки горючою тканиною, папером;

- використовувати побутові електрокип'ятильники, чайники і таке інше (окрім приміщень, спеціально відведених та обладнаних для цього), залишати без нагляду увімкненими в електромережу кондиціонери, комп'ютери, лічильні та друкарські машинки тощо;

- захаращувати підступи до засобів пожежогасіння, використовувати пожежні крани, рукави й пожежний інвентар не за призначенням, зберігати документи, різні матеріали, предмети та інвентар у шафах (нішах) інженерних комунікацій;

- палити (окрім спеціально відведених для цього адміністрацією місць, позначених написом «Місце для паління» та забезпечених урною чи попільницею з негорючого матеріалу), проводити зварювальні та інші вогневі роботи без оформлення відповідального дозволу, застосовувати легкозаймисті рідини.

ВИСНОВКИ

Пристрій, спроектований в цій роботі, має свою особливість, а саме відсутність конкурентів на ринку. Особливістю є те що він має коефіцієнт спотворень 7%. Всі важливі функції виконує мікроконтролер ATMEGA8, а саме приймання сигналів з USB та керування реле, що відповідає за стан зарядного пристрою: ввімкнений або вимкнений.

Важливим моментом було живлення, передача сигналів інформаційних виводів та виводу Reset. Живлення було забезпечене за допомогою кабеля USB та стабілізатора AMS1117-3.3, для погашення шумів було використано по 2 конденсатори: перший електrolітичний для усунення загальних завад, другий для усунення завад електrolітичного конденсатора. Інформаційні виходи USB підключені через резистори 68Ом цю інформацію було знайдено саме на форумах радіоаматорів, до інформації були додані практичні докази. Вивід МК Reset був під'єднаний до виходу USB D-, а також з'єднаний з виходом стабілізатора для забезпечення потрібного рівня сигналу.

Тактовий резонатор був створений з частотою 12МГц. Бібліотека вимагала частоти 15МГц або 12МГц. Методом вибору було обрано на 12МГц.

Індикація виконана двома світлодіодами червоним і зеленим. Коли зелений блимає тричі – це свідчить про старт роботи, коли горить зелений то зарядний пристрій працює, коли горить червоний, значить сталася помилка та зарядний пристрій непрацює.

Також для захисту від перевантажень використані 2 запобіжники FU1 на 250мА та FU2 на 1А.

Обравши прикладом «Lenovo ThinkPad Slim 135W AC Adapter Slim tip» можна зазначити, що перевагою мого пристрою над обраним, є наявність ключової функції, що запобігає швидкій втраті ресурсу батареї ноутбука. Зарядне, що було обрано, є якісним пристроєм, що відповідає своїй ціні, але, все ж таки, поступається проектованому, в цьому аспекті.

СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ

1. Цирульник С., Ткачук В., Роптанов В. Прикладне програмування Embedded та IOT пристроїв «ІНТЕРНЕТ-ОСВІТА-НАУКА-2020», XII Міжнародна науково-практична конференція ІОН-2020, 26-29 травня, 2020: Збірник праць. Вінниця : ВНТУ, 2020, с. 243-245.
2. Новацький А. О. Мікропроцесорні та мікроконтролерні системи: підручник. Ч. 1. Мікропроцесорні системи. Київ: КПІ ім. Ігоря Сікорського, Вид-во «Політехніка», 2019. – 367 с.
3. Gary Willoughby. Purebasic A Beginner's Guide To Computer Programming. Aardvark Global Publishing", 2006. 336 с.
4. Новацький А.О. Мікропроцесорні та мікроконтролерні системи Частина 2. Проектування мікропроцесорних систем. Лабораторний практикум. Київ: КПІ ім. Ігоря Сікорського. 2021 268 с.
5. Нестерчук Д. М. Контрольно вимірювальні прилади з основами метрології. Конспекти лекцій. М. Видавничо-поліграфічний центр «Люкс». 2020. - 256 с.
6. Цирульник С. М., Азаров О. Д., Крупельницький Л. В., Трояновська Т. І. Програмування мікроконтролерів AVR. Навчальний посібник. Вінниця: ВНТУ, 2018. 110 с.
7. Поджаренко В.О., Кучерук В.Ю., Севастьянов В.М. Основи мікропроцесорної техніки. Навчальний посібник. - Вінниця: ВНТУ, 2006. - 226 с.
8. Цирульник С. М. Азаров О. Д., Крупельницький Л. В., Трояновська Т. І. Мікропроцесорна техніка. Навчальний посібник. Вінниця: ВНТУ, 2017. 123с.
9. Новацький А.О. Комп'ютерна електроніка; КПІ ім. Ігоря Сікорського. Київ: КПІ ім. Ігоря Сікорського, 2018. – 468 с.
10. Адаптер живлення Lenovo ThinkPad Slim 135W AC Adapter Slim tip (4X20Q88543) - [Електронний ресурс] - Режим доступу: <http://surl.li/tyeae> (дата звернення: 26.04.2024). — Назва з екрана.

11. Мультиметр M830B - [Електронний ресурс] - Режим доступу: <https://cutt.ly/xUicQLZ> (дата звернення: 26.04.2024). — Назва з екрана.
12. ДСТУ OHSAS 18002:2015. Системи управління гігієною та безпекою праці. Основні принципи виконання вимог OHSAS 18001:2007 (OHSAS 18002:2008, IDT). К. : ГП «УкрНИУЦ», 2016. 21.
13. ДСТУ 8604:2015 Дизайн і ергономіка. Робоче місце для виконання робіт у положенні сидячи. Загальні ергономічні вимоги. - [Електронний ресурс] - Режим доступу: http://online.budstandart.com/ua/catalog/doc-page?id_doc=71028 (дата звернення: 26.04.2024). — Назва з екрана.
14. НПАОП 0.00-7.15-18 Вимоги щодо безпеки та захисту здоров'я працівників під час роботи з екранними пристроями. - [Електронний ресурс] - Режим доступу: http://sop.zp.ua/norm_праор _0 _ 00-7_ 15-18_01_ua.php (дата звернення: 26.04.2024). — Назва з екрана.
15. Правила улаштування електроустановок - [Електронний ресурс] - Режим доступу: <http://www.energiy.com.ua/PUE.html> (дата звернення: 26.04.2024). — Назва з екрана.
16. Гігієнічна класифікація праці (за показниками шкідливості і небезпеки факторів виробничого середовища від 12.08.1986 № 4137-86. - [Електронний ресурс] - Режим доступу: <http://zakon4.rada.gov.ua/laws/show/v4137400-86> (дата звернення: 26.04.2024). — Назва з екрана.
17. СН 3.3.6.042-99 Санітарні норми мікроклімату виробничих приміщень. - [Електронний ресурс] - Режим доступу: <http://mozdocs.kiev.ua/view.php?id=1972> (дата звернення: 26.04.2024). — Назва з екрана.
18. ДБН В.2.5-28-2018 Природне і штучне освітлення - [Електронний ресурс] - Режим доступу: <http://surl.li/umfwl> (дата звернення: 26.04.2024). — Назва з екрана.
19. ДСН 3.3.6.037-99 Санітарні норми виробничого шуму, ультразвуку та інфразвуку. - [Електронний ресурс] - Режим доступу: <http://surl.li/umfwv> (дата звернення: 26.04.2024). — Назва з екрана.

20. ДБН В.1.1-7:2016 Пожежна безпека об'єктів будівництва. Загальні вимоги. URL: http://www.poliplast.ua/doc/dbn_v.1.1-7-2002.pdf (дата звернення: 26.04.2024). — Назва з екрана.

Додаток А
(обов'язковий)

ІЛЮСТРАТИВНИЙ МАТЕРІАЛ

РОЗРОБКА СИСТЕМИ АВТОМАТИЧНОГО РЕГУЛЮВАННЯ ПІДСИЛЕННЯ ШИРОКОСМУГОВОГО ЦИФРОВОГО ПРИЙМАЧА

Виконав: студент 2(4)-го курсу, групи ТКР-22мс
спеціальності, 172 – Телекомунікації та радіотехніка
(шифр і назва напрямку підготовки, спеціальності)

Чубатюк Ю.С.

(прізвище та ініціали)

Керівник: к.т.н., доц. каф. каф. ІРТС

Савицький А.Ю.

(прізвище та ініціали)

«13» 06 2024 р.

