

МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ
СХІДНОУКРАЇНСЬКИЙ НАЦІОНАЛЬНИЙ УНІВЕРСИТЕТ ІМ. В. ДАЛЯ
ФАКУЛЬТЕТ ІНФОРМАЦІЙНИХ ТЕХНОЛОГІЙ ТА ЕЛЕКТРОНІКИ
КАФЕДРА КОМП'ЮТЕРНОЇ ІНЖЕНЕРІЇ

До захисту допускається
Завідувач кафедри
_____ Скарга-Бандурова І.С.
« ____ » _____ 2017 р.

ДИПЛОМНИЙ ПРОЕКТ БАКАЛАВРА

ПОЯСНЮВАЛЬНА ЗАПИСКА

НА ТЕМУ:

**«Драйвер підтримки послідовної шини USB
на базі мікроконтролера AVR»**

Освітньо-кваліфікаційний рівень “бакалавр”
Напрямок підготовки 6.050102 – “Комп’ютерна інженерія”

Керівник проекту:

(підпис)

Кардашук В. С.

(ініціали, прізвище)

Консультант з охорони праці:

(підпис)

Критська Я. О.

(ініціали, прізвище)

Студент:

(підпис)

Бойчук А. М.

(ініціали, прізвище)

Група:

КІ-13 ад

Севєродонецьк 2017

**СХІДНОУКРАЇНСЬКИЙ НАЦІОНАЛЬНИЙ УНІВЕРСИТЕТ
імені ВОЛОДИМИРА ДАЛЯ**

Факультет інформаційних технологій та електроніки
Кафедра комп'ютерної інженерії
Напрямок підготовки 6.050102 «Комп'ютерна інженерія»
(шифр і назва)

ЗАТВЕРДЖУЮ

Завідувач кафедри КІ
_____ І.С. Скарга-Бандурова
“ ____ ” _____ 2017 р.

**ЗАВДАННЯ
НА ДИПЛОМНИЙ ПРОЕКТ (РОБОТУ) СТУДЕНТУ**

Бойчук Артур Мирославович
(прізвище, ім'я, по батькові)

1. Тема проекту (роботи): «Драйвер підтримки послідовної шини USB на базі мікроконтролера AVR»
затверджена наказом по інституту від «15» травня 2017 р. № 124/48
2. Термін подання студентом закінченого проекту (роботи): 15.06.2017 р.
3. Початкові дані до проекту (роботи): матеріали переддипломної практики
4. Зміст розрахунково-пояснювальної записки (перелік питань, які підлягають розробці):
Розробити драйвер підтримки послідовної шини USB на базі мікроконтролерів AVR з застосуванням інструментального середовища налаштування AVR Studio 4.
Основна частина повинна містити постановку задачі, короткі теоретичні відомості, опис елементної бази, розробку програмного забезпечення, алгоритми роботи, додатки.
5. Перелік графічного матеріалу (з точною вказівкою обов'язкових креслень): немає

6. Консультанти розділів проекту (роботи)

Розділ	Прізвище, ініціали та посада консультанта	Підпис, дата	
		завдання видав	завдання прийняв
Охорона праці та безпека в надзвичайних ситуаціях	Критська Я. О., асистент. кафедри комп'ютерної інженерії		

7. Дата видачі завдання 15.05.2017 р.

Керівник

_____ (підпис)

Завдання прийняв до виконання

_____ (підпис)

КАЛЕНДАРНИЙ ПЛАН

№ з/п	Назва етапів дипломного проекту	Строк виконання етапів	Примітка
1.	Отримання завдання, збір матеріалів	15.05.17 – 16.05.17	
2.	Огляд літератури й обґрунтування необхідності розроблення	15.05.17– 18.05.17	
3.	Розроблення технічного завдання	21.05.17 – 20.05.17	
4.	Визначення конфігурації апаратних та програмних засобів	26.05.17 – 27.05.17	
5.	Розроблення функціональної схеми	23.04.17 – 24.05.17	
6.	Розроблення драйверу підтримки шини USB	25.05.17 – 13.06.17	
7.	Охорона праці і навколишнього середовища	13.06.17 – 14.06.17	
8.	Оформлення пояснювальної записки	14.06.17 – 15.06.17	

Студент

_____ (підпис)

Бойчук А. М.

_____ (прізвище та ініціали)

Керівник

_____ (підпис)

Кардашук В. С.

_____ (прізвище та ініціали)

РЕФЕРАТ

Пояснювальна записка до дипломного проекту бакалавра: 59 сторінок, 12 рисунків, 6 таблиць, 27 джерел.

У дипломному проекті розроблено драйвер підтримки послідовної шини USB на базі мікроконтролерів AVR.

Універсальна послідовна шина (USB) стала надзвичайно популярною за рахунок надання ряду зручностей кінцевим користувачам, наприклад, функція "Plug and Play", яка дозволяє ідентифікувати підключений пристрій без необхідності рестарту комп'ютера. Однак для розробників інтегрувати USB-інтерфейс у свої проекти виявилось більш складним у порівнянні, наприклад, з інтерфейсом RS232. Крім цього, на стороні ПК також повинен бути передбачений спеціальний драйвер пристрою. Однак, як правило, в сучасних ПК фізичний порт RS232 не встановлюється і заміщується на порти USB.

У дипломній роботі проводиться огляд та аналіз апаратних та програмних засобів реалізації послідовної шини USB з використанням мікроконтролерів AVR. За результатами дослідження сформульовані мета та завдання випускної роботи. Здійснена постановка задачі на розробку апаратної та програмної частини реалізації послідовної шини USB. Визначені шляхи реалізації поставленого завдання та елементна база з використанням мікроконтролерів AVR фірми Atmel.

У якості реалізації апаратної частини обрано мікроконтролера ATmega8 фірми Atmel для застосування в системах передачі інформації в засобах автоматизації керування технологічними процесами. Програмування пам'яті мікроконтролера проведено в інструментальній середовищі налаштування AVR Studio 4.

Для реалізації драйвера використано C++ Builder та Microsoft Visual C++.

ПОСЛІДОВНА ШИНА, АЛГОРИТМ, МІКРОКОНТРОЛЕР, ПЕРЕДАЧА ДАНИХ, НАЛАШТУВАННЯ.

Умови отримання дипломного проекту:

СНУ ім. Володимира Даля, пр. Центральний 59а, м. Сєверодонецьк, 93406.

ЗМІСТ

ВСТУП	6
1 ОГЛЯД ІСНУЮЧИХ АПАРАТНО-ПРОГРАМНИХ ЗАСОБІВ РЕАЛІЗАЦІЇ ПОСЛІДОВНОЇ ШИНИ USB З ЗАСТОСУВАННЯМ МІКРОКОНТРОЛЕРІВ	7
1.1 Загальні положення	7
1.2 . Принцип дії послідовної шини USB	9
1.3 Програмна реалізація послідовної шини USB	14
1.4 Постановка задачі на розроблення	14
2 РОЗРОБЛЕННЯ ДРАЙВЕРУ ПІДТРИМКИ ПОСЛІДОВНОЇ ШИНИ USB З ВИКОРИСТАННЯМ МІКРОКОНТРОЛЕРІВ AVR	15
2.1 Вибір елементної бази	15
2.2 Вибір програмного середовища налаштування та програмна реалізація	19
2.3 Алгоритм реалізації процедури прийому	21
2.4 Процедура оброблення переривання	22
2.5 Основний програмний цикл	23
2.6 Опис підпрограм	24
2.7 Програма для ПК	25
2.8 Використання бібліотеки AVR309.dll фірми Atmel	29
2.9 Похибка генерації швидкості універсального асинхронного передавача/приймача	29
2.10 Вихідний код програми для мікроконтролера AVR	30
3 ОХОРОНА ПРАЦІ ТА БЕЗПЕКА В НАДЗВИЧАЙНИХ СИТУАЦІЯХ	32
3.1 Загальні питання з охорони праці	32
3.2 Правові та організаційні основи охорони праці	32
3.3 Організаційно-технічні заходи з безпеки праці.....	33
3.4 Аналіз стану умов праці та вимоги до приміщення.....	33
3.5 Вимоги до організації робочого місця	34
3.6 Навантаження та напруженість процесу праці	35

3.7 Аналіз небезпечних та шкідливих факторів при роботі на персональному комп'ютері	36
3.8 Пожежна безпека	37
3.9 Електробезпека	39
3.10 Мікроклімат	39
3.11 Освітлення робочого місця	40
3.12 Шум, вібрація та електромагнітне випромінювання	43
3.13 Розрахунок захисного заземлення	44
Висновки до розділу 3	47
ВИСНОВКИ	48
ПЕРЕЛІК ПОСИЛАНЬ	49
ДОДАТОК А. USB драйвер для сімейства мікроконтролерів AVR	51

ВСТУП

Універсальна послідовна шина (USB) стала надзвичайно популярною за рахунок надання ряду зручностей кінцевим користувачам, наприклад, функція "Plug and Play", яка дозволяє ідентифікувати підключений пристрій без необхідності рестарту комп'ютера. Однак для розробників інтегрувати USB-інтерфейс у свої проекти виявилось більш складним у порівнянні, наприклад, з інтерфейсом RS232. Крім цього, на стороні ПК також повинен бути передбачений спеціальний драйвер пристрою. Як наслідок, інтерфейс RS232 залишається дуже популярним серед виробників кінцевих систем. Даний інтерфейс добре вивчений і в достатній мірі підтримується операційною системою. Однак, як правило, в сучасних ПК фізичний порт RS232 не встановлюється і замінюється на порти USB.

Реалізувати інтерфейс USB в зовнішньому пристрої можна двома способами:

За допомогою мікроконтролера, у якого інтерфейс USB реалізований апаратно. В цьому випадку необхідно знати, як працює USB і відповідно до цього 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 зі звичайними мікроконтролерами, які використовуються для зв'язку через "RS232".

Використання універсального перетворювача інтерфейсів: USB і будь-якого іншого. В якості іншого інтерфейсу зазвичай використовується RS232, 8-розрядна шина даних або шина TWI. В 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. Рішення, що розглядається в даному документі, ґрунтується на використанні 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 повільної швидкості - 12 Мбіт/сек, режим високої швидкості – 480 Мбіт/сек. Мікроконтролери AVR повністю відповідають вимогам для реалізації режиму USB.

1 ОГЛЯД ІСНУЮЧИХ АПАРАТНО-ПРОГРАМНИХ ЗАСОБІВ РЕАЛІЗАЦІЇ ПОСЛІДОВНОЇ ШИНИ USB З ЗАСТОСУВАННЯМ МІКРОКОНТРОЛЕРІВ

1.1 Загальні положення

Специфікація периферійної шини USB розроблена лідерами комп'ютерної і телекомунікаційної промисловості – Compaq, DEC, IBM, Intel, Microsoft, NEC і Northern Telecom – для підключення комп'ютерної периферії поза корпусом персонального комп'ютера (ПК) за стандартом plug'n'play, в результаті відпадає необхідність в установці додаткових плат у слоти розширення і переконфігуруванні системи. ПК, що мають шину USB, дозволяють підключати периферійні пристрої і здійснюють їх автоматичне конфігурування, як тільки пристрій фізично буде приєднано до ПК, і при цьому немає необхідності перезавантажувати або вимикати комп'ютер, а так само запускати програми установки і конфігурування. Шина USB дозволяє одночасно підключати послідовно до 127 пристроїв, таких, як монітори або клавіатури, що виконують роль додатково підключених компонентів або хабів (тобто пристрій, через який підключається ще декілька пристроїв).

Розробка специфікацій на шину USB проводиться в рамках міжнародної некомерційної організації USB Implementers Forum (USB-IF), що об'єднує розробників і виробників обладнання з шиною USB.

Для підключення периферійних пристроїв до шини USB використовується чотирихпровідний кабель, при цьому два дроти (вита пара) в диференціальному включенні використовуються для прийому і передачі даних, а два дроти - для живлення периферійного пристрою. Завдяки вбудованим лініям живлення USB дозволяє підключати периферійні пристрої без власного джерела живлення (максимальна сила струму, споживаного пристроєм по лініях живлення шини USB, не повинна перевищувати 500 мА, у USB 3.0 – 900 мА).

Кабель USB складається з 4 мідних провідників – 2 провідника живлення та 2 провідника даних в звитій парі – і заземленою обплетення (екрана).

Кабелі USB орієнтовані, тобто мають фізично різні наконечники «до пристрою» і «до хосту». Можлива реалізація USB пристрою без кабелю, з вбудованим в корпус наконечником «до хосту». Можливо і нероз'ємне вбудовування кабелю в пристрій, як в миша (стандарт забороняє це для пристроїв full і high speed, але виробники його порушують). Існують (хоч і заборонені стандартом) і пасивні USB подовжувачі, які мають роз'єми «від хоста» і «до хосту».

За допомогою кабелів формується інтерфейс між USB-пристроями і USB-хостом. В якості хоста виступає програмно-керований USB-контролер, який забезпечує функціональність всього інтерфейсу. Контролер, як правило, інтегрований в мікросхему південного моста, хоча може бути виконаний і в окремому корпусі. З'єднання контролера з зовнішніми пристроями відбувається через USB-концентратор (інші назви - хаб, розширювач). В силу того, що USB-шина має деревоподібну топологію, концентратор самого верхнього рівня називається кореневим (root hub). Він вбудований в USB-контролер і є його невід'ємною частиною.