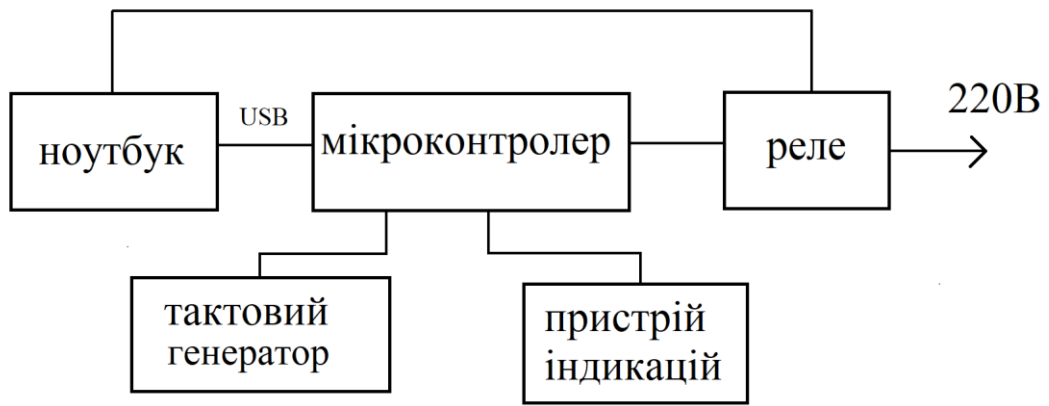


Рисунок 1 – Структурна схема

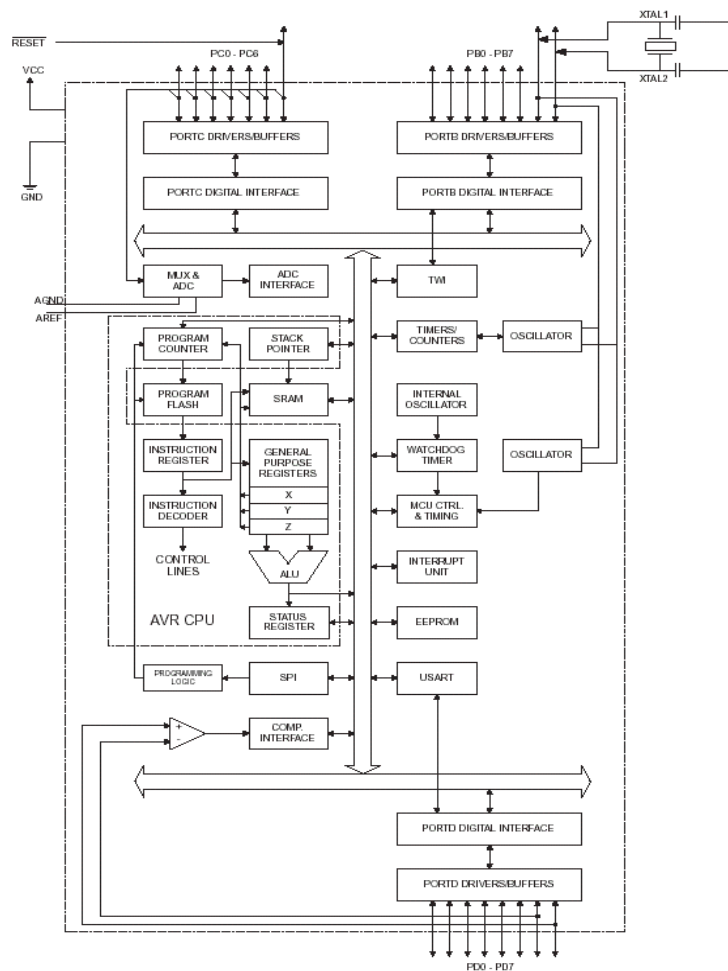


Рисунок 2 – Блок схема ATMEGA8

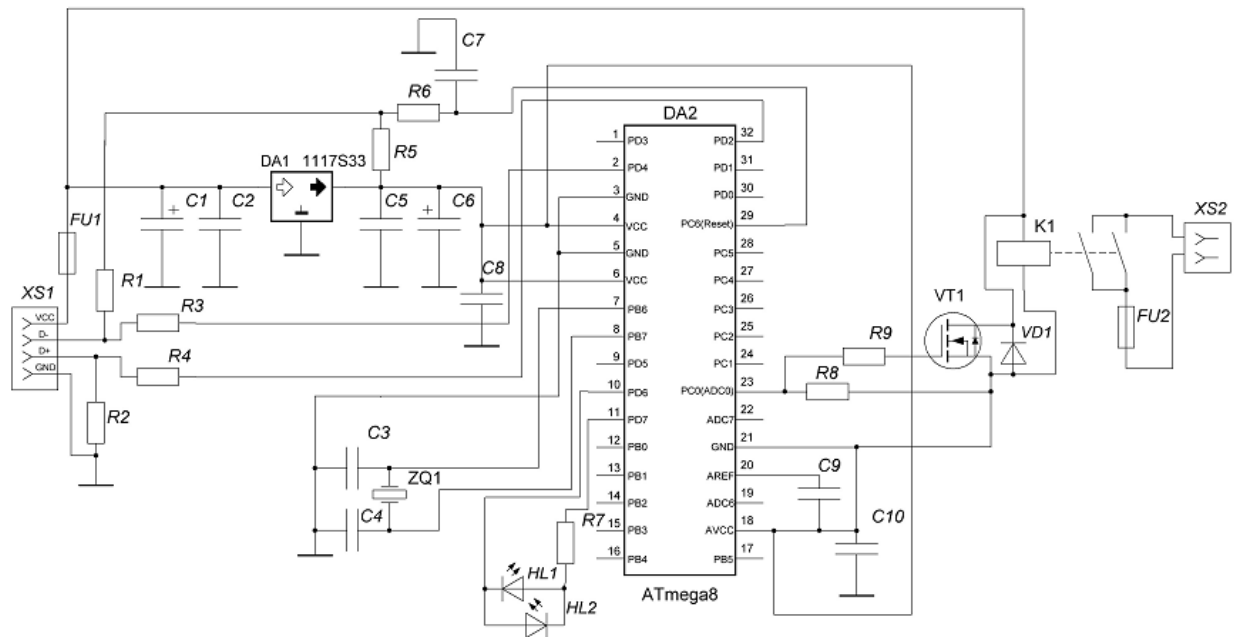


Рисунок 3 – Схема електрична принципова

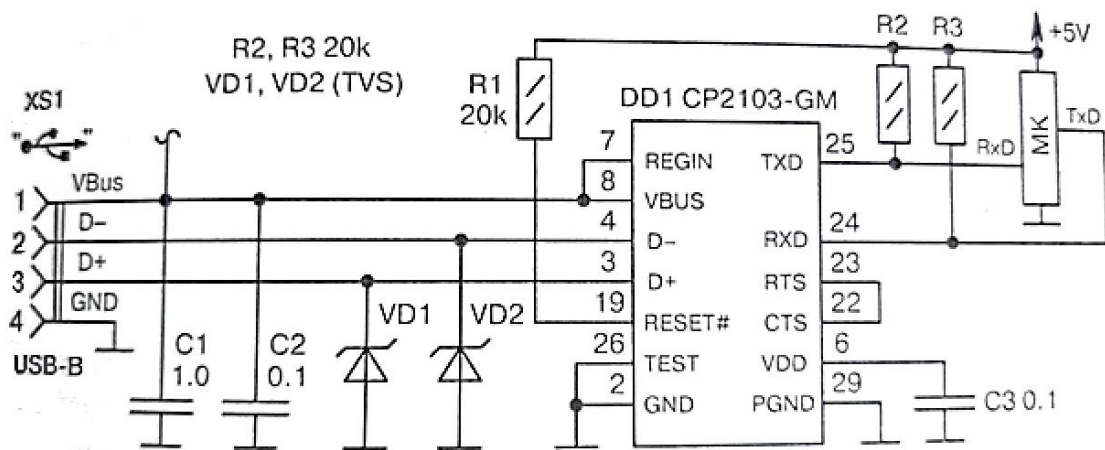


Рисунок 4 – Схема підключення USB за допомогою UART

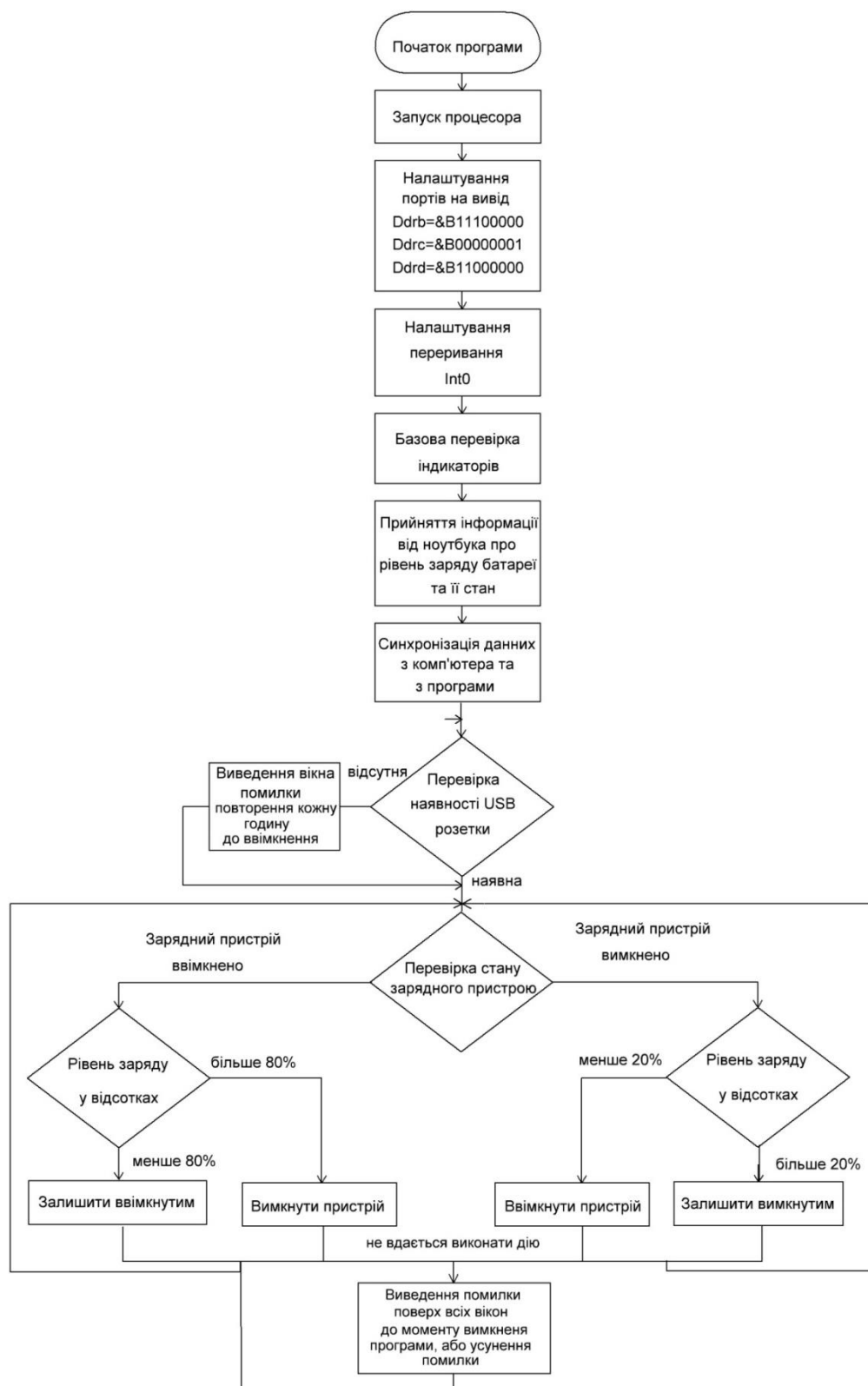


Рисунок 5 – Алгоритм роботи програми

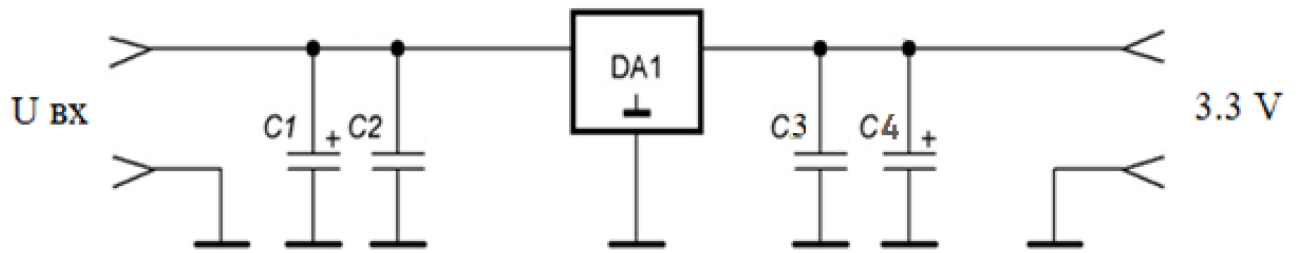


Рисунок 6 – схема стабілізатора напруги

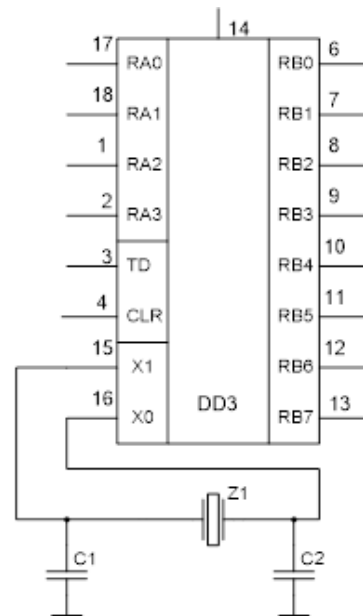


Рисунок 7 –Схема задаючого генератора

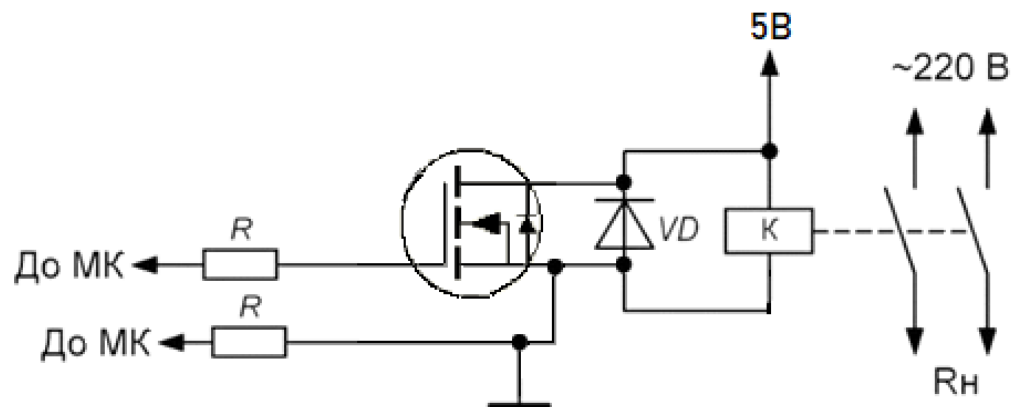


Рисунок 8 – Схема блоку реле

Додаток Б
(обов'язковий)

ПРОТОКОЛ ПЕРЕВІРКИ РОБОТИ

**РОЗРОБКА СИСТЕМИ АВТОМАТИЧНОГО РЕГУЛЮВАННЯ
ПІДСИЛЕННЯ ШИРОКОСМУГОВОГО ЦИФРОВОГО ПРИЙМАЧА**

ПРОТОКОЛ
ПЕРЕВІРКИ КВАЛІФІКАЦІЙНОЇ РОБОТИ
НА НАЯВНІСТЬ ТЕКСТОВИХ ЗАПОЗИЧЕНЬ

Назва роботи: «Розробка системи автоматичного регулювання підсилення широкосмугового цифрового приймача»

Тип роботи: БДР
(БДР, МКР)

Підрозділ ІРТС, ІЕС
(кафедра, факультет)

Показники звіту подібності Unichesk

Оригінальність 76,8% Схожість 23,2%

Аналіз звіту подібності (відмітити потрібне):

- ☒ 1. Запозичення, виявлені у роботі, оформлені коректно і не містять ознак плагіату.
- ☐ 2. Виявлені у роботі запозичення не мають ознак плагіату, але їх надмірна кількість викликає сумніви щодо цінності роботи і відсутності самостійності її виконання автором. Роботу направити на розгляд експертної комісії кафедри.
- ☐ 3. Виявлені у роботі запозичення є недобросовісними і мають ознаки плагіату та/або в ній містяться навмисні спотворення тексту, що вказують на спроби приховування недобросовісних запозичень.

Особа, відповідальна за перевірку 
(підпис)

Олександр ЗВЯГІН
(прізвище, ініціали)

Ознайомлені з повним звітом подібності, який був згенерований системою Unichesk щодо роботи.