Для підключення зовнішніх пристроїв до USB-концентратора в ньому передбачені порти, що закінчуються роз'ємами. До роз'ємів за допомогою кабельної системи можуть підключатися USB-пристрої або USB-хаби нижніх рівнів. Такі хаби – активні електронні пристрої (пасивних не буває), що обслуговують кілька власних USB-портів. За допомогою USB-концентраторів допускається до п'яти рівнів каскадування, не рахуючи кореневого. USB-інтерфейс дозволяє з'єднати між собою і два комп'ютери, але це вимагає наявності спеціальної електроніки, що емулює Ethernet-адаптер з драйверною підтримкою з обох сторін.

Пристрої можуть одержувати живлення від шини, але можуть і вимагати зовнішнє джерело живлення. За замовчуванням пристроїв гарантується струм до 100 мА, а після узгодження з хост-контроллером – до 500 мА. Підтримується і черговий режим для пристроїв і розгалужувачів по команді з шини зі зняттям основного живлення при збереженні чергового живлення і включенням по команді з шини.

USB підтримує «гаряче» підключення і відключення пристроїв. Це досягнуто збільшеною довжиною заземлювального контакту роз'єму по відношенню до сигнального. При підключенні кабелю USB першими замикаються заземлюючі контакти, потенціали корпусів двох пристроїв стають рівні і подальше з'єднання сигнальних провідників не призводить до перенапруг, навіть якщо пристрої живляться від різних фаз силової трифазної мережі.

На логічному рівні пристрій USB підтримує транзакції прийому і передачі даних. Кожен пакет кожної транзакції містить в собі номер кінцевої точки (endpoint) на пристрої. При підключенні пристрою драйвери в ядрі ОС читають з пристрою список кінцевих точок і створюють керуючі структури даних для спілкування з кожною кінцевою точкою пристрою. Сукупність кінцевих точок і структур даних в ядрі ОС називається каналом (pipe).

1.2 Принцип дії послідовної шини USB

Детальна інформація з фізичної USB-зв'язку наведена на сайті www.usb.org. Однак дана документація є дуже складною для засвоєння. Більш доступно інформація по USB-зв'язку викладена в документі "USB in a Nutshell. Making Sense of the USB Standard".

В рекомендаціях щодо шини USB обмежуються описом програми пристрою. Фізичний інтерфейс USB складається з чотирьох провідників: 2 для живлення зовнішніх пристроїв (V_{CC} 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 V_{CC} і GND . Сигнальні провідники називаються $DATA+$ і $DATA-$ і керують зв'язком між комп'ютером і пристроєм. Сигнали в цих провідниках є двонаправленими. Рівні напруги – диференціальні: коли $DATA+$ має високий рівень, тоді $DATA-$ знаходиться на низькому рівні. Проте, 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.

Відповідно до стандарту USB високий рівень на сигнальних провідниках повинен складати 3,0...3,6 В, при цьому, напруга живлення V_{CC} шини USB, що надходить від комп'ютера становить 4.4...5.25В. Таким чином, якщо мікроконтролер живиться безпосередньо від шини USB, то лінії даних повинні пройти через каскад перетворення рівнів для компенсації рівнів диференціальної напруги. Іншим рішенням може бути використання стабілізатора напруги, який знизить напругу V_{CC} до рівня 3.3В, при цьому, мікроконтролер буде працювати з цією зниженою напругою і, відповідно, генерувати знижені рівні напруг.

На рис. 1.1 наведені осцилограми сигналів низькошвидкісного драйвера USB.

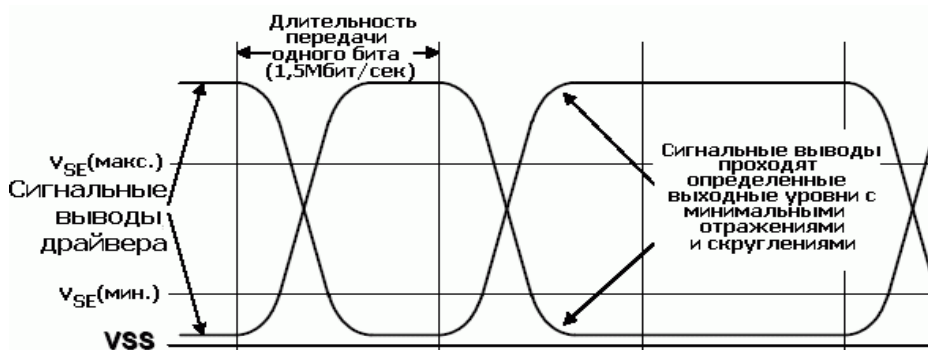


Рисунок 1.1– Осцилограми сигналів низькошвидкісного драйвера USB

На рис. 1.2 наведені рівні напруг при передачі пакетів по шині USB.

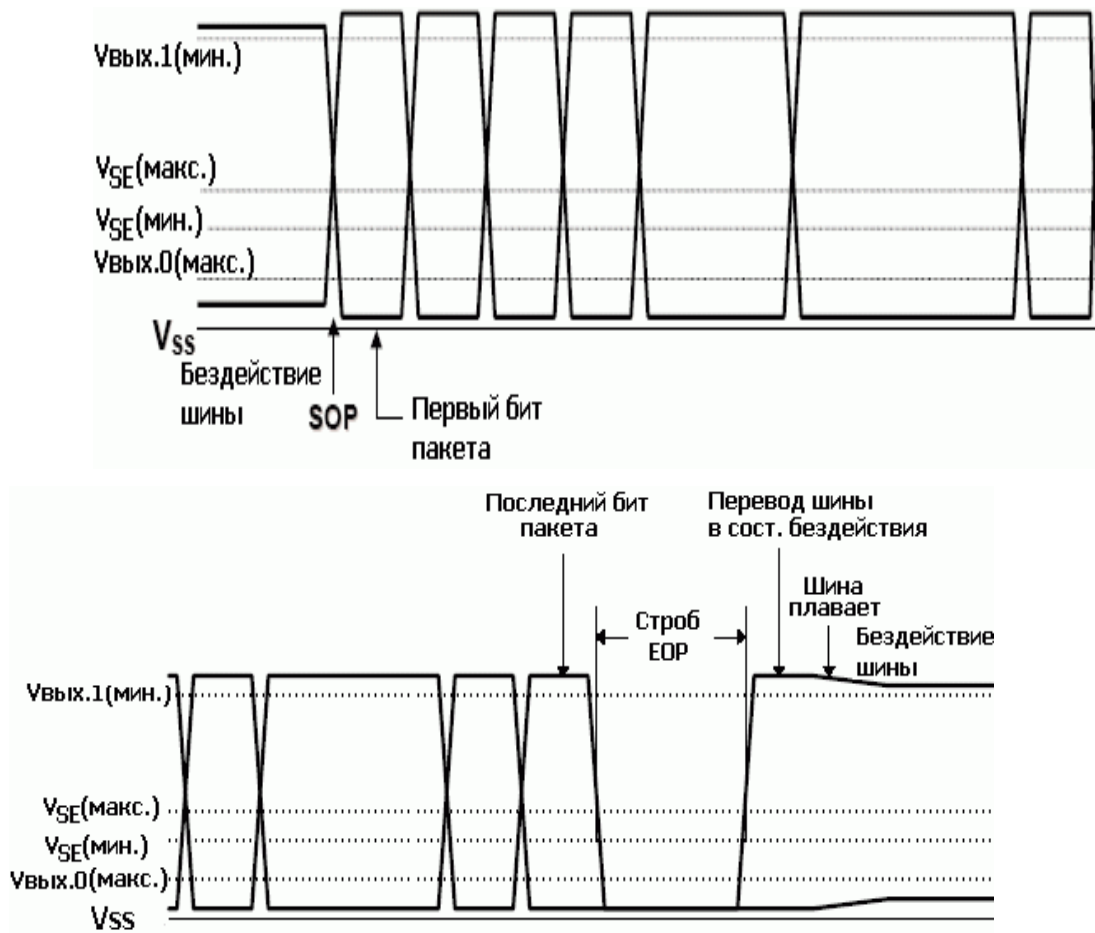


Рисунок 1.2 – Рівні напруг при передачі пакетів по шині USB

На рис. 1.3 наведено підключення кабелю і резисторів до пристрою USB.

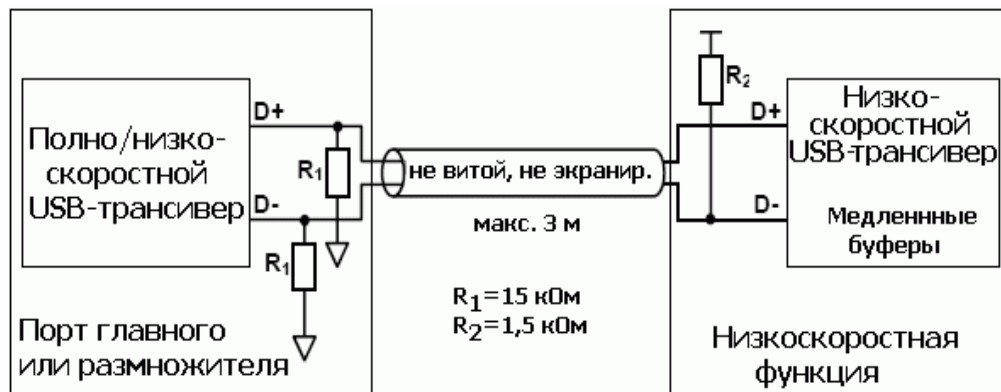


Рисунок 1.3 – Підключення кабелю і резисторів до пристрою

Принцип детекції підключення і відключення USB-пристрою заснований на контролі опору лінії USB. У низькошвидкісних USB-пристроях необхідний підтягуючий резистор між сигналом DATA– і Vcc. У повношвидкісних пристроях даний резистор підключається до DATA+.

Визначаючи, на якій лінії підключено підтягуючий резистор, головний комп'ютер визначає який новий пристрій підключено до лінії USB.

Після визначення нового пристрою головний комп'ютер починає зв'язок відповідно з фізичного протоколу USB. Протокол USB, на відміну від UART, 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.

На рис. 1.4 наведена синхронізація шини USB.



Рисунок 1.4 – Синхронізація шини USB

Для підтримки синхронізації потрібна передача зразкової синхронізації кожну мілісекунду у повношвидкісному режимі зв'язку, а в низькошвидкісному режимі кожну мілісекунду необхідно встановлювати низький рівень на обох сигнальних лініях.

В апаратно-реалізованому USB-приймачі дана синхронізація гарантується цифровим ФАПЧ (фазова автопідстроювання частоти). У даній реалізації період перетворення даних повинен бути синхронізований з зразковою синхронізацією, потім очікується два нулі, а потім починається процес прийому даних.

Прийом даних повинен задовольняти вимогу можливості засинхронізувати 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 одиночне зміна (один біт). Сигнали по лініях USB передаються в коді NRZI. У коді NRZI кожен 0 представляється шляхом зсуву поточного рівня сигналу, а кожна 1 шляхом утримання поточного рівня. На рівні

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

На рис. 5.1 наведено кодування даних у кодї NRZI.

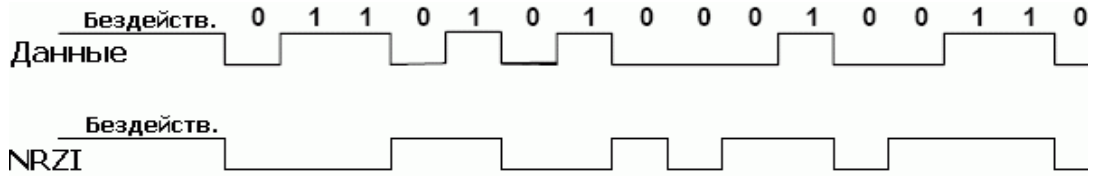


Рисунок 1.5 – Кодування даних у кодї NRZI

На рис. 1.6 наведена послїдовнїсть кодування даних.

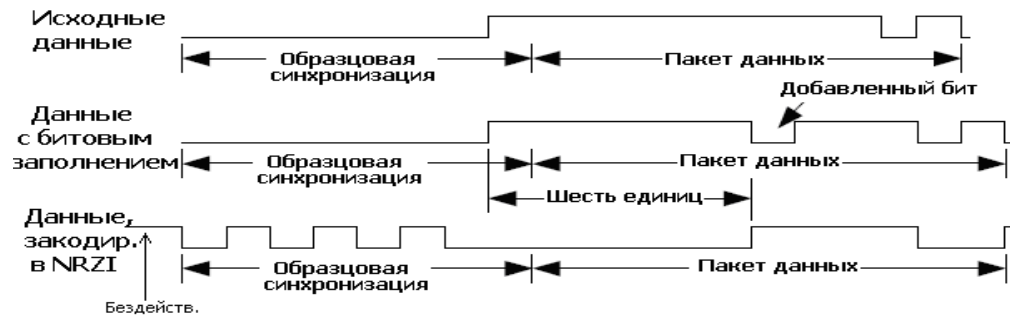


Рисунок 1.6 – Послїдовнїсть кодування даних

Повідомлення про закінчення передачі даних виконується за допомогою передачі сигналу "кінець пакета" (EOP). EOP передається шляхом установки низьких рівнів на обох лініях даних DATA+ і DATA-. EOP передається нетривалий час (мінімум два періоди швидкості даних). Після цього виконується наступна транзакція.

На рис. 1.7 наведена часова діаграма сигналу EOP

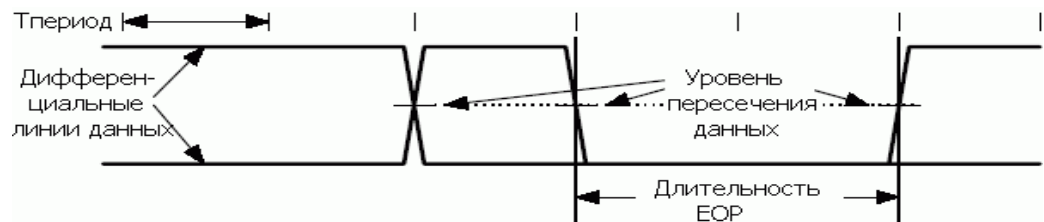


Рисунок 1.7 – Часова діаграма сигналу EOP

Дані, які передаються між зразковою синхронізацією і EOP, закодовані в кодї NRZI. Потік даних складається з пакетів, пакет в свою чергу складається з декількох полів: поле синхронізації (зразкова синхронізація), ідентифікатор пакету (PacketID, PID), поле адреси (ADDR), поле кінцевої точки (ENDP), дані і поле циклічного контролю (CRC). USB реалізує чотири типи передачі: передача керування, передача переривання, ізохронна передача і передача потоку. Кожен з цих типів передач специфічний для різних вимог пристрою.

У даній реалізації розглядається тип передачі керування. Даний режим, як правило, використовується для керування налаштуваннями пристрою, однак, може також використовуватися для передачі загального призначення. Режим передачі керування повинен бути присутнім у кожного USB-пристрої, тому що він використовується для конфігурації при підключенні пристрою, коли необхідно отримати інформацію про пристрій, встановлені адреси пристрою та ін. Кожна передача керування 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 пристрою за одну USB-передачу. В апаратно-реалізованих USB-приймачах різні частини передачі автоматично дешифруються і пристрій повідомляється, коли всі повідомлення призначено для окремого пристрою. При програмній реалізації USB-повідомлення дешифрується програмно після прийому в буфер всього повідомлення. Внаслідок цього виникають вимоги і 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. Дані можливості обмежуються швидкістю мікроконтролера і розміром пам'яті програм/даних, тому що програма повинна бути ретельно оптимізована. В деяких випадках обчислювальні можливості мікроконтролера дуже близькі до мінімальних вимог, тому, вся програма, як правило, реалізується на Асемблері.

1.3 Програмна реалізація послідовної шини USB

Згідно вимог USB2.0 апаратура, що працює з цим інтерфейсом, має підтримувати низькошвидкісний режим режиму USB-зв'язку на швидкості 1.5Мбіт/сек.

Програмна реалізація повинна працювати в складі AVR-мікроконтролерів з об'ємом пам'яті від 2 кбайт і вище та включати в себе процедури передачі та прийому інформації, процедуру переривання.

1.4 Постановка задачі розроблення

Для реалізації апаратної підтримки з використанням МК необхідно тільки кілька зовнішніх компонентів:

- один резистор для детекції низької швидкості USB;
- діодник/стабілізатор напруги з фільтрацією.

Необхідно реалізувати наступні функції:

- безпосереднє керування лінією вводу-виводу;
- перетворення USB–RS232;

Для реалізації таких функцій необхідно використати допоміжний регістр EEPROM та застосувати DLL-бібліотеку функцій з використанням мови програмування C.

2 РОЗРОБЛЕННЯ ДРАЙВЕРУ ПІДТРИМКИ ПОСЛІДОВНОЇ ШИНИ USB 3 ВИКОРИСТАННЯМ МІКРОКОНТРОЛЕРІВ AVR

2.1 Вибір елементної бази

Швидкий розвиток компонентів елементної бази для побудови мікроконтролерних (МК) систем керування різноманітними пристроями та технологічними процесами сприяє науково-технічному розвитку країни, є основою удосконалення архітектури таких систем, якісного підвищення їх продуктивності і надійності.

Номенклатура та область застосування таких систем постійно розширюється. На сучасному етапі науково-технічного розвитку їх впровадження охоплює практично всі види виробничої та наукової діяльності.

Застосування МК у науково-технічних рішеннях вимагає від спеціалістів досконалого володіння сучасними методами проектування МК систем, вміння використовувати їх при практичному вирішенні інженерних задач.

МК в повсякденному житті застосовуються як в складній побутовій техніці, так і у супутникових навігаційних системах. До сфери застосування МК входить керування пристроями різного призначення за допомогою дискретних сигналів і багато іншого. Можна сказати, що без МК в даний час не обходиться практично жодний сучасний електронний пристрій.

У 2002 році фірма Atmel почала випуск нових серій 8-розрядних МК на базі ядра AVR. МК AVR на сьогоднішній час діляться на 3 групи: Classic, Mega і Tiny.

Базовий комплект системи команд сімейства Classic, Mega та Tiny МК налічує до 120 команд різноманітного призначення.

Популяризації використання AVR-мікроконтролерів сприяло використання RISC-архітектури, яка характеризується потужним набором інструкцій, більшість яких виконуються за один машинний цикл. Це означає, що при рівній частоті тактового генератора вони забезпечують продуктивність в 12 (6) раз більше продуктивності попередніх мікроконтролерів на основі CISC-архітектури (наприклад, MCS51). З іншого боку, в рамках однієї програми із заданим швидкодією, AVR-мікроконтролер може тактіроваться в 12 (6) раз меншою тактовою частотою, забезпечуючи рівне швидкодію, але при цьому споживаючи набагато меншу потужність.

Для прикладу в таблиці 2.1 наведені технічні характеристики декількох мікроконтролерів ATtiny фірми Atmel.

Таблиця 2.1 – Характеристики мікроконтролерів ATtiny фірми Atmel

ATtiny2313	1.8-5.5	20	15	2K	128	128	SPI UART	-	1x8bit 1x16bit
ATtiny24	1,8...5,5	20	12	2K	128	128	USI 4xPWM RTC	8x10bit	1x8bit 1x16bit
ATtiny25	2,7...5,5	20	32	2K	128	128	SPI UART	4x10bit	1x8bit 1x8bit high speed
ATtiny25 Automotive	2,7...5,5	16	32	2K	128	128	SPI UART	4x10bit	1x8bit 1x8bit high speed
ATtiny25V	1.8 - 5.5	10	32	2K	128	128	SPI UART	4x10bit	1x8bit 1x8bit high speed
ATtiny26	2.7-5.5	16	16	1K	128	128	SPI UART	11x10bit	2x8bit
ATtiny261	1.8-5.5	20	16	2K	128	128	PWM USI	11x10bit	1x8bit 1x16bit
ATtiny461	1.8-5.5	20	16	4K	256	256	PWM USI	11x10bit	1x8bit 1x16bit
ATtiny28L	1.8-5.5	4	20	2K	-	-	-	-	1x8bit
ATtiny44	1,8...5,5	20	12	4K	256	256	USI 4xPWM RTC	8x10bit	1x8bit 1x16bit

На рисунку 2.1 наведена загальна структурна схема МК AVR.

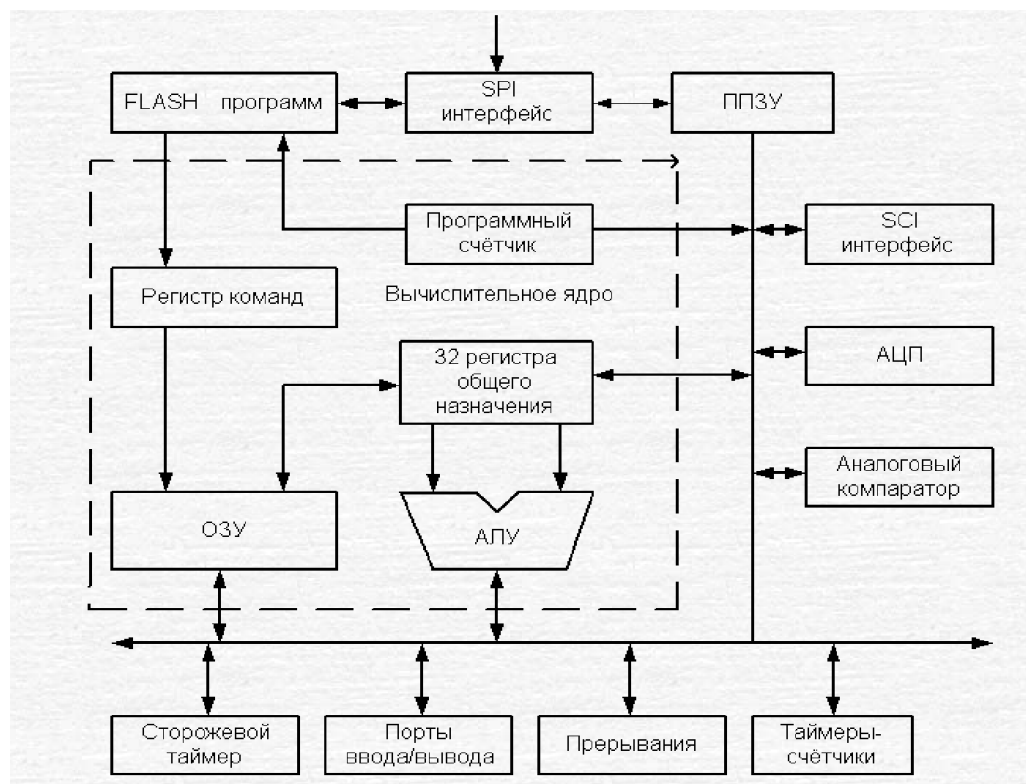


Рисунок 2.1 – Загальна структурна схема МК AVR

Таким чином, AVR-мікроконтролери представляють більш широкі можливості по оптимізації продуктивності/енергоспоживання, що особливо важливо при розробці додатків з батарейним харчуванням. Мікроконтролери забезпечує продуктивність до 16 млн. оп. в секунду і підтримують флеш-пам'ять програм різної ємності: 1...256 кбайт. AVR-архітектура оптимізована під язык високого рівня Сі, а більшість представників сімейства megaAVR містять 8-канальний 10-розрядний АЦП, а також сумісний з IEEE 1149.1 інтерфейс JTAG або debugWIRE для вбудованого налагодження. Крім того, всі мікроконтролери megaAVR з флеш-пам'яттю ємністю 16 кбайт і більше можуть програмуватися через інтерфейс JTAG.

На рисунках 2.2 і 2.3 показана схема підключення мікроконтролера до шини USB. Дані схеми мають специфічне призначення: перетворювач USB–RS232. На них також реалізовані специфічні функції, такі як безпосереднє керування виведенням і читання/запис EEPROM.

На рис. 2.2 наведена схема підключення шини USB до мікроконтроллера ATtiny2313 в якості перетворювача USB–RS232 з 32 байтним буфером FIFO, 8-розрядним керуванням вводом-виводом та 128 байтним EEPROM.

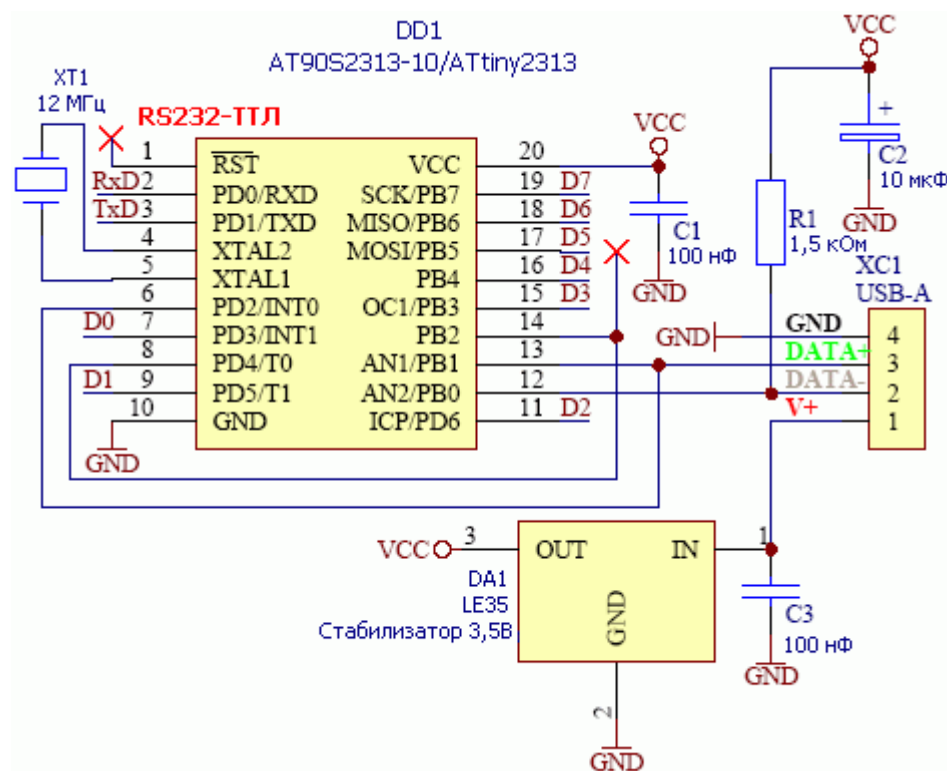


Рисунок 2.2 – Підключення шини USB до мікроконтроллера ATtiny2313

На рис. 2.3 наведена схема підключення шини USB з мікроконтролером ATmega8/48/88/168 в якості перетворювача USB–RS232 з 800 байтним буфером FIFO, керуванням вводом-виводом та 512 байтним EEPROM.