Автор роботи 
(підпис)

Юрій ЧУБАТЮК
(прізвище, ініціали)

Керівник роботи 
(підпис)

Антон САВИЦЬКИЙ
(прізвище, ініціали)

Додаток В (довідниковий)

Лістинг Програми

```
'Test setup is on the STK500 with Vtarget set to 3.6V
'Using an atmega644 and external 12MHz crystal
'
'when you connect the RS232 SPARE to your COM port of your PC
'you must connect TXD SPARE to PORTD.1 (txd pin)

'Save about 38 bytes of code size
$noramclear

'Enable debugging output
'0 = disable
'1 = debugging messages
'2 = dbg frame/stack output
Const Usbdebug = 0

#if Usbdebug = 2
$dbg
#endif

$hwstack = 40
$swstack = 40
$framesize = 50

$regfile = "m8def.dat"

$crystal = 12000000

#if Usbdebug > 0
'$baud = 57600
'Open "coma.0:57600,8,n,1" For Output As #1
#endif

$eepromhex                                     'for
STK500 programmer

'Include the software USB library
$lib "swusb.lbx"
$external _swusb

Declare Sub Usb_reset()
Declare Sub Usb_processsetup(txstate As Byte)
Declare Sub Usb_send(txstate As Byte , Byval Count As Byte)
Declare Sub Usb_senddescriptor(txstate As Byte , Maxlen As Byte)

'***** Begin USB Configuration *****
```

```

'Set the following parameters to match your hardware configuration
and USB
'device parameters.

'***** USB Connections *****

'Define the AVR port that the two USB pins are connected to
_usb_port Alias Portd
_usb_pin Alias Pind
_usb_ddr Alias Ddrd

'Define the D+ and D- pins. (put D+ on an interrupt pin)
Const _usb_dplus = 2                                '4
Const _usb_dminus = 4                                '6

'Configure the pins as inputs
'Config Pind.2 = Input
'Config Pind.3 = Input
'Configure the pins as inputs

'Ddrb = &B11100000
'направление линий порта B: 1 - на вывод
'Portb = &B00011111
'исходное состояние линий B: 00000000
Ddrc = &B00000001
'направление линий порта C: 1 - на вывод
Portc = &B00000000
'исходное состояние линий C: 00000000
Ddrd = &B11000000
'направление линий порта D: 1 - на вывод
Portd = &B00000000
'исходное состояние линий D: 00000000

'disable pullups
_usb_port._usb_dplus = 0
_usb_port._usb_dminus = 0

'***** USB Configuration Constants *****

'Use EEPROM or FLASH to store USB descriptors
'1 = EEPROM, 0 = FLASH. Storing to EEPROM will reduce code size
slightly.
Const _usb_use_eeprom = 0

'Don't wait for sent packets to be ACK'd by the host before marking
the
'transmission as complete. This option breaks the USB spec but
improves
'throughput with faster polling speeds.
'This may cause reliability issues. Should leave set to 0 to be
safe.
Const _usb_assume_ack = 0

```

```

' ***** Device Descriptor *****

'USB Vendor ID and Product ID (Assigned by USB-IF)
Const _usb_vid = &H2012
Const _usb_pid = &H2112

'USB Device Release Number (BCD)
Const _usb_devrel = &H0001

'USB Release Spec (BCD)
Const _usb_spec = &H0110

'USB Device Class, subclass, and protocol (assigned by USB-IF).
'h00 = Class defined by interface. (HID is defined in the
interface)
'hFF = Vendor-defined class (You must write your own PC driver)
'See http://www.usb.org/developers/defined\_class for more
information
Const _usb_devclass = 0
Const _usb_devsubclass = 0
Const _usb_devprot = 0

'These are _indexes_ to UNICODE string descriptors for the
manufacturer,
'product name, and serial number. 0 means there is no descriptor.
Const _usb_imanufacturer = 1
Const _usb_iproduct = 2
Const _usb_iserial = 0

'Number of configurations for this device. Don't change this unless
'you know what you are doing. Ordinarily it should just be 1.
Const _usb_numconfigs = 1

' ***** Config Descriptor *****

'The number of interfaces for this device (Typically 1)
Const _usb_numifaces = 1

'Configuration Number (do not edit)
Const _usb_confignum = 1

'Index of UNICODE string descriptor that describes this config (0 =
None)
Const _usb_iconfig = 2
&H80 = device powered from USB bus.
'&HC0 = self-powered (has a power supply
,
)
Const _usb_powered = &H80                                '&HC0

'Required current in 2mA increments (500mA max)
Const _usb_maxpower = 150                                '150 *
2mA = 300mA

```

```

' ***** Interface Descriptor *****

'Number of interfaces for this device (1 or 2)
Const _usb_ifaces = 1

'Interface number
Const _usb_ifaceaddr = 0

'Alternate index
Const _usb_alterdate = 0

'Number of endpoints for this interface (excluding endp 0)
Const _usb_ifaceendpoints = 2

'USB Interface Class, subclass, and protocol (assigned by USB-IF).
'&h00 = RESERVED
'&hFF = Vendor-defined class (You must write your own PC driver)
'      Other values are USB interface device class. (such as HID)
'See   http://www.usb.org/developers/defined\_class   for   more
information
Const _usb_ifclass = 3
Const _usb_ifsubclass = 0
Const _usb_ifprotocol = 0

'Index to UNICODE string descriptor for this interface (0 = None)
Const _usb_iiface = 0

'HID class devices are things like keyboard, mouse, joystick.
'See http://www.usb.org/developers/hidpage/ for the specification,
'tools, and resources.

'Note that for a HID device, the device class, subclass, and
protocol
'must be 0. The interface class must be 3 (HID).
'Interface subclass and protocol must be 0 unless you are making a
'keyboard or a mouse that supports the predefined boot protocol.
'See pages 8 and 9 of the HID 1.11 specification PDF.
'Number of HID descriptors (EXCLUDING report and physical)
'If you are not making a HID device, then set this constant to 0
Const _usb_hids = 1

'BCD HID releasenum. Current spec is 1.11
Const _usb_hid_release = &H0111

'Country code from page 23 of the HID 1.11 specifications.
'Usually the country code is 0 unless you are making a keyboard.
Const _usb_hid_country = 0

'The number of report and physical descriptors for this HID
'Must be at least 1! All HID devices have at least 1 report
descriptor.
Const _usb_hid_numdescriptors = 1

```

```

'Use a software tool to create the report descriptor and $INCLUDE
it.

'Endpoint 0 is not included here. These are only for optional
'endpoints.
'Note: HID devices require 1 interrupt IN endpoint

'Address of optional endpoints (Must be > 0. comment-out to not use)
Const _usb_endp2addr = 1
Const _usb_endp3addr = 2

'Valid types are 0 for control or 3 for interrupt
Const _usb_endp2type = 3
Const _usb_endp3type = 3

'Directions are: 0=Out, 1=In. Ignored by control endpoints
Const _usb_endp2direction = 1
Const _usb_endp3direction = 0

'Polling interval (ms) for interrupt endpoints. Ignored by control
endpoints
' (Must be at least 10)
Const _usb_endp2interval = 200
Const _usb_endp3interval = 100

'The includes need to go right here--between the configuration
constants above
'and the start of the program below. The includes will
automatically calculate
'constants based on the above configuration, dimension required
variables, and
'allocate transmit and receive buffers. Nothing inside the includes
file needs
'to be modified.
$include "Const_swusb-includes.bas"

'Set all the variables, flags, and sync bits to their initial states
Call Usb_reset()

Const _usb_intf = Intf0
Config Int0 = Rising
On Int0 Usb_isr Nosave
Enable Int0
Enable Interrupts

'***** End Of USB Configuration*****

Dim Resetcounter As Word
Dim Idlemode As Byte
'Dim Buttons_current As Byte
'Dim Buttons_last As Byte

'Config Portb = Output

```

```

'Config Portc = Input

'Помигаем светодиодом статуса - контроллер стартовал
Waitms 200
Portd.7 = 1
Portd.6 = 0
Waitms 100
Portd.7 = 0
Portd.6 = 0
Waitms 200
Portd.7 = 1
Portd.6 = 0
Waitms 100
Portd.7 = 0
Portd.6 = 0
Waitms 200
Portd.7 = 1
Portd.6 = 0
Waitms 100
Portd.7 = 0
Portd.6 = 0

Do
    Resetcounter = 0

    'Check for reset here
    While _usb_pin._usb_dminus = 0
        Incr Resetcounter
        If Resetcounter = 1000 Then
            Call Usb_reset()
        End If
    Wend

    'Check for received data
    If _usb_status._usb_rxc = 1 Then
        If _usb_status._usb_setup = 1 Then
            'Process a setup packet/Control message
            Call Usb_processsetup(_usb_tx_status)
        ElseIf _usb_status._usb_endpl = 1 Then

            Portc.0 = _usb_rx_buffer(2).0
            Portd.6 = _usb_rx_buffer(2).6
            Portd.7 = _usb_rx_buffer(2).7

        End If
        'Reset the RXC bit and set the RTR bit (ready to receive a new
packet)
        _usb_status._usb_rtr = 1
        _usb_status._usb_rxc = 0
    End If

```

Loop

End

```
'          Do not change the order of the descriptors!
'
```

```
#if _usb_use_eeprom = 1
    $eeprom
#else
    $data
#endif
```

```
'Device Descriptor
_usb_devicedescriptor:
Data 18 , 18 , _usb_desc_device , _usb_spec1 , _usb_specch ,
_usb_devclass
Data _usb_devsubclass , _usb_devprot , 8 , _usb_vidl , _usb_vidh ,
_usb_pidl
Data _usb_pidh , _usb_devrell , _usb_devrelh , _usb_imanufacturer
Data _usb_iproduct , _usb_iserial , _usb_numconfigs
```