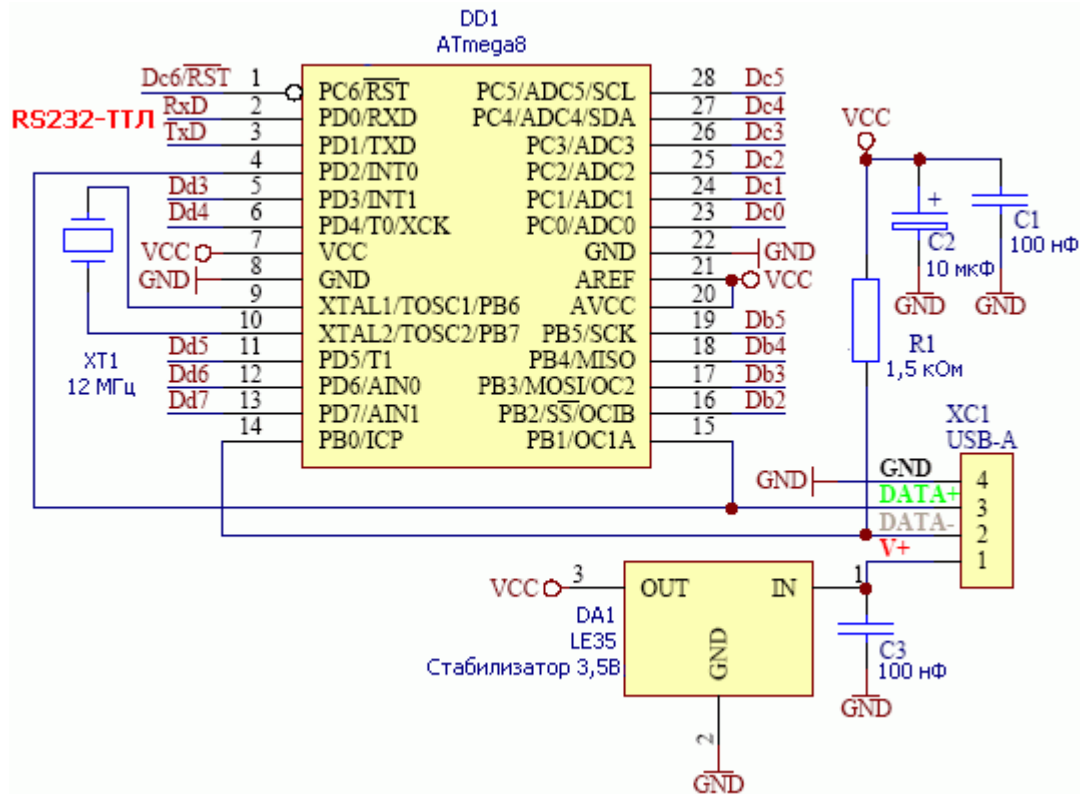


Рисунок 2.3 – Підключення шини USB з мікроконтролером ATmega8/48/88/168

Лінії даних USB, DATA– і DATA+, підключені до виводів PB0 і PB1 мікроконтролера AVR. Це з'єднання не можна змінити, тому що в програмі використовуються унікальності ядра AVR для швидкого прийому сигналу: бітовий сигнал захоплюється з ліній даних і молодший розряд (PB0) зсувається вправо в перенос, а потім в приймальний регістр, який накопичує біти, прийнятих з ліній даних. PB1 використовується в якості вхідного сигналу, оскільки у 8-вивідного AT90S2323 даний контакт може використовуватися в якості входу зовнішнього переривання без будь-яких зовнішніх підключень до INT0. В інших мікроконтролерах AVR необхідно зовнішнє підключення DATA+ виводу INT0, що гарантує незмінність програми при переході між мікроконтролерами.

Для відповідного підключення USB-пристрою і сигналізації мікроконтролер AVR, що працює як низькошвидкісний USB-пристрій, повинен мати підтягуючий резистор 1,5 кОм на лінії DATA–.

Напруга V_{cc} , що надходить від USB-порту комп'ютера, може змінюватись в діапазоні 4.4...5.25В. Перед його подачею до мікроконтролера і підтягуючого резистору опором 1,5 кОм дана напруга стабілізується на рівні 3.0...3.6В. Тип стабілізатора напруги залежить від рівня навантаження цільової системи. Стабілізатор напруги повинен відноситись до типу стабілізаторів з малим мінімальним перепадом напруги.

У схемах на рисунках 2.1 і 2.2 використовується стабілізатор LE35 з номінальною вихідною напругою 3,5 В. Однак можна використовувати будь-яке інше рішення по стабілізації напруги, якщо воно відповідає вимогам проектованої системи. В деяких випадках може бути цілком прийнятним параметричний стабілізатор на основі стабілітрона.

Інші компоненти необхідні для відповідної роботи мікроконтролера: кварцовий резонатор для синхронізації мікроконтролера і конденсатори для фільтрації живлення.

Таким чином, для отримання завершеного пристрою, який забезпечує зв'язок з комп'ютером через інтерфейс USB, необхідно невелику кількість елементів. Для розширення функціональних можливостей можуть бути додані додаткові компоненти.

Інфрачервоний датчик TSOP1738 може використовуватися для прийому інфрачервоного сигналу. Перетворювач рівня ТТЛ–RS232 MAX232 необхідно додати при необхідності розробки перетворювача інтерфейсів USB-RS232. Для керування світлодіодами або дисплеєм їх необхідно підключити до ліній вводу-виводу безпосередньо або через резистори.

2.2 Вибір програмного середовища налаштування та програмна реалізація

Для програмування МК сімейства AVR існує багато засобів розробки, проте, найбільш популярним, поза сумнівом, є програмний пакет AVR Studio 4. Є ряд причин такої популярності – це безкоштовний програмний пакет, розроблений фірмою ATMEL, який об'єднує в собі текстовий редактор та програмний емулятор МК різних типів. Пакет AVR Studio 4 використовується також спільно з апаратними засобами налаштування фірми ATMEL.

Вибір мови програмування залежить від вимог, що пред'являються до самої програми. Якщо потрібна максимальна швидкодія, компактність коду і його надійність, та "ручне" гнучке керування різними елементами МК, то це, звичайно, Асемблер. Якщо ж потрібна простота та комфортність в написанні, а також наочність програмного коду, то це C++, проте в цьому випадку буде деякий програш в продуктивності. Синтаксис Асемблера дуже простий, всі його команди базуються тільки на архітектурі самого МК,

тобто, при програмуванні все зводиться до системи команд.

Протокол USB-прийому та дешифрування повністю реалізований програмно. Програма спочатку приймає бітовий потік USB в один USB-пакет у внутрішньому буфері. Початок прийому ініціюється за зовнішнім перериванням INT0, яке відповідає за зразкову синхронізацію. У процесі прийому перевіряється тільки сигнал кінця пакета (визначається тільки EOP). Це необхідно з огляду на дуже високу швидкість передачі даних по шині USB. Після успішного прийому програма дешифрує пакети даних і аналізує їх.

По-перше, за допомогою адреси аналізується, що прийнятий пакет належить 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 заданою адресою, пристрій повинен відповісти комп'ютеру підтвердження пакетом ACK. Таким чином, дешифрування адреси є критичною частиною USB-відповіді.

Після прийому бітового потоку отримуємо послідовність, закодовану в коді NRZI, шляхом порозрядного заповнення вхідного буфера. У процесі дешифрування ми спочатку видаляємо порозрядне заповнення, а потім NRZI-кодування. Всі ці зміни виконуються у другому буфері (копія приймального буфера). Після дешифрування поточного пакета може прийматися новий пакет. Для даної точки, швидкість дешифрування не так важлива, тому що пристрій може затримати свою відповідь. 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 знаходиться в процесі виконання. У цьому випадку комп'ютер відправить запит знову.

Мікроконтролер також дешифрує основну USB-транзакцію і виконує запитувану дію; наприклад, відправлення символу по лінії RS232 і очікування завершення транзакції, а також підготовка відповідної відповіді. При виконанні даного процесу пристрій буде перериватися деякими пакетами, що надходять від комп'ютера, зазвичай пакети IN для отримання відповіді від пристрою.

На дані пакети IN пристрій повинен відповісти пакетами NAK. Якщо відповідь готова і пристрій виконав необхідну дію, відповідь має спочатку пройти через функцію обчислення і приєднання CRC, потім виконується NRZI-кодування, а потім порозрядне заповнення. Тепер, коли комп'ютер запитує відповідь, даний бітовий потік передається по лініях даних відповідно до вимог стандарту USB.

2.3 Алгоритм реалізації процедури прийому

На рис. 2.4 наведий алгоритм реалізації процедури прийому.

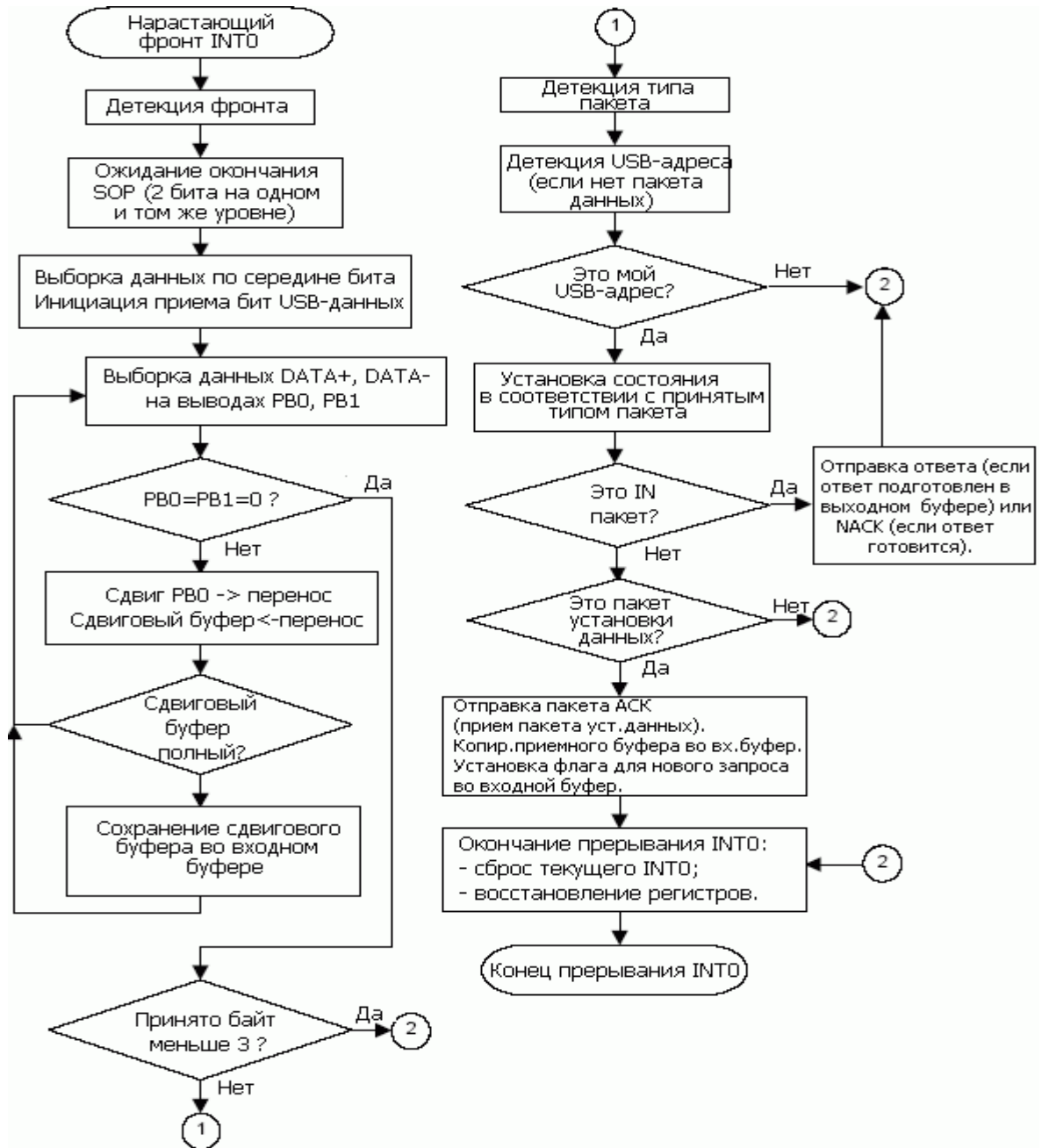


Рисунок 2.4 — Алгоритм процедуры прийому

Програма розділена на блоки: процедури переривання, процедури дешифрування, USB-прийом, USB-передача, дешифрування запитуваної дії та виконання необхідних дій.

Користувач, при необхідності, може додати власні функції. У коді програми можна знайти приклади, як зробити специфічні для користувача функції. Таким чином,

користувач на основі існуючих вбудованих функцій може написати власні розширення. Наприклад, на основі вбудованої функції безпосереднього керування виводу можна додати підтримку TWI.

2.4 Процедура оброблення переривання

Зовнішнє переривання 0 перебуває в активному стані протягом усього часу виконання програми. Дана процедура ініціює прийом даних, послідовно передаються по шині USB. Зовнішнє переривання виникає при появі наростаючого фронту на контакті INT0. Наростаючий фронт вказує на початок зразкової синхронізації (див. рис. 1.4) і ініціює виконання процедури USB-прийому.

Спочатку процес оцифровки даних повинен бути засінхронізовано до середини біта. Це виконується відповідно до зразкової синхронізації. Оскільки тривалість передачі одного біта дорівнює 8 періодам синхронізації XTAL і виникнення переривання може бути затримано (+/- 4 періоди), то синхронізацію до фронту зразкової синхронізації необхідно виконати ретельно. Закінчення передачі зразковою синхронізацією і початок передачі біт даних визначається наступним за зразковою синхронізацією двома бітами з низьким рівнем.

Після цього починається фактична оцифровка даних. Оцифровка виконується по середині біта. При швидкості передачі даних 1,5 Мбіт/с (1.5МГц) і частоті синхронізації мікроконтролера 12 МГц в розпорядженні є тільки 8 тактів для оцифровки даних, запису результату в однобайтний буфер, зсуву даних в однобайтном буфері, перевірки на прийом всього байта, запису байта в статичну ОП та визначення закінчення пакета (EOP). Це найбільш критична частина програми; все повинно бути зроблено синхронно і за встановлені часові рамки. Після прийому всього USB-пакета необхідно виконати дешифрування пакета. Спочатку необхідно швидко визначити тип пакету (SETUP, IN, OUT, DATA) і прийняти USB-адресу. Швидке дешифрування повинно виконуватися всередині процедури обробки переривання, тому що необхідно дуже швидко відповісти після прийому USB-пакета (пристрій повинен відповісти пакетом ACK, якщо прийнятий пакет з адресою пристрою і NAK, якщо прийнятий пакет, адресований пристрою, але відповідь ще не готовий).

В кінці процедури прийому (після відправки пакета підтвердження ACK/NAK) записані в буфер дані повинні бути скопійовані в інший буфер, в якому виконується дешифрування даних. Це необхідно для звільнення приймального буфера для прийому нового пакета.

У процесі прийому дешифрується тип пакета і встановлюється відповідне значення прапора. Даний прапор перевіряється в основному циклі програми і залежно від його значення виконується відповідна дія і готується відповідний відповідь без пред'явлення будь-яких вимог по швидкодії мікроконтролера.

Для збереження високої швидкості обробки зовнішнього переривання INTO воно повинно залишатися активним постійно, навіть при обробці інших переривань (наприклад, переривання по послідовному прийому). Швидкість прийому в процедурі обробки переривання INTO дуже важлива, тому програму необхідно оптимізувати за швидкодією і часом виконання. Важливою проблемою є оптимізація резервування регістра в процедурах обробки переривань.

2.5 Основний програмний цикл

В основному програмному циклі необхідно тільки перевіряти стан прапора дії і визначати, що робити після отримання даних. Крім того, в основному циклі перевіряється, скинутий чи USB-інтерфейс (обидві лінії даних знаходяться в низькому рівні протягом тривалого часу) і, в 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 запитуваної відповіді в буфері передавача (з порозрядним заповненням і NRZI-кодуванням). Після цього активізується прапор для сигналізації про готовність відповіді до відправки.

Фізична передача вихідного буфера по лініях USB виконується в процедурі прийому як відповідь на пакет IN.

2.6 Опис підпрограм

Наведемо опис підпрограм, що реалізовані у даній роботі.

Підпрограма Reset – виконує ініціалізацію ресурсів мікроконтролера AVR: стек, лінії вводу-виводу, буфери USB, переривання.

Підпрограма Main – основний цикл програми. Перевіряє стан прапора дії; якщо прапорець встановлений, то виконує необхідну дію. Додатково, дана процедура перевіряє скидання ліній даних USB-шини і в разі визначення скидання проводить повторну ініціалізацію USB-інтерфейсу.

Підпрограма `Int0Handler` – процедура обробки зовнішнього переривання `INT0`. Основний процесор прийому/передачі. Запис даних в буфер, визначення одержувача USB-пакета (USB-адреси), розпізнавання пакета, відправка відповіді комп'ютера.

Підпрограма `SetMyNewUSBAddresses` – процедура зміни USB-адреси. Адреса змінюється і кодується в NRZI еквівалент. Це необхідно для того, щоб швидко виконати дешифрування адреси в процесі прийому USB-пакета.

Підпрограма `FinishReceiving` – копіює неперевірені дані з прийнятого USB-пакета для його дешифрування (дешифрування NRZI і порозрядного заповнення).

Підпрограма `SendPreparedUSBAnswer` – відправляє підготовлений вміст вихідного буфера по лініях USB. NRZI-кодування і порозрядне заповнення виконується в процесі передачі. Пакет завершується сигналом EOP.

`ComposeZeroDATA1PIDAnswer` – складає нульову відповідь для передачі. Нульова відповідь не містить даних і використовується в деяких випадках в якості відповіді, коли в пристрої немає доступних додаткових даних.

`InitACKBuffer` – ініціалізує буфер в оперативній пам'яті (ОП) даними ACK (пакет підтвердження ACK). Даний буфер часто використовується в якості відповіді, що він завжди залишається готовим у пам'яті.

`SendACK` – передає пакет ACK по лініях USB.

`InitNAKBuffer` – ініціалізує буфер в ОП даними NAK (пакет немає підтвердження NAK). Даний буфер часто використовується як відповідь про готовність буфера в пам'яті.

`BitStuff` – видаляє/додає порозрядне заповнення в прийнятих USB-даних. Порозрядне заповнення додається в комп'ютері на апаратному рівні у відповідності до вимог USB, щоб гарантувати синхронізацію оцифровки даних. Дана процедура генерує прийняті дані без порозрядного заповнення або формує дані з порозрядним заповненням для передачі.

`CheckCRCIn` – виконує перевірку CRC (контроль циклічної надмірності) в пакеті прийнятих даних. CRC додається до USB-пакету для детекції пошкодження даних.

`LoadDescriptorFromROM` – завантаження даних з постійної пам'яті (ПП) у вихідний буфер USB.

`LoadDescriptorFromROMZeroInsert` – завантаження даних з ПП у вихідний буфер USB, але кожен парний байт записується з нульовим значенням. Використовується, коли дескриптор рядка запитується у форматі UNICODE (збереження ПП).

`LoadDescriptorFromSRAM` – завантаження даних з ОЗУ в вихідний буфер USB.

`LoadDescriptorFromEEPROM` – завантаження даних з EEPROM вихідний буфер USB.

`Load [X] Descriptor` - виконує вибір джерела відповіді: ПП, ОП або EEPROM.

PrepareUSBOutAnswer – готує USB-відповідь у вихідному буфері відповідно до запиту від комп'ютера і виконує запитувану дію. Додає порозрядне заповнення у відповідь.

Стандартні запити необхідні відповідно до технічних вимог USB.

Специфічними запитами виробника є запити, які можуть отримати специфічні дані виробника (передача керування IN USB). Передача керування IN USB використовується для зв'язку AVR з комп'ютером.

Розробники можуть додати свої власні функції і, таким чином, розширити гнучкість пристрою. Різні документовані вбудовані функції в вихідному коді можуть використовуватися як шаблон для розробки власних функцій.

Головний комп'ютер шини USB також пов'язують із пристроєм за допомогою передач керування IN. Головний відправляє пристрою 8-байтний IN-пакет даних в форматі визначеному вище (номер функції і параметри), а потім пристрій відповідає запитаними даними. Розмір запитуваних даних обмежений у програмі в деяких випадках до 255 байт, але основне обмеження виконується драйвером пристрою на стороні керуючого комп'ютера.

2.7 Програма для ПК

Для забезпечення зв'язку з налаштуванням необхідна програмна підтримка на стороні ПК. Програма розділена на три рівні:

Драйвер пристрою: використовується для зв'язку на низькому рівні з пристроєм і для інсталяції в операційну систему WindowsXP.

DLL-бібліотека використовується для інкапсуляції функцій пристрою і зв'язку з драйвером пристрою. DLL спрощує доступ до функцій пристрою з програми користувача. До складу бібліотеки входять деякі функції пристрою і операційної системи (завдання, буфери та ін.)

Додаток користувача створює користувацький інтерфейс для зручного зв'язку між користувачем і пристроєм. Викликаються функції тільки з DLL-бібліотеки.

Спочатку, USB-пристрій підключається до комп'ютерного USB-порту, потім операційна система визначить пристрій і запросить файли драйвера. Даний процес називається інсталяцією пристрою. Для виконання інсталяції необхідно не тільки створити драйвер пристрою, але також інсталяційний скрипт, в якому описується послідовність інсталяції.

Драйвер працює під всіма версіями 32-розрядних операційних систем Windows. Інсталяційний скрипт, записаний в INF-файл, використовується в процесі інсталяції пристрою. В даному INF-файлі описані різні інсталяційні кроки. Файл "AVR309.inf" створений з допомогою текстового редактора. Він запитується операційною системою в процесі інсталяції.

Після ініціації інсталяції файл драйвера копіюється в операційну систему, а потім виконуються необхідні системні зміни. INF-файл гарантує інсталяцію DLL-бібліотеки в системну папку, що гарантує простоту її виклику з різних додатків. Для інсталяції пристрою необхідно три файли: INF-файл "AVR309.inf", драйвер "AVR309.sys" і DLL-бібліотека "AVR309.dll".

Нижче наведено INF-файл "AVR309.inf".

```
[Version]
Signature="$CHICAGO$"
Class=USB
provider=%MB%
LayoutFile=layout.inf
[Manufacturer]
%MfgName%=IgorCeskoCompany
[IgorCeskoCompany]
%USB\VID_03EB&PID_21FF.DeviceDesc%=AVR309Prj.Dev,
USB\VID_03EB&PID_21FF
%USB\VID_03EB&PID_21FE.DeviceDesc%=AVR309Prj.Dev,
USB\VID_03EB&PID_21FE
[PreCopySection]
HKR,,NoSetupUI,1
[DestinationDirs]
AVR309Prj.Files.Ext = 10,System32\Drivers
AVR309Prj.Files.Inf = 10,INF
AVR309Prj.Files.Dll = 25

[AVR309Prj.Dev]
CopyFiles=AVR309Prj.Files.Ext, AVR309Prj.Files.Inf, AVR309Prj.Files.Dll
AddReg=AVR309Prj.AddReg
[AVR309Prj.Dev.NT]
```

```

CopyFiles=AVR309Prj.Files.Ext, AVR309Prj.Files.Inf, AVR309Prj.Files.Dll
AddReg=AVR309Prj.AddReg
[AVR309Prj.Dev.NT.Services]
Addservice = AVR309Prj, 0x00000002, AVR309Prj.AddService
[AVR309Prj.AddService]
DisplayName = %AVR309Prj.SvcDesc%
ServiceType = 1 ; SERVICE_KERNEL_DRIVER
StartType = 3 ; SERVICE_DEMAND_START
ErrorControl = 1 ; SERVICE_ERROR_NORMAL
ServiceBinary = %10%\System32\Drivers\AVR309.sys
LoadOrderGroup = Base
[AVR309Prj.AddReg]
HKR,,DevLoader,,*ntkern
HKR,,NTMPDriver,,AVR309.sys
HKLM,"System\Currentcontrolset\Services\AVR309Prj\Parameters","MaximumTransferS
ize",0x10001,256
HKLM,"System\Currentcontrolset\Services\AVR309Prj\Parameters","DebugLevel",0x100
01,2
[AVR309Prj.Files.Ext]
AVR309.sys
[AVR309Prj.Files.Inf]
AVR309.inf
[AVR309Prj.Files.Dll]
AVR309.dll

```

Нижче наведено файл "AVR309.sys"

File Version Information :

Version language : Английский (США)

Comments : Windows driver for AVR309:USB to UART protocol
converter. Application note AVR309.

CompanyName : Kravchenko

FileDescription : WinXP driver for AVR309-USB device

FileVersion : 1.0.0.0

InternalName : AVR309USB_0

LegalCopyright	: free application note
LegalTrademarks	: Atmel corporation
OriginalFilename	: AVR309.sys
ProductName	: AVR interface AVR309-USB driver
ProductVersion	: 1.0.0.0
Creation Date	: 18/05/2012 17:45:14
Last Modif. Date	: 14/02/2006 16:56:10
Last Access Date	: 18/05/2012 18:40:08
FileSize	: 8652 bytes (8.449 KB, 0.008 MB)
FileVersionInfoSize	: 2252 bytes
File type	: Application (0x1)
Target OS	: Win32 API (Windows NT) (0x40004)
File/Product version	: 2.1.0.0 / 2.1.0.0
Language	: Английский (США) (0x409)
Character Set	: 1200 (ANSI - Unicode (BMP of ISO 10646)) (0x4B0)
Build Information :	
Debug Version	: no
Patched Version	: no
Prerelease Version	: no
Private Version	: no
Special Build	: no

2.8 Використання бібліотеки AVR309.dll фірми Atmel

DLL-бібліотека пов'язується з драйвером пристрою і всіма функціями пристрою, реалізованих в цій бібліотеці. Спосіб написання програми кінцевого користувача гранично спрощений. DLL-бібліотека гарантує привілейований доступ до пристрою (впорядковує доступ до пристрою), містить системний буфер для прийому даних через RS232 і створює одну системну задачу для читання буфера даних RS232.

Упорядкування в DLL гарантує, що тільки один додаток/завдання буде пов'язано з пристроєм в даний час. Це необхідно через можливість накладення запитів/відповідей від різних додатків в одне і теж час.

Системний буфер для прийому даних через RS232 гарантує, що прийняті дані будуть розміщені в одному буфері, який є загальним для всіх програм. В такому разі прийняті

дані пристроєм будуть відправлятися всіх програм. Не варто турбуватися про те, що додаток прийме не повністю дані через те, що інший додаток до цього прочитував буфер даних.

Для всіх програм існує тільки одна системна задача, і дані будуть періодично запитуватися у пристрою через RS232. В цьому випадку завдання буде зберігати прийняті дані в системний буфер. Наявність тільки одного системного буфера гарантує невелику завантаження ЦПУ (в порівнянні, коли кожен додаток має свою власну задачу) і спрощує збереження даних в системний буфер. Всі функції пристрою визначено в бібліотеці DLL, а їх експорт виконується в зручній формі: не у вигляді номера функції і параметрів, а у вигляді зручного імені функції з параметрами. Деякі функції більш складні внутрішньо, наприклад, функція читання буфера даних RS232. Таким чином, розробники можуть швидко розробляти свої програми, використовуючи тільки інтерфейс DLL. Немає необхідності вивчати функції низького рівня пристрої, тому що DLL-бібліотека розділяє рівень прикладного програмування від апаратного рівня.