'Retrieving the configuration descriptor also gets all the interface and

'endpoint descriptors for that configuration. It is not possible to retrieve

'only an interface or only an endpoint descriptor. Consequently, this is a

'large transaction of variable size.

```
_usb_configdescriptor:
```

```
Data _usb_descr_total , 9 , _usb_desc_config , _usb_descr_totall
Data _usb_descr_totalh , _usb_numifaces , _usb_confignum ,
_usb_iconfig
Data _usb_powered , _usb_maxpower
```

```
'_usb_IFaceDescriptor
Data 9 , _usb_desc_iface , _usb_ifaceaddr , _usb_alterate
Data _usb_ifaceendpoints , _usb_ifclass , _usb_ifsubclass ,
_usb_ifprotocol
Data _usb_iiface
```

```
#if _usb_hids > 0
```

```
'_usb_HIDDescriptor
```

```
Data _usb_hid_descr_len , _usb_desc_hid , _usb_hid_releasel ,
_usb_hid_releaseh
Data _usb_hid_country , _usb_hid_numdescriptors
```

'Next follows a list of bType and wLength bytes/words for each report and

'physical descriptor. There must be at least 1 report descriptor. In practice,


```

'There are usually 0 physical descriptors and only 1 report
descriptor.
Data _usb_desc_report
Data 33 , 0
'End of report/physical descriptor list
#endif

#if _usb_endpoints > 1
'_usb_EndpointDescriptor
Data 7 , _usb_desc_endpoint , _usb_endp2attr , _usb_endp2type , 8 ,
0
Data _usb_endp2interval
#endif

#if _usb_endpoints > 2
'_usb_EndpointDescriptor
Data 7 , _usb_desc_endpoint , _usb_endp3attr , _usb_endp3type , 8 ,
0
Data _usb_endp3interval
#endif

#if _usb_hids > 0
_usb_hid_reportdescriptor:
Data 33                                     ' Length
= 33 bytes
Data &H06 , &H00 , &HFF                      '
Usage_page(vendor Defined Page 1)              '
Data &H09 , &H01                              '
Usage(vendor Usage 1)                          '
Data &HA1 , &H02                              '
Collection(logical)
Data &H09 , &H01                              '
Usage(pointer)                                '
Data &H15 , &H00                              '
Logical_minimum(0)                            '
Data &H25 , &HFF                              '
Logical_maximum(255)                          '
Data &H75 , &H08                              '
Report_size(8)                                '
Data &H95 , &H01                              '
Report_count(1)                              '
Data &H81 , &H02                              '
Input(data , Var , Abs)
Data &H09 , &H01                              '
Usage(pointer)                                '
Data &H15 , &H00                              '
Logical_minimum(0)                            '
Data &H26 , &HFF , 0                          '
Logical_maximum(255)                          '
Data &H75 , &H08                              '
Report_size(8)

```

```

    Data &H95 , &H01
    Report_count(1)
    Data &H91 , &H02
    Output(data , Var , Abs)

    Data &HC0
    End_collection

#endif

'Yes, they MUST be written like "t","e","s","t". Doing so pads them
with
'0's. If you write it like "test," I promise you it won't work.

'Default language descriptor (index 0)
'Дескриптор языка по умолчанию (индекс 0)
_usb_langdescriptor:
Data 4 , 4 , _usb_desc_string , 09 , 04      '&h0409
= English
'Data 4 , 4 , _usb_desc_string , 19 , 04      '&h0419
= Русский

'Manufacturer Descriptor (unicode)
'Дескриптор изготовителя (unicode)
_usb_mandescriptor:
Data 8 , 8 , _usb_desc_string

Data "M" , "a" , "x"                        ', 0 , 0

'Product Descriptor (unicode)
'Дескриптор продукта (unicode)
_usb_proddescriptor:
' USB-розетка для ноутбука
Data 50 , 50 , _usb_desc_string
Data &H55 , &H00 , &H53 , &H00 , &H42 , &H00 , &H2D , &H00 , &H40 ,
&H04
Data &H3E , &H04 , &H37 , &H04 , &H35 , &H04 , &H42 , &H04 , &H3A ,
&H04
Data &H30 , &H04 , &H20 , &H00 , &H34 , &H04 , &H3B , &H04 , &H4F ,
&H04
Data &H20 , &H00 , &H3D , &H04 , &H3E , &H04 , &H43 , &H04 , &H42 ,
&H04
Data &H31 , &H04 , &H43 , &H04 , &H3A , &H04 , &H30 , &H04 , &H00 ,
0

_usb_numdescriptor:
Data 4 , 4 , _usb_desc_string
Data "1" , 0 , 0

Sub Usb_processsetup(txstate As Byte)
Senddescriptor = 0
#if Usbdebug = 2
Dbg
#endif

```

```

'Control transfers reset the sync bits like so
Txstate = _usb_setup_sync

'These are the standard device, interface, and endpoint requests
that the
'USB spec requires that we support.
Select Case _usb_rx_buffer(2)
  'Standard Device Requests
  Case &B10000000:
    Select Case _usb_rx_buffer(3)
      CASE _usb_REQ_GET_STATUS:
        Case _usb_req_get_descriptor:
          Select Case _usb_rx_buffer(5)
            Case _usb_desc_device:
              #if Usbdebug = 1
                Print "GETDD"
              #endif
              'Send the device descriptor
              #if _usb_use_eeprom = 1
                Readeeprom          _usb_eepromaddr1
            ,
            _usb_devicedescriptor
              #else
                Restore _usb_devicedescriptor
              #endif
              Senddescriptor = 1
            Case _usb_desc_config:
              #if Usbdebug = 1
                Print "GETCD"
              #endif
              'Send the configuration descriptor
              #if _usb_use_eeprom = 1
                Readeeprom          _usb_eepromaddr1
            ,
            _usb_configdescriptor
              #else
                Restore _usb_configdescriptor
              #endif
              Senddescriptor = 1
            Case _usb_desc_string:
              Select Case _usb_rx_buffer(4)
                Case 0:
                  'Send the language descriptor
                  #if _usb_use_eeprom = 1
                    Readeeprom          _usb_eepromaddr1
            ,
            _usb_langdescriptor
                  #else
                    Restore _usb_langdescriptor
                  #endif
                  Senddescriptor = 1
                Case 1:
                  'Send the manufacturer descriptor
                  #if _usb_use_eeprom = 1
                    Readeeprom          _usb_eepromaddr1
            ,
            _usb_mandescriptor

```

```

        #else
            Restore _usb_mandescrptor
        #endif
        Senddescriptor = 1
    Case 2:
        'Send the product descriptor
        #if _usb_use_eeprom = 1
            Readeeprom          _usb_eepromaddr1
        _usb_proddescrptor
        #else
            Restore _usb_proddescrptor
        #endif
        Senddescriptor = 1
    End Select
End Select
'
    CASE _usb_REQ_GET_CONFIG:
End Select
Case &B00000000:
    Select Case _usb_rx_buffer(3)
    '
        CASE _usb_REQ_CLEAR_FEATURE:
    '
        CASE _usb_REQ_SET_FEATURE:
    Case _usb_req_set_address:
        #if Usbdebug = 1
            Print "SETA"
        #endif
        'USB status reporting for control writes
        Call Usb_send(txstate , 0)
        While Txstate._usb_txc = 0 : Wend
        'We are now addressed.
        _usb_deviceid = _usb_rx_buffer(4)
    '
        CASE _usb_REQ_SET_DESCRIPTOR:
    Case _usb_req_set_config:
        #if Usbdebug = 1
            Print "SETC"
        #endif
        'Have to do status reporting
        Call Usb_send(txstate , 0)
    End Select
'Standard Interface Requests
Case &B10000001:
    Select Case _usb_rx_buffer(3)
    '
        CASE _usb_REQ_GET_STATUS:
    '
        CASE _usb_REQ_GET_IFACE:
    Case _usb_req_get_descriptor
        '_usb_rx_buffer(4) is the descriptor index and (5) is
the type
        Select Case _usb_rx_buffer(5)
            Case _usb_desc_report:
                #if Usbdebug = 1
                    Print "GETRD"
                #endif
                #if _usb_use_eeprom = 1
                    Readeeprom          _usb_eepromaddr1
                _usb_hid_reportdescriptor