2.9 Похибка генерації швидкості універсального асинхронного передавача/приймача

Мікроконтролер використовує тактову частоту 12МГц для здійснення перетворення USB. Недоліком використання такої частоти синхронізації є наявність невеликої похибки при генерації стандартних швидкостей зв'язку. Проте за рахунок високого значення частоти синхронізації похибка мінімальна. Максимально допустима похибка генерації швидкості зв'язку повинна бути 4%, тому що максимальна похибка дорівнює відношенню тривалості половини біта (0.5) до максимальної тривалості пакета 12 біт = 1 старт-біт + 8 біт даних + 1 біт паритету + 2 стоп-біта.

Таким чином, похибка дорівнює $0.5/12 * 100\% = 4.1\%$.

Функції в DLL автоматично обчислюють похибку і встановлюють таку швидкість зв'язку, відхилення від стандартної якої не перевищує 4%. У разі визначення швидкості, яка не підтримується, виводиться повідомлення про помилку. Однак не рекомендується використовувати швидкості з похибкою більше 2%.

У таблиці 2.2 підсумовуються похибки в генерації стандартних швидкостей при роботі мікроконтролера на частоті 12 МГц.

Таблиця 2.2 – Похибки генерації швидкостей UART і мікроконтролера AVR

Стандартная скорость	Скорость в AVR	Погрешность, %
600	602	+0.33
1200	1204	+0.33
2400	2408	+0.33
4800	4808	+0.17
9600	9616	+0.17
19200	19230	+0.16
28800	28846	+0.16
38400	38462	+0.16
57600	57692	+0.16
115200	115384	+0.16

2.10 Вихідний код програми для мікроконтролера AVR

Вихідний код програми USBtoRS232.asm для мікроконтролера ATmega8 реалізовано в AVR Studio 4. Програма займає в пам'яті МК AVR 68 кБ.

Фрагмент ініціалізації МК.

```

;*****
;* Init program
;*****
;-----
reset: ;initialization of processor and variables to right values
    ldi temp0,StackBegin ;initialization of stack
    out SPL,temp0
    clr XH ;RS232 pointer
    clr YH ;USB pointer
    clr ZH ;ROM pointer
    sts RS232BufferBegin+0,YH ;clear lengths of RS232 code in buffer
    ldi temp0,RS232BufferBegin+4
    sts RS232BufferBegin+1,temp0;znuluj ukazovatel citania
    sts RS232BufferBegin+2,temp0;znuluj ukazovatel zapisu
    clr RS232BufferFull
rcall InitACKBuffer ;initialization of ACK buffer
call InitNAKBuffer ;initialization of NAK buffer
rcall USBReset ;initialization of USB addresses
    sbi TSOPpullupPort,TSOPpin ;set pull-up on TSOP input
    ldi temp0,(1<<LEDlsb0)+(1<<LEDlsb1)+(1<<LEDlsb2)
    out LEDPortLSB,temp0 ;set pull-up on all LED LSB
    out LEDPortMSB,temp0 ;set pull-up on all LED MSB
    sbi PORTD,0 ;set pull-up on RxD input

```

```

ldi temp0,InitBaudRate ;set UART speed
out UBRRL,temp0
sbi UCSRB,TXEN ;enable transmitting of UART
sbi UCSRB,RXEN ;enable receiving of UART
sbi UCSRB,RXCIE ;enable interrupt from receiving of UART
ldi temp0,0x0F ;INT0 - respond to leading edge
out MCUCR,temp0 ;
ldi temp0,1<<INT0 ;enable external interrupt INT0
out GIMSK,temp0

```

Бібліотека AVR309.dll використана з сайту фірми Atmel і реалізована на мові C.

Приклад використання бібліотеки DLL у вигляді додатку користувача представлена у вигляді файлу AVR309USBdemo.exe, що виконується (рис. 2.5).

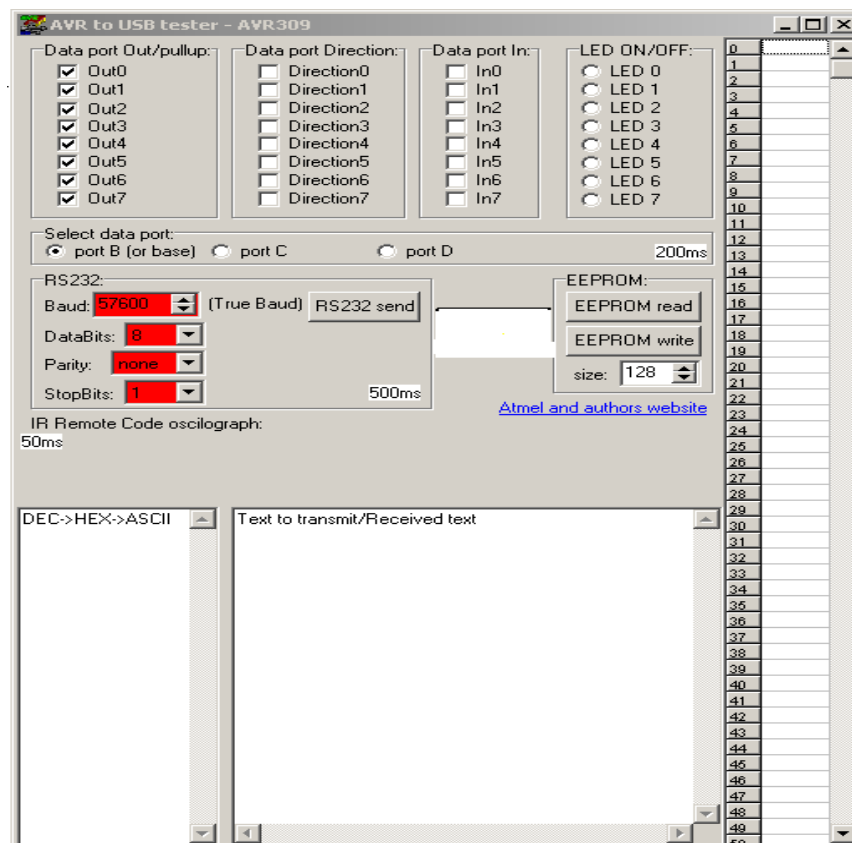


Рисунок 2.5 – Додаток користувача з використанням бібліотеки AVR309.dll

Текст програми підтримки послідовної шини USB на базі мікроконтролерів AVR наведено у додатку А.

3 ОХОРОНА ПРАЦІ ТА БЕЗПЕКА В НАДЗВИЧАЙНИХ СИТУАЦІЯХ

3.1 Загальні питання з охорони праці

Умови праці на робочому місці, безпека технологічних процесів, машин, механізмів, устаткування та інших засобів виробництва, стан засобів колективного та індивідуального захисту, що використовуються працівником, а також санітарно-побутові умови повинні відповідати вимогам нормативних актів про охорону праці. В законі України «Про охорону праці» визначається, що охорона праці - це система правових, соціально-економічних, організаційно-технічних, санітарно-гігієнічних і лікувально-профілактичних заходів та засобів, спрямованих на збереження життя, здоров'я і працездатності людини у процесі трудової діяльності.

Завданням даної дипломної роботи було розроблення драйверу підтримки послідовної шини USB з застосування мікроконтролера Atmel для забезпечення зв'язку пристроїв на базі інтерфейсу USB в комп'ютерних системах, і як результат створено драйвер підтримки програми, що забезпечує необхідний зв'язок та швидкість обміну даними.

При роботі з обчислювальною технікою змінюються фізичні і хімічні фактори навколишнього середовища: виникає статична електрика, електромагнітне випромінювання, змінюється температура і вологість, рівень вміст кисню і озону в повітрі. Забезпечення цих умов покладається на власника або уповноважений ним орган (далі роботодавець). Умови праці на робочому місці, безпека технологічних процесів, машин, механізмів, устаткування та інших засобів виробництва, стан засобів колективного та індивідуального захисту, що використовуються працівником, а також санітарно-побутові умови повинні відповідати вимогам нормативних актів про охорону праці.

3.2 Правові та організаційні основи охорони праці

Державна політика в галузі охорони праці визначається відповідно до Конституції України Верховною Радою України і спрямована на створення належних, безпечних і здорових умов праці, запобігання нещасним випадкам та професійним захворюванням. Відповідно до статті 3 Закону України «Про охорону праці» (далі – Закону) законодавство про охорону праці складається з Закону, Кодексу законів про працю України, Закону України "Про загальнообов'язкове державне соціальне страхування від нещасного випадку на виробництві та професійного захворювання, які спричинили втрату

працездатності" та прийнятих відповідно до них нормативно-правових актів, норм міжнародного договору (ратифіковані Конвенції і Рекомендації МОТ, директиви Європейської Ради).

Обов'язки працівників щодо додержання вимог нормативно-правових актів з охорони праці (ст. 14), відповідальність робітників всіх категорій за порушення вимог щодо охорони праці (ст. 44) та структура організації/виробництва системи управління охорони праці визначені безпосередньо «Інструкцією на робоче місце № 1», та іншими затвердженими власними нормативними актами з питань охорони праці (правилами, нормами, регламентами, положеннями, стандартами, інструкціями та іншими документами, обов'язковими до виконання), тобто тих, що діють на підприємстві/організації, і визначені в [11].

3.3 Організаційно-технічні заходи з безпеки праці

В організації/підприємстві проводиться навчання і перевірка знань з питань охорони праці відповідно до вимог Типового положення про порядок проведення навчання і перевірки знань з питань охорони праці, затвердженого наказом Держнаглядохоронпраці України від 26.01.2005 N 15, зареєстрованого в Міністерстві юстиції України 15.02.2005 за N 231/10511 [12].

3.4 Аналіз стану умов праці та вимоги до приміщення

Робота над створенням дипломного проекту проходитиме в приміщенні відповідної установи (компанії, підприємстві тощо). Для даної роботи достатньо однієї людини, для якої надано робоче місце зі стаціонарним комп'ютером. Геометричні розміри приміщення зазначені в таблиці 3.1.

Таблиця 3.1 – Розміри приміщення

Найменування	Значення
Довжина, м	5
Ширина, м	5
Висота, м	3
Площа, м ²	25
Об'єм, м ³	75

Згідно з [13] розмір площі для одного робочого місця оператора персонального комп'ютера має бути не менше 6 кв. м, а об'єм — не менше 20 куб. м. Отже, дане

приміщення цілком відповідає зазначеним нормам. Для зручності спільної роботи з іншими працівниками (обговорення ідей, з'ясування проблем і т.д.) в кімнаті є дивани і журнальний стіл, обставлені живими квітами. Також робочий процес пов'язаний з багатьма документами, теками, журналами для чого приміщення облаштоване принтером і шафою для зручності. Задля дотримання визначеного рівня мікроклімату в будівлі встановлено систему опалення та кондиціонування. Для забезпечення потрібного рівного освітленості кімната має вікно та систему загального рівномірного освітлення, що встановлена на стелі. Для дотримання вимог пожежної безпеки встановлено порошковий вогнегасник та систему автоматичної пожежної сигналізації.

3.5 Вимоги до організації робочого місця

При порівнянні відповідності характеристик робочого місця нормативним основні вимоги до організації робочого місця за [14] і відповідними фактичними значеннями для робочого місця, констатуємо повну відповідність в таблиці 3.2.

Таблиця 3.2 - Характеристики робочого місця

Найменування параметра	Фактичне значення	Нормативне значення
Висота робочої поверхні, мм	750	680 ÷ 800
Висота простору для ніг, мм	730	не менше 600
Ширина простору для ніг, мм	660	не менше 500
Глибина простору для ніг, мм	700	не менше 650
Висота поверхні сидіння, мм	470	400 ÷ 500
Ширина сидіння, мм	400	не менше 400
Глибина сидіння, мм	400	не менше 400
Висота поверхні спинки, мм	600	не менше 300
Ширина опорної поверхні спинки, мм	500	не менше 380
Радіус кривини спинки в горизонтальній площині, мм	400	400
Відстань від очей до екрану дисплея, мм	800	700 ÷ □ 800

Робочий стіл на досліджуваному місці також містить достатньо простору для ніг. Крісло, що використовується в якості робочого сидіння, є підйомно-поворотним, має підлокітники і можливість регулювання за висотою і кутом нахилу спинки, також воно м'яке і виконане з екологічної шкіри, що дає можливість працювати у комфорті. Екран монітору знаходиться на відстані 0.8 м, клавіатура має можливість регулювання кута нахилу 5-15°. Отже, за всіма параметрами робоче місце відповідає нормативним вимогам.

Приміщення кабінету знаходиться на другому поверсі трьох поверхової будівлі і має об'єм 78 м³, площу — 18 м². У цьому кабінеті обладнано три місця праці, з яких два укомплектовані ПК.

Температура в приміщенні протягом року коливається у межах 18–24°C, відносна вологість — близько 50%. Швидкість руху повітря не перевищує 0,2 м/с. Шум на робочому місці знаходиться на рівні 50 дБА. Система вентилявання приміщення — природна неорганізована, а опалення — централізоване.

Розміщення вікон забезпечує природне освітлення з коефіцієнтом природного освітлення не менше 1,5%, а загальне штучне освітлення, яке здійснюється за допомогою восьми люмінесцентних ламп, забезпечує рівень освітленості не менше 200 Лк.

У кабінеті є електрична мережа з напругою 220 В, яка створює небезпеку ураження електричним струмом. ПК та периферійні пристрої можуть бути джерелами електромагнітних випромінювань, аерозолів та шкідливих речовин (часток тонеру, оксидів нітрогену та озону).

За ступенем пожежної безпеки приміщення належить до категорії В. Кабінет має бути оснащений переносним вуглекислотним вогнегасником ВВК-5.

Наявна аптечка для надання долікарської допомоги, а також у кабінеті роблять вологе прибирання та щоденно провітрюють приміщення.

3.6 Навантаження та напруженість процесу праці

Під час виконання робіт використовують ПК та периферійні пристрої, що призводить до навантаження на окремі системи організму. Такі перекося у напруженні різних систем організму, що трапляються під час роботи з ПК, зокрема, значна напруженість зорового аналізатора і довготривале малорухоме положення перед екраном, не тільки не зменшують загального напруження, а навпаки, призводять до його посилення і появи стресових реакцій.

Найбільшому ризику виникнення різноманітних порушень піддаються: органи зору, м'язово скелетна система, нервово-психічна діяльність, репродуктивна функція у жінок.

Тобто наявне психофізіологічні небезпечні та шкідливі фактори:

а) фізичного перевантаження:

- статичного;
- динамічного;

б) нервово-психічного перевантаження:

- розумового перенапруження;

- монотонності праці;
- перенапруження аналізаторів;
- емоційних перевантажень.

Роботу за дипломним проектом визнано, таку, що займає 50% часу робочого дня та за восьмигодинної робочої зміни рекомендовано встановити додаткові регламентовані перерви:

- для розробників програм тривалістю 15 хв через кожен годину роботи.

3.7 Аналіз небезпечних та шкідливих факторів при роботі на персональному комп'ютері

Роботу, пов'язану з електронно-обчислювальними машинами (далі - ПК) з відео дисплейними терміналами (далі - ВДТ), у тому числі на тих, які мають робочі місця, обладнані ЕОМ з ВДТ і периферійними пристроями (далі - ПП), виконують із забезпеченням виконання [17], які встановлюють вимоги безпеки до обладнання робочих місць, до роботи із застосуванням ПК з ВДТ і ПП. Переважно роботи за проектами виконують у кабінетах чи інших приміщеннях, де використовують різноманітне електрообладнання, зокрема ПК та периферійні пристрої.

Робочі місця мають відповідати вимогам цих Правил та Державних санітарних правил і норм роботи з візуальними дисплейними терміналами електронно-обчислювальних машин, затверджених постановою Головного державного санітарного лікаря України від 10.12.98 N 7 [14].

Це передбачає, що визначена виробнича діяльність пов'язана з наявністю певної кількості небезпечних та/або шкідливих виробничих факторів. Тому у першій частині цього підрозділу за результатами аналізу повинні бути визначені такі фактори.

Робота ПК та периферійних пристроїв супроводжує виділення багатьох хімічних речовин, зокрема озону, оксидів нітрогену та аерозолів (високодисперсних частинок тонера). Для прикладу, за умов роботи з ПК виникають наступні небезпечні та шкідливі чинники: несприятливі мікрокліматичні умови, освітлення, електромагнітні випромінювання, забруднення повітря шкідливими речовинами (джерелом яких може бути принтер, сканер та ін.), шум, вібрація, електричний струм, електростатичне поле, напруженість трудового процесу та інше.

Аналіз небезпечних та шкідливих виробничих факторів виконується у табличній формі (табл. 3.3).

Таблиця 3.3 - Аналіз небезпечних і шкідливих виробничих факторів

Небезпечні і шкідливі виробничі фактори	Джерела факторів (види робіт)	Кількісна оцінка	Нормативні документи
1	2	3	4
фізичні			
- підвищена температура поверхонь обладнання	експлуатація ЕОМ	2	ДСН 3.3.6.042-99
- підвищений рівень шуму на робочому місці	Система охолодження ЕОМ	2	ДСН 3.3.6.037-99
- підвищений рівень вібрації	Система охолодження ЕОМ, привід	2	ДСН 3.3.6.039-99 ДСТУ ГОСТ 12.1.012-90
- недостатність природного світла	порушення умов праці (вимог до приміщень)	2	ДБН В.2.5-28:2015
- недостатнє освітлення робочої зони	порушення гігієнічних параметрів виробничого середовища	3	ДБН В.2.5-28:2015
- підвищена яскравість світла	порушення умов праці (організації місця праці-налагодження моніторів)	1	ДСанПіН 3.3.2.007-98
психофізіологічні:			
- нервово-психічна перевантаження (розумове, перенапруження аналізаторів-зорових)	- пошук інформації для постановки теми; - пошук та аналіз аналогів і літератури; - пошук наявних технологій, моделювання та аналіз алгоритмів; - виконання роботи за темою диплома, тестування; - оформлення роботи	4	НПАОП 0.00-1.28-10 ДСанПіН 3.3.2.007-98
- фізичні (статичне – сидіння)	порушення умов праці (організації місця праці-сидіння користувача,) та організації робочого часу - безпервна робота)	2	НПАОП 0.00-1.28-10 ДСанПіН 3.3.2.007-98

3.8 Пожежна безпека

Пожежна безпека при застосуванні ЕОМ забезпечується:

- системою запобігання пожежі,
- системою протипожежного захисту,
- організаційно-технічними заходами.

Згідно НАПБ Б.03.002-2007 таке приміщення, площею 25 м², відноситься до категорії "В" (пожежонебезпечної) та для протипожежного захисту в ньому проектом

передбачено устаткування автоматичною пожежною сигналізацією із застосуванням датчиків-сповіщувачів РІД-1 (сповіщувач димовий ізоляційний) в кількості 1 шт., і застосуванням первинних засобів пожежогасіння. Відповідно до норм первинних засобів пожежогасінні пропонується використовувати:

- ручний вуглекислий вогнегасник ОУ-5 в кількості 1 шт. або хімічний пінний ОХП-10 – 1 шт;
- повсть 1 1 м², кошму 2×1,5 м² або азбестове полотно 2×2 м² в кількості 1 шт.

Виникнення пожежі можливе, якщо на об'єкті є горючі речовини, окислювач і джерела запалювання. Вірогідність пожежної небезпеки приймається значною, якщо ймовірна взаємодія цих трьох чинників. Горючими компонентами є: будівельні матеріали для акустичної і естетичної обробки приміщень, перегородки, підлоги, двері, ізоляція силових, сигнальних кабелів і т.д.

Горючими матеріалами в приміщенні, де розташовані ЕОМ, є:

- поліамід – матеріал корпусу мікросхем, горюча речовина, температура самозаймання 420 °С,
- полівінілхлорид – ізоляційний матеріал, горюча речовина, температура запалювання 335 °С, температура самозаймання 530 °С,
- склотекстоліт ДЦ – матеріал друкарських плат, важкогорючий матеріал, показник горючості 1.74, не схильний до температурного самозаймання,
- пластикат кабельний №.489 – матеріал ізоляції кабелів, горючий матеріал, показник горючості більше 2.1,
- деревина – будівельний і обробний матеріал, з якого виготовлені меблі, горючий матеріал, показник горючості більше 2.1, температура запалювання 255 °С, температура самозаймання 399 °С.

Для відводу теплоти від ЕОМ діє система кондиціонування. Тому кисень, як окиснювач процесів горіння, є в будь-якій точці приміщень ВЦ.

Простори усередині приміщень в межах, яких можуть утворюватися або знаходиться пожежонебезпечні речовини і матеріали відповідно до [27] відносяться до пожежонебезпечної зони класу П-Па. Це обумовлено тим, що в приміщенні знаходяться тверді горючі та важкозаймісті речовини та матеріали. Приміщенню, у якому розташоване робоче місце, присвоюється II ступень вогнестійкості.

Потенційними джерелами запалювання можуть бути:

- іскри і дуги короткого замикання;
- електрична іскра при замиканні і розмиканні ланцюгів;
- перегрів від тривалого перевантаження,

- відкритий вогонь і продукти горіння,
- наявність речовин, нагрітих вище за температуру самозаймання,
- розрядна статична електрика.

Причинами можливого загоряння і пожежі можуть бути:

- несправність електроустановки;
- конструктивні недоліки устаткування;
- коротке замикання в електричних мережах;
- запалювання горючих матеріалів, що знаходяться в безпосередній близькості від електроустановки.

Продуктами згорання, що виділяються на пожежі, є: окис вуглецю; сірчистий газ; окис азоту; синильна кислота; акромін; фосген; хлор і ін. При горінні пластмас, окрім звичних продуктів згорання, виділяються різні продукти термічного розкладання: хлорангідридні кислоти, формальдегіди, хлористий водень, фосген, синильна кислота, аміак, фенол, ацетон, стирол [20].

Для захисту персоналу від дії небезпечних і шкідливих чинників пожежі проектом передбачається застосування промислового протигазу, що фільтрує, з коробкою марки «В» із сірою відміткою забарвлення – захист від неорганічних газів (хлор, фтор, бром, сірководень, сірковуглець, хлорціан, галогени), а цей фільтр не захистить від СО (тобто від чадного газу).

Можливе також відповідне застосування фільтрувальної коробки з маркуванням «СО» із фіолетовим забарвленням на фільтрі означає, що він захищає від Чадного газу. Або фільтру для протигазу з літерним маркуванням «SX» із фіолетовим забарвленням захистить від спец речовин таких як (зарин, зоман та фосген).

3.9 Електробезпека

На робочому місці виконуються наступні вимоги електробезпеки: ПК, периферійні пристрої та устаткування для обслуговування, електропроводи і кабелі за виконанням та ступенем захисту відповідають класу зони за ПУЕ (правила улаштування електроустановок), мають апаратуру захисту від струму короткого замикання та інших аварійних режимів. Лінія електромережі для живлення ПК, периферійних пристроїв і устаткування для обслуговування, виконана як окрема групова три-провідна мережа, шляхом прокладання фазового, нульового робочого та нульового захисного провідників. Нульовий захисний провідник використовується для заземлення (занулення) електроприймачів. Штепсельні з'єднання та електророзетки крім контактів фазового та

нульового робочого провідників мають спеціальні контакти для підключення нульового захисного провідника. Електромережа штепсельних розеток для живлення персональних ПК, укладено по підлозі поруч зі стінами відповідно до затвердженого плану розміщення обладнання та технічних характеристик обладнання. Металеві труби та гнучкі металеві рукави заземлені. Захисне заземлення включає в себе заземлюючих пристроїв і провідник, який з'єднує заземлюючий пристрій з обладнанням, яке заземлюється - заземлюючий провідник.

3.10 Мікроклімат

Мікроклімат робочих приміщень – це клімат внутрішнього середовища цих приміщень, що визначається діючої на організм людини з'єднанням температури, вологості, швидкості переміщення повітря. В даному приміщенні проводяться роботи, що виконуються сидячи і не потребують динамічного фізичного напруження, то для нього відповідає категорія робіт Ia. Отже оптимальні значення для температури, відносної вологості й рухливості повітря для зазначеного робочого місця відповідають [13] і наведені в таблиці 3.4.

Таблиця 3.4 – Норми мікроклімату робочої зони об'єкту

Період року	Категорія робіт	Температура С ⁰	Відносна вологість %	Швидкість руху повітря, м/с
Холодна	легка-1 а	22 - 24	40 – 60	0,1
Тепла	легка-1 а	23 - 25	40 – 60	0,1

Дане приміщення обладнане системами опалення, кондиціонування повітря або припливно-витяжною вентиляцією. У приміщенні на робочому місці забезпечуються оптимальні значення параметрів мікроклімату: температури, відносної вологості й рухливості повітря у відповідності до [13]. Рівні позитивних і негативних іонів у повітрі мають відповідати [13]. Для забезпечення оптимальних параметрів мікроклімату в приміщенні проводяться перерви в роботі користувача, з метою його провітрювання. Існують спеціальні системи кондиціонування, які забезпечують підтримання в приміщенні балансу оптимальних параметрів мікроклімату. Контроль параметрів мікроклімату в холодний і теплий період року здійснюється не менше 3-х разів на зміну (на початку, середині, в кінці).

3.11 Освітлення робочого місця

Світло є природною умовою існування людини. Воно впливає на стан вищих психічних функцій і фізіологічні процеси в організмі. Хороше освітлення діє тонізуюче, створює гарний настрій, покращує протікання основних процесів вищої нервової діяльності.

Збільшення освітленості сприяє поліпшенню працездатності навіть в тих випадках, коли процес праці практично не залежить від зорового сприйняття. При поганому освітленні людина швидко втомлюється, працює менш продуктивно, виникає потенційна небезпека помилкових дій і нещасних випадків.

Освітленість приміщення має велике значення при роботі на ПЕОМ. Вона багато в чому визначається колірною і мережевий обстановкою. Для зменшеного поглинання світла стеля і стіни вище панелей (1,5-1,7м.). Якщо вони не облицьовані звукопоглинальним матеріалом, фарбуються білою водоемульсійною фарбою (коефіцієнт відбиття повинен бути не менше 0,7). Для забарвлення стіни панелей рекомендується віддавати перевагу світлим фарбам.

Основний потік природного світла при цій повинен бути зліва. Не допускається спрямування основного світлового потоку природного світла праворуч, ззаду і спереду працівника на ПЕОМ.

Природне освітлення, коли робочі місця з ПЕОМ розташовуються в один ряд по довжині приміщення на відстані 0,8 - 1,0 м від стіни з віконними прорізами, і екрани знаходяться перпендикулярно цієї стіни. Основний потік природного світла при цій повинен бути зліва. Не допускається спрямування основного світлового потоку природного світла праворуч, ззаду і спереду працює на ПЕОМ. Оптимальна відстань очей до екрана відео монітора повинна становити 60-70 см, допустиме не менше 50 см. Розглядати інформацію ближче 50 см не рекомендується.