```

```

        #else
            Restore _usb_hid_reportdescriptor
        #endif
        Senddescriptor = 1
    '
        CASE _usb_DESC_PHYSICAL
    '
        CASE _USB_DESC_HID

        End Select
    End Select
'CASE &B00000001:
    'SELECT CASE _usb_rx_buffer(3)
    '    CASE _usb_REQ_CLEAR_FEATURE:

    '    CASE _usb_REQ_SET_FEATURE:

    '    CASE _usb_REQ_SET_IFACE:

    'END SELECT
'Standard Endpoint Requests
'CASE &B10000010:
    'SELECT CASE _usb_rx_buffer(3)
    '    CASE _usb_REQ_GET_STATUS:

    'END SELECT
'CASE &B00000010:
    'SELECT CASE _usb_rx_buffer(3)
    '    CASE _usb_REQ_CLEAR_FEATURE:

    '    CASE _usb_REQ_SET_FEATURE:

    'END SELECT

'Class specific requests (useful for HID)
'CASE &b10100001:
    'Class specific GET requests
    'SELECT CASE _usb_rx_buffer(3)
    '    CASE _usb_REQ_GET_REPORT:
    '    CASE _usb_REQ_GET_IDLE:
    '    CASE _usb_REQ_GET_PROTOCOL:
    'END SELECT
    '0-byte answer
    'Call Usb_Send(TxState, 0)
Case &B00100001:
    'Class specific SET requests
    Select Case _usb_rx_buffer(3)
        'CASE _usb_REQ_SET_REPORT:
        Case _usb_req_set_idle:
            Idlemode = 1
            'Do status reporting
            Call Usb_send(txstate , 0)
        'CASE _usb_REQ_SET_PROTOCOL:
    End Select
End Select
End Select

```

```

    If Senddescriptor = 1 Then
        Call Usb_senddescriptor(txstate , _usb_rx_buffer(8))
    End If

End Sub

Sub Usb_senddescriptor(txstate As Byte , Maxlen As Byte)
    #if Usbdebug = 2
        Dbg
    #endif
    'Break the descriptor into packets and send to TxState
    Local Size As Byte
    Local I As Byte
    Local J As Byte
    Local Timeout As Word

    #if _usb_use_eeprom = 1
        'EEPROM access is a little funky.  The size has already been
        fetched
        'and stored in _usb_EEPROMADDRL, and the address of the
        descriptor
        'is now in the EEAR register pair.

        Size = _usb_eepromaddrl

        'Fetch the location of the descriptor and use it as an address
        pointer
        push R24
        in R24, EEARL
        sts {_USB_EEPROMADDRL}, R24
        in R24, eearH
        sts {_USB_EEPROMADDRH}, R24
        pop R24

    #else
        Read Size
    #endif

    #if Usbdebug = 1
        Print Size
        Print Maxlen
    #endif

    If Maxlen < Size Then Size = Maxlen

    I = 2
    For J = 1 To Size
        Incr I
        #if _usb_use_eeprom = 1
            Incr _usb_eepromaddr
            Readeeprom Txstate(i) , _usb_eepromaddr
        #else
            Read Txstate(i)
        #endif
    Next J
End Sub

```

```

#endif

#if Usbdebug = 1
    Print Bin(txstate(i))
#endif

If I = 10 Or J = Size Then
    I = I - 2
    Call Usb_send(txstate , I)
    While Txstate._usb_txc = 0
        Timeout = 0
        'To prevent an infinite loop, check for reset here
        While _usb_pin._usb_dminus = 0
            Incr Timeout
            If Timeout = 1000 Then
                Call Usb_reset()
                Exit Sub
            End If
        Wend
    Wend
    I = 2
End If

Next
End Sub

Sub Usb_send(txstate As Byte , Byval Count As Byte)
#if Usbdebug = 2
    Dbg
#endif

    'Calculates and adds the CRC16,adds the DATAx PID,
    'and signals to the ISR that the data is ready to be sent.
    '
    '"Count" is the DATA payload size. Range is 0 to 8. Do not
    exceed 8!
    'Reset all the flags except TxSync and RxSync
    Txstate = Txstate And _usb_syncmask

    'Calculate the 16-bit CRC
    _usb_crc = 0
    If Count <> 0 Then
        _usb_crc = Crc16uni(txstate(3) , Count , &HFFFF , &H8005 , 1 ,
1)
        Toggle _usb_crc
    End If
    'Bytes to transmit will be PID + DATA payload + CRC16
    Count = Count + 3
    Txstate = Txstate + Count

    Txstate(count) = Low(_usb_crc)
    Incr Count
    Txstate(count) = High(_usb_crc)

```

```

'Add the appropriate DATAx PID
Txstate(2) = _usb_pid_data1
If Txstate._usb_txsync = 0 Then
    Txstate(2) = _usb_pid_data0
End If
'The last step is to signal that the packet is Ready To Transmit
Txstate._usb_rtt = 1
Txstate._usb_txc = 0
End Sub

Sub Usb_reset()
#If Usbdebug = 2
Dbg
#endif
'Reset the receive flags
_usb_status._usb_rtr = 1
_usb_status._usb_rxc = 0

'Reset the transmit flags
_usb_tx_status = _usb_endp_init
#If Vexist( "_usb_Endp2Addr")
_usb_tx_status2 = _usb_endp_init
#endif
#If Vexist( "_usb_Endp3Addr")
_usb_tx_status3 = _usb_endp_init
#endif

'Reset the device ID to 0
_usb_deviceid = 0

#If Usbdebug = 1
Print "RESET"
#endif

Idlemode = 0

End Sub

Enumeration
#ImageSave
#ImageDelete
#Close
#arrow_up
#arrow_down
#language
#led_on
#led_off
#led_err
EndEnumeration
;}-----
;{-----Глобальные переменные-----
#USB_PID=$2112
;идентификатор PID

```



```

#USB_VID=$2012
;идентификатор VID
Global Name.s="USB-розетка" ;сообщения
программы
Global NO_Device.s="Нет связи с USB-розеткой" ;сообщения
программы
Global YES_Device.s="USB-розетка подключена" ;сообщения
программы
Global Maska.s = "Заряд батареи"
Global Maska1.s = "Батарея не обнаружена"
Global Maska2.s = "Принудительно включить"
Global Maska3.s = "Принудительно выключить"
Global Maska4.s = "Порог включения зарядного устройства не может
быть больше порога выключения!"
Global Maska5.s = "Порог включения зарядного устройства не может
быть равен порогу выключения!"
Global Maska6.s = "Порог выключения зарядного устройства не может
быть равен порогу включения!"
Global Maska7.s = "Порог выключения зарядного устройства не может
быть меньше порога включения!"
Global Maska8.s = "Программа уже запущена!"
Global Maska9.s = "Проверьте зарядное устройство!"
Global Language.s
Global Lang.a
Global Charge_err.b=0
Global R_DeviceHandle, W_DeviceHandle
UsePNGImageDecoder()
IDI_EXCLAMATION = LoadIcon_(0, #IDI_EXCLAMATION)
;}-----

```

```

Procedure ProgramAtStartup(State, ProgName.s) ; Управление
автозагрузкой программы
valueName$=ProgramFilename()
GetHandle = RegOpenKeyEx_(#HKEY_CURRENT_USER,
"Software\Microsoft\Windows\CurrentVersion\Run", 0, #KEY_ALL_ACCESS,
@hKey)
If GetHandle = #ERROR_SUCCESS
If State=1
RegSetValueEx_(hKey, @ProgName, 0, #REG_SZ, @valueName$,
StringByteLength(valueName$))
Else
RegDeleteValue_(hKey, @ProgName)
EndIf
RegCloseKey_(hKey)
EndIf
EndProcedure

```

```

Procedure FindDevice_Timer(); таймер - процедура проверки
подключения нашего устройства
Static Old_Test
Test=HID_Lib_DeviceTest(#USB_PID, #USB_VID)
If Test<>Old_Test
Old_Test=Test
If Test

```

```

        HID_Lib_CloseDevice(R_DeviceHandle)                                :
HID_Lib_CloseDevice(W_DeviceHandle)
        W_DeviceHandle=HID_Lib_OpenDevice(#USB_PID, #USB_VID)
        R_DeviceHandle=HID_Lib_OpenDevice(#USB_PID, #USB_VID)
        SetGadgetText(10,YES_Device)
    Else
        HID_Lib_CloseDevice(R_DeviceHandle)                                :
HID_Lib_CloseDevice(W_DeviceHandle)
        R_DeviceHandle=0 : W_DeviceHandle=0
        SetGadgetText(10,NO_Device)
    EndIf
EndIf
EndProcedure

Procedure SendDevice(Command.a); Передача данных устройству
    Dim OutBuffer.a(2)
    If GetGadgetState(5)=1
        Command | %10000001
        SetGadgetText(5, Maska3)
        SetGadgetState(39, ImageID(#led_on))
    Else
        If Charge_err = 1
            Command | %01000001
            SetGadgetState(39, ImageID(#led_err))
        Else
            Command & %01111110

            SetGadgetText(5, Maska2)
            SetGadgetState(39, ImageID(#led_off))
        EndIf
        Debug Charge_err
    EndIf
    If W_DeviceHandle
        OutBuffer(0) = 0
        OutBuffer(1) = Command
        HID_Lib_WriteDevice(W_DeviceHandle, @OutBuffer(), 2)
    Else
        MessageRequester(Name, NO_Device, 16)
    EndIf
EndProcedure

Procedure BatteryPercent()
Static Percent
BatteryPercent=GetBatteryLifePercent()
If BatteryPercent<>Percent
    Percent=BatteryPercent
If BatteryPercent
    SetGadgetText(57,Maska+" "+Str(GetBatteryLifePercent())+"%")
    If GetBatteryLifePercent() = -1
        SetGadgetText(57,Maska1)
    EndIf

If GetBatteryLifePercent() =< Val(GetGadgetText(34))
    SetGadgetState(5, 1)

```

```

        Command | %10000001
    ElseIf GetBatteryLifePercent() >= Val(GetGadgetText(35))
        SetGadgetState(5, 0)
        Command & %01111110
    EndIf
    SendDevice(Command)

    EndIf
EndIf
EndProcedure

Procedure Open_Window()
    Shared IDI_EXCLAMATION
    OpenWindow(1,0,0,240,55,Name,#PB_Window_TitleBar|#PB_Window_ScreenCe
ntered)
    StickyWindow(1,1)
    MessageBeep_(#MB_ICONHAND)
    ImageGadget(0,15, 10, 32, 32, IDI_EXCLAMATION)
    TextGadget(22,60,20,170,30, Maska9, #PB_Text_Center)

    AddWindowTimer(1, 200, 15000)
EndProcedure

Procedure BatteryChargeStatus()
    Static Status = -1
    ChargeStatus = GetBatteryChargeStatus() & 8
    If ChargeStatus <> Status
        Status = ChargeStatus
        If ChargeStatus
            Charge_err=0
            DisableGadget(5, 0)
            DisableGadget(34, 0)
            DisableGadget(35, 0)
            SetGadgetState(5, 1)
            If IsWindow(1)
                CloseWindow(1)
            EndIf
        Else
            If GetBatteryLifePercent() > Val(GetGadgetText(34)) And
GetBatteryLifePercent() < Val(GetGadgetText(35))
                Charge_err=1
                DisableGadget(34, 1)
                DisableGadget(35, 1)
                SetGadgetState(5, 0)
                Command | %01000001
                Open_Window()
                DisableGadget(5, 1)
            EndIf
        EndIf
        SendDevice(Command)

    EndIf

EndProcedure

```

```

Procedure Write_setting() ;Запись в файл настроек

    If CreatePreferences("settings.ini")

        WritePreferenceString ("On", GetGadgetText(34))
        WritePreferenceString ("Off", GetGadgetText(35))
        WritePreferenceString("Language", Language)

    ClosePreferences()
    EndIf
EndProcedure

Procedure LoadText(File.s)

    ;Получает полный путь запущенного приложения
    Result$ = ProgramFilename()

    ;Извлекает путь из полного пути. Например, если полный путь
    "C:\PureBasic\PB.exe", результат будет "C:\PureBasic\".
    Path$ = GetPathPart(Result$)
    File = Path$+"Files\Language\"+File

    OpenPreferences(File)
    CreatePopupMenu(68, #PB_Menu_ModernLook)
    MenuItem(1, ReadPreferenceString("mark_0", "Развернуть"),
ImageID(#arrow_up))
    MenuItem(2, ReadPreferenceString("mark_1", "Свернуть"),
ImageID(#arrow_down))
    OpenSubMenu(ReadPreferenceString("mark_2", "Язык"),
ImageID(#language))

    Directory$=Path$+"Files\Language\"

    If ExamineDirectory(0, Directory$, "*.lng")
    Lang=6
        While NextDirectoryEntry(0)
            MenuItem(Lang, Left(DirectoryEntryName(0),
Len(DirectoryEntryName(0))-4))
                Lang=Lang+1
            Wend
        FinishDirectory(0)
    EndIf
    CloseSubMenu()
    MenuBar()
    MenuItem(500, ReadPreferenceString("mark_3", "Выход"),
ImageID(#Close))

    SetWindowTitle(0, ReadPreferenceString("mark_4", "USB-розетка"))
    SysTrayIconToolTip(67, ReadPreferenceString("mark_4", "USB-
розетка"))
    Name = ReadPreferenceString("mark_4", "USB-розетка")
    NO_Device = ReadPreferenceString("mark_5", "Нет связи с USB-
розеткой")

```

```

YES_Device          =      ReadPreferenceString("mark_6","USB-розетка
подключена")
    If HID_Lib_DeviceTest(#USB_PID, #USB_VID)
        SetGadgetText(10, YES_Device)
    Else
        SetGadgetText(10, NO_Device)
    EndIf

Maska = ReadPreferenceString("mark_7","Заряд батареи")
    If GetBatteryLifePercent()
        SetGadgetText(57,Maska+" "+Str(GetBatteryLifePercent())+"%")
    EndIf

Maska1 = ReadPreferenceString("mark_8","Батарея не обнаружена")
    If GetBatteryLifePercent() = -1
        SetGadgetText(57,Maska1)
    EndIf

SetGadgetText(55, ReadPreferenceString("mark_9","Включить"))
SetGadgetText(56, ReadPreferenceString("mark_10","Выключить"))
Maska2 = ReadPreferenceString("mark_11","Принудительно включить")
Maska3 = ReadPreferenceString("mark_12","Принудительно выключить")
If GetGadgetState(5)=1
    SetGadgetText(5, Maska3)
Else
    SetGadgetText(5, Maska2)
EndIf

Maska4 = ReadPreferenceString("mark_13","Порог включения зарядного
устройства не может быть больше порога выключения!")

GadgetToolTip(5,      ReadPreferenceString("mark_14","Принудительное
включение/отключение зарядного устройства"))
GadgetToolTip(34, ReadPreferenceString("mark_15","Порог  включения
зарядного устройства"))
GadgetToolTip(35, ReadPreferenceString("mark_16","Порог  выключения
зарядного устройства"))
GadgetToolTip(50,      ReadPreferenceString("mark_17","Добавить      в
автозагрузку"))
GadgetToolTip(51,      ReadPreferenceString("mark_18","Удалить      из
автозагрузки"))

Maska5 = ReadPreferenceString("mark_19","Порог включения зарядного
устройства не может быть равен порогу выключения!")
Maska6      =      ReadPreferenceString("mark_20","Порог      выключения
зарядного устройства не может быть равен порогу включения!")
Maska7      =      ReadPreferenceString("mark_21","Порог      выключения
зарядного устройства не может быть меньше порога включения!")
Maska8 = ReadPreferenceString("mark_22","Программа уже запущена!")
Maska9      =      ReadPreferenceString("mark_23","Проверьте      зарядное
устройство!")

ClosePreferences()

```

```

EndProcedure

Procedure SetWindow_CloseButton(hWnd, Enabled) ; Отключаем кнопку
"Закреть"
  If (Enabled = #True Or Enabled = #False) And hWnd
    hSysMenu = GetSystemMenu_(hWnd, #False)
    If hSysMenu
      EnableMenuItem_(hSysMenu, #SC_CLOSE, #MF_BYCOMMAND | Enabled !
1)
      DrawMenuBar_(hWnd)
    EndIf
  EndIf
EndProcedure

CatchImage(#ImageSave, ?ImageSave)
CatchImage(#ImageDelete, ?ImageDelete)
CatchImage(#Close, ?Close)
CatchImage(#arrow_up, ?arrow_up)
CatchImage(#arrow_down, ?arrow_down)
CatchImage(#language, ?language)
CatchImage(#led_on, ?led_on)
CatchImage(#led_off, ?led_off)
CatchImage(#led_err, ?led_err)
; }-----
;-----Создадим окно формы-----
hWnd=OpenWindow(0, 0, 0, 220, 79, Name,
#PB_Window_MinimizeGadget|#PB_Window_ScreenCentered|#PB_Window_Invisi
ble)
SetWindow_CloseButton(hWnd, 0)

CompilerIf #PB_Editor_CreateExecutable
  ProgramAtStartup(1, "USB-розетка");"Добавить в автозагрузку"
CompilerEndIf

StickyWindow(0,1);поверх всех открытых окон

;-----Системный трей-----
AddSysTrayIcon(67, WindowID(0), ExtractIcon_(0,ProgramFilename(),0))
SysTrayIconToolTip(67, Name)
;Добавим к иконке меню
CreatePopupMenu(68, #PB_Menu_ModernLook)
MenuItem(1, "Развернуть", ImageID(#arrow_up))
MenuItem(2, "Свернуть", ImageID(#arrow_down))
OpenSubMenu("Язык", ImageID(#language))

;Получает полный путь запущенного приложения
Result$ = ProgramFilename()
;Извлекает путь из полного пути. Например, если полный путь
"C:\PureBasic\PB.exe", результат будет "C:\PureBasic\".
Path$ = GetPathPart(Result$)
;Просмотр всех элементов директории запущенного приложения (без
поддиректорий)
Directory$ = Path$+"Files\Language\"
If ExamineDirectory(0, Directory$, "*.lng")