У проєкті, що розробляється, передбачається використовувати суміщене освітлення. У світлий час доби використовуватиметься природне освітлення приміщення через віконні отвори, в решту часу використовуватиметься штучне освітлення. Штучне освітлення створюється газорозрядними лампами.

Штучне освітлення в робочому приміщенні передбачається здійснювати з використанням люмінесцентних джерел світла в світильниках загального освітлення, оскільки люмінесцентні лампи мають високу потужність (80 Вт), тривалий термін служби (до 10000 годин), спектральний складом випромінюваного світла, близький до сонячного. При експлуатації ЕОМ виконується зорова робота IV в розряді точності (середня

точність). При цьому нормована освітленість на робочому місці (E_n) рівна 200 лк. Джерелом природного освітлення є сонячне світло.

У приміщенні, де розташовані ЕОМ передбачається природне бічне освітлення, рівень якого відповідає СНіП 11-4-79 [13]. Джерелом природного освітлення є сонячне світло. Регулярно повинен проводитися контроль освітленості, який підтверджує, що рівень освітленості задовольняє СНіП і для даного приміщення в світлий час доби достатньо природного освітлення.

Розрахунок освітлення.

Для будівель виробництв світловий коефіцієнт приймається в межах 1/6 - 1/10:

$$\sqrt{a^2 + b^2} \cdot S_b = (1/8 \div 1/10) \cdot S_n \quad (3.1)$$

де S_b – площа віконних прорізів, м²;

S_n – площа підлоги, м².

$$S_n = a \cdot b = 5 \cdot 5 = 25 \text{ м}^2$$

$$S_{\text{вік}} = 1/8 \cdot 25 = 3,125 \text{ м}^2$$

Приймаємо 2 вікна площею $S = 1,6 \text{ м}^2$ кожне.

Світильники загального освітлення розташовуються над робочими поверхнями в рівномірно-прямокутному порядку. Для організації освітлення в темний час доби передбачається обладнати приміщення, довжина якого складає 5 м, ширина 5 м, світильниками ЛПО2П, оснащеними лампами типа ЛБ (дві по 80 Вт) з світловим потоком 5400 лм кожна.

Розрахунок штучного освітлення виробляється по коефіцієнтах використання світлового потоку, яким визначається потік, необхідний для створення заданої освітленості при загальному рівномірному освітленні. Розрахунок кількості світильників N виробляється по формулі (4.2.2):

$$n = \frac{E \cdot S \cdot Z \cdot K}{F \cdot U \cdot M} \quad (3.2)$$

де E – нормована освітленість робочої поверхні, визначається нормами – 300 лк;

S – освітлювана площа, м²; $S = 25 \text{ м}^2$;

Z – поправочний коефіцієнт світильника (для стандартних світильників $Z = 1.1$ - 1.3) приймаємо рівним 1,1;

K – коефіцієнт запасу, що враховує зниження освітленості в процесі експлуатації – 1,5;

U – коефіцієнт використання, залежний від типу світильника, показника індексу приміщення і т.п. – 0,575

M – число люмінесцентних ламп в світильнику – 2;

F – світловий потік лампи – 5400лм.

Підставивши числові значення у формулу (3.1), отримуємо:

$$n = \frac{300 \cdot 25 \cdot 1,1 \cdot 1,5}{5400 \cdot 0,575 \cdot 2} \approx 2,64$$

Приймаємо освітлювальну установку, яка складається з 3-х світильників, які складаються з двох люмінесцентних ламп загальною потужністю 160 Вт, напругою – 220 В.

Потужність електроосвітлювальної установки з урахуванням місцевого освітлення визначається за формулою:

$$N = \frac{n \cdot W + (0,1 \div 0,2) \cdot n \cdot W}{1000}, \text{ кВт} \quad (3.3)$$

де n – розрахункова кількість ламп для освітлення даного приміщення;

W – потужність однієї лампи, Вт;

$(0,1 \div 0,2)$ – додаткова потужність для ламп місцевого освітлення, Вт

$$N = \frac{3 \cdot 160 + 0,2 \cdot 3 \cdot 160}{1000} = 0,576 \text{ кВт}$$

3.12 Шум, вібрація та електромагнітне випромінювання

Рівень шуму, що супроводжує роботу користувачів персональних комп'ютерів, а також зовнішніми чинниками, коливається у межах 50–65 дБА (ДСН 3.3.6.037-99). Шум такої інтенсивності на тлі високого ступеня напруженості праці негативно впливає на функціональний стан користувачів. Тому на практиці рекомендують знижувати фактичний рівень шуму у приміщеннях, де створюють комп'ютерні програми, виконують теоретичні та творчі роботи, проводять навчання до 40 дБА, а в приміщеннях, де

виконують роботу, що потребує зосередженості, — до 55 дБА. У залах опрацювання інформації та комп'ютерного набору рівні шуму не повинні перевищувати 65 дБА.

Для зниження шуму на шляху його поширення передбачається розміщення в приміщенні штучних поглиначів. Для зниження рівня шуму стелю або стіни вище 1.5 - 1.7 метра від підлоги повинні облицьовуватися звукопоглинальним матеріалом з максимальним коефіцієнтом звукопоглинання в області частот 63-8000 Гц. Додатковим звукопоглинанням в КВТ можуть бути фіранки, підвішені в складку на відстані 15-20 см. Від огорожі, виконані з щільної, важкої тканини. У приміщенні з ЕОМ коректований рівень звукової потужності не перевищує 45 дБА. Оскільки рівень шуму не перевищує гранично допустимих величин, які встановлені санітарними нормами, заходи для зниження шуму не проводяться.

Віброізоляція можливо здійснювати за допомогою спеціальної прокладки під системний блок, який послаблює передачу вібрацій робочого столу. Вібрація на робочому місці в приміщенні, що розглядається, відповідає нормам [26]. Допустимий рівень вібрацій на робочому місці: - для 1 ступеня шкідливості до 3 дБ; - для 2-3 - 1-6 дБ; - для 3 - більше 6 дБ.

Для захисту від електромагнітного випромінювання передбачаються наступні заходи:

- застосування нових плазмових моніторів,
- віддалення робочого місця не менше, ніж на 0,4 – 0,5 м, оскільки напруженість електричного поля зменшується при віддаленні від джерела поля,
- встановлення раціональних режимів роботи персоналу (обмеження часу перебування),
- раціональне розміщення в робочому приміщенні устаткування, що випромінює електромагнітну енергію.

3.13 Розрахунок захисного заземлення

Згідно з класифікацією приміщень за ступенем небезпеки ураження електричним струмом [22], приміщення в якому проводяться всі роботи відноситься до першого класу (без підвищеної небезпеки). Під час роботи використовуються електроустановки з напругою живлення 36 В, 220 В, та 360 В. Опір контура заземлення повинен мати не більше 4 Ом.

Розрахунок проводять за допомогою методу коефіцієнта використання (екранування) електродів. Коефіцієнт використання групового заземлювача η – це

відношення діючої провідності цього заземлювача до найбільш можливої його провідності за нескінченно великих відстаней між його електродами. Коефіцієнт використання вертикальних заземлювачів η_v в залежності від розміщення заземлювачів та їх кількості знаходиться в межах 0,4...0,99. Взаємну екрануючу дію горизонтального заземлювача (з'єднувальної смуги) враховують за допомогою коефіцієнта використання горизонтального заземлювача η_c .

Визначимо необхідний опір штучних заземлювачів $R_{шт.з.}$:

$$R_{шт.з.} = \frac{R_d \cdot R_{пр.з.}}{R_{пр.з.} - R_d}, \quad (3.4)$$

—де $R_{пр.з.}$ — опір природних заземлювачів; R_d — допустимий опір заземлення. Якщо природні заземлювачі відсутні, то $R_{шт.з.} = R_d$.

Підставивши числові значення у формулу (А.3), отримуємо:

$$R_{шт.з.} = \frac{4 \cdot 40}{40 - 4} \approx 4 \text{ Ом}$$

Опір заземлення в значній мірі залежить від питомого опору ґрунту ρ , Ом•м. Приблизне значення питомого опору глини приймаємо $\rho = 40$ Ом•м (табличне значення).

Розрахунковий питомий опір ґрунту, $\rho_{розр.}$, Ом•м, визначається відповідно для вертикальних заземлювачів $\rho_{розр.в.}$, і горизонтальних $\rho_{розр.г.}$, Ом•м за формулою:

$$\rho_{розр.} = \psi \cdot \rho, \quad (3.5)$$

—де ψ — коефіцієнт сезонності для вертикальних заземлювачів І кліматичної зони з нормальною вологістю землі, приймається для вертикальних заземлювачів $\rho_{розр.в.} = 1,7$ і горизонтальних $\rho_{розр.г.} = 5,5$ Ом•м.

$$\rho_{розр.в.} = 1,7 \cdot 40 = 68 \text{ Ом•м}$$

$$\rho_{розр.г.} = 5,5 \cdot 40 = 220 \text{ Ом•м}$$

Розрахуємо опір розтікання струму вертикального заземлювача R_v , Ом, за (А.5).

$$R_v = \frac{\rho_{розр.в.}}{2 \cdot \pi \cdot l_b} \cdot \left(\ln \frac{2 \cdot l_b}{d_{ст}} + \frac{1}{2} \cdot \ln \frac{4 \cdot t + l_b}{4 \cdot t - l_b} \right), \quad (3.6)$$

—де l_b — довжина вертикального заземлювача (для труб - 2–3 м; $l_b = 3$ м);

– $d_{ст}$ – діаметр стержня (для труб - 0,03–0,05 м; $d_{ст}=0,05$ м);

– t – відстань від поверхні землі до середини заземлювача, яка визначається за ф.

(3.6):

$$t = h_b + \frac{l_b}{2}, \quad (3.7)$$

–де h_b – глибина закладання вертикальних заземлювачів (0,8 м);

$$\text{тоді } t = 0,8 + \frac{3}{2} = 2,3 \text{ м}$$

$$R_b = \frac{68}{2 \cdot \pi \cdot 3} \cdot \left(\ln \frac{2 \cdot 3}{0,05} + \frac{1}{2} \cdot \ln \frac{4 \cdot 2,3 + 3}{4 \cdot 2,3 - 3} \right) = 18,5 \text{ Ом}$$

Визначаємо теоретичну кількість вертикальних заземлювачів n штук, без урахування коефіцієнта використання η_b :

$$n = \frac{2 \cdot R_b}{R_d} = \frac{2 \cdot 18,5}{4} = 9,25 \quad (3.8)$$

Визначаємо коефіцієнт використання вертикальних електродів групового заземлювача без врахування впливу з'єднувальної стрічки $\eta_b = 0,57$ (табличне значення).

Визначаємо необхідну кількість вертикальних заземлювачів з урахуванням коефіцієнта використання n_b , шт:

$$n_b = \frac{2 \cdot R_b}{R_d \cdot \eta_b} = \frac{2 \cdot 18,5}{4 \cdot 0,57} = 16,2 \approx 16 \quad (3.9)$$

Визначаємо довжину з'єднувальної стрічки горизонтального заземлювача l_c , м:

$$l_c = 1,05 \cdot L_b \cdot (n_b - 1), \quad (3.10)$$

–де L_b – відстань між вертикальними заземлювачами, (прийняти за $L_b = 3$ м);

– n_b – необхідна кількість вертикальних заземлювачів.

$$l_c = 1,05 \cdot 3 \cdot (16 - 1) \approx 48 \text{ м}$$

Визначаємо опір розтіканню струму горизонтального заземлювача (з'єднувальної стрічки) R_r , Ом:

$$R_{\Gamma} = \frac{\rho_{\text{розр.}\Gamma}}{2 \cdot \pi \cdot l_c} \cdot \ln \frac{2 \cdot l_c^2}{d_{\text{см}} \cdot h_{\Gamma}}, \quad (3.11)$$

–де $d_{\text{см}}$ – еквівалентний діаметр смуги шириною b , $d_{\text{см}} = 0,95b$, $b = 0,15$ м;

– h_{Γ} – глибина закладання горизонтальних заземлювачів (0,5 м);

– l_c - довжина з'єднувальної стрічки горизонтального заземлювача l_c , м

$$R_{\Gamma} = \frac{220}{2 \cdot \pi \cdot 48} \cdot \ln \frac{2 \cdot 48^2}{0,95 \cdot 0,15 \cdot 0,5} = 8,1 \text{ Ом}$$

Визначаємо коефіцієнт використання горизонтального заземлювача η_c . відповідно до необхідної кількості вертикальних заземлювачів n_b .

Коефіцієнт використання з'єднувальної смуги $\eta_c = 0,3$ (табличне значення).

Розраховуємо результуючий опір заземлювального електроду з урахуванням з'єднувальної смуги:

$$R_{\text{заг}} = \frac{R_b \cdot R_{\Gamma}}{R_b \cdot \eta_c + R_{\Gamma} \cdot n_b \cdot \eta_b} \leq R_d. \quad (3.12)$$

Висновок: дане захисне заземлення буде забезпечувати електробезпеку будівлі, так як виконується умова: $R_{\text{заг}} < 4$ Ом, а саме:

$$R_{\text{заг}} = \frac{18,5 \cdot 8,1}{18,5 \cdot 0,3 + 8,1 \cdot 16 \cdot 0,57} = 1,9 \leq R_d$$

Висновки до розділу