```

```

    Lang=6
    While NextDirectoryEntry(0)
        MenuItem(Lang, Left(DirectoryEntryName(0),
Len(DirectoryEntryName(0))-4))
        Lang=Lang+1
    Wend
    FinishDirectory(0)
EndIf

CloseSubMenu()
MenuBar()
MenuItem(500, "Выход", ImageID(#Close))

TextGadget(10,0,0,220,13,NO_Device, #PB_Text_Center)

ButtonGadget(5,125,39,90,35,"",#PB_Button_Toggle|#PB_Button_MultiLin
e)
GadgetToolTip(5, "")

TextGadget(55, 5, 35, 60 ,15, "")
TextGadget(56, 5, 55, 60 ,15, "")

TextGadget(57, 5, 15, 125 ,15, "")

    ComboBoxGadget(34, 65, 33, 55, 20)
    GadgetToolTip(34, "")
    For a = 5 To 100 Step 5
        AddGadgetItem(34, -1,Str(a)+"%")
        SetGadgetState(34,a-5)
    Next a

    ComboBoxGadget(35, 65, 54, 55, 20)
    GadgetToolTip(35, "")
    For i = 5 To 100 Step 5
        AddGadgetItem(35, -1,Str(i)+"%")
        SetGadgetState(35,i-5)
    Next i

ButtonImageGadget(50,164,14,23,23, ImageID(#ImageSave))
GadgetToolTip(50, "")
ButtonImageGadget(51,192,14,23,23, ImageID(#ImageDelete))
GadgetToolTip(51, "")

ImageGadget(39,140,18,15,15, ImageID(#led_off))

;-----Чтение настроек-----
;{-----
OpenPreferences("settings.ini")

SetGadgetText(34, ReadPreferenceString("On", "5%"))
SetGadgetText(35, ReadPreferenceString("Off", "50%"))
Language=ReadPreferenceString("Language", "Русский")

ClosePreferences()

```

```

; }-----

LoadText (Language+".lng")

;-----Запрет на запуск программы более одного раза-----
; {-----
a = CreateSemaphore_(#Null,0,1,"USB-розетка")
If a<>0 And GetLastError_()= #ERROR_ALREADY_EXISTS
    CloseHandle_(a)
    If MessageRequester(Name, Maska8, #MB_OK|#MB_ICONWARNING)=#IDOK
        End
    EndIf
EndIf
; }-----

    If Val(GetGadgetText(34)) > Val(GetGadgetText(35))
        MessageRequester(Name, Maska4, 16)
        SetGadgetState(35, GetGadgetState(34)+1)
    EndIf
    ;
    If Val(GetGadgetText(34)) = Val(GetGadgetText(35))
        MessageRequester(Name, Maska5, 16)
        SetGadgetState(34, GetGadgetState(35)-1)
    EndIf
    If Val(GetGadgetText(34)) = 100
        SetGadgetState(34, GetGadgetState(35)-1)
    EndIf
    If Val(GetGadgetText(35)) = Val(GetGadgetText(34))
        MessageRequester(Name, Maska6, 16)
        SetGadgetState(35, GetGadgetState(34)+1)
    EndIf
    If Val(GetGadgetText(35)) < Val(GetGadgetText(34))
        MessageRequester(Name, Maska7, 16)
        SetGadgetState(34, GetGadgetState(35)-1)
    EndIf
    If Val(GetGadgetText(35)) = 5
        SetGadgetState(34, GetGadgetState(35)+1)
        SetGadgetState(35, GetGadgetState(34)+1)
    EndIf

;Галка в меню выбора языка.
For i=6 To Lang
    If Language = GetMenuItemText(68, i)
        SetMenuItemState(68, i, 1)
        Break
    EndIf
Next i

FindDevice_Timer()
BatteryChargeStatus()
SetTimer_(hwnd, 1, 400, @FindDevice_Timer())
SetTimer_(hwnd, 2, 500, @BatteryPercent())
SetTimer_(hwnd, 3, 500, @BatteryChargeStatus())

;-----Главный цикл-----

```


Repeat

```

    Event = WaitWindowEvent()
    Window = EventWindow()
    Gadget = EventGadget()

;{-----Событие при закрытии окна-----
If Event=#PB_Event_CloseWindow ;
    SetGadgetState(5, 0)
    Command & %01111110
    SendDevice(Command)
    Write_setting()
EndIf
;}-----

If Event=#PB_Event_Gadget
    Select EventGadget()

        Case 5
            SendDevice(Command)

        Case 34
            GetGadgetText(34)
            If Val(GetGadgetText(34)) > Val(GetGadgetText(35))
                MessageRequester(Name, Mask4, 16)
                SetGadgetState(35, GetGadgetState(34)+1)
            EndIf ;
            If Val(GetGadgetText(34)) = Val(GetGadgetText(35))
                MessageRequester(Name, Mask5, 16)
                SetGadgetState(34, GetGadgetState(35)-1)
            EndIf
            If Val(GetGadgetText(34)) = 100
                SetGadgetState(34, GetGadgetState(35)-1)
            EndIf
            Write_setting()
            If GetBatteryLifePercent() > Val(GetGadgetText(34))
                SetGadgetState(5, 0)
                Command & %01111110
            EndIf
            SendDevice(Command)

        Case 35
            GetGadgetText(35)
            If Val(GetGadgetText(35)) = Val(GetGadgetText(34))
                MessageRequester(Name, Mask6, 16)
                SetGadgetState(35, GetGadgetState(34)+1)
            EndIf
            If Val(GetGadgetText(35)) < Val(GetGadgetText(34))
                MessageRequester(Name, Mask7, 16)
                SetGadgetState(34, GetGadgetState(35)-1)
            EndIf
            If Val(GetGadgetText(35)) = 5
                SetGadgetState(34, GetGadgetState(35)+1)
                SetGadgetState(35, GetGadgetState(34)+1)
            EndIf
    EndSelect
EndIf

```

```

EndIf
Write_setting()
If GetBatteryLifePercent() < Val(GetGadgetText(35))
SetGadgetState(5, 1)
Command | %10000001
EndIf
SendDevice(Command)

Case 50
    ProgramAtStartup(1, "USB-розетка");Добавить в автозагрузку

Case 51
    ProgramAtStartup(0, "USB-розетка");Удалить из автозагрузки

EndSelect
EndIf

If Event = #PB_Event_SysTray

    If EventType() = #PB_EventType_RightClick ; Обработка правой
кнопки мышки
        DisplayPopupMenu(68, WindowID(0)) ; показ всплывающего меню
    EndIf

EndIf

If Event = #PB_Event_Menu
    Select EventMenu()
        Case 1 ;"Развернуть"
            HideWindow(0, 0)
            SetWindowState(0, #PB_Window_Normal)
        Case 2 ;"Свернуть"
            HideWindow(0, 1)
        Case 6 To Lang ;Язык
            Menu = EventMenu()
            For i=6 To Lang
                If i = Menu
                    Language=GetMenuItemText(68, i)
                    LoadText(GetMenuItemText(68, i)+".lng")
                    SetMenuItemState(68,i,1)
                Else
                    SetMenuItemState(68,i,0)
                EndIf
            Next i

            Language=GetMenuItemText(68, Menu)
            Write_setting()
        Case 500 ;"Выход"
            SetGadgetState(5, 0)
            Command & %01111110
            SendDevice(Command)
            Write_setting()
            Break
    EndSelect
EndIf

```

```

EndIf

If Event=#PB_Event_MinimizeWindow ;Событие при сворачивании окна
    HideWindow(0,1) ;свернуть окно
EndIf

    If Event = #PB_Event_Timer
        If EventTimer() = 200
            MessageBeep_(#MB_ICONHAND)
        EndIf
    EndIf

Until Event=#PB_Event_CloseWindow
DataSection

    Close:
    IncludeBinary "Files\Icons\Close.ico"

    arrow_up:
    IncludeBinary "Files\Icons\arrow_up.png"

    arrow_down:
    IncludeBinary "Files\Icons\arrow_down.png"

    language:
    IncludeBinary "Files\Icons\language.png"

    ImageSave:
    IncludeBinary "Files\Icons\accept.png"

    ImageDelete:
    IncludeBinary "Files\Icons\delete.png"

    led_on:
    IncludeBinary "Files\Icons\led_on.PNG"

    led_off:
    IncludeBinary "Files\Icons\led_off.PNG"

    led_err:
    IncludeBinary "Files\Icons\led_err.PNG"

EndDataSection
;}

End

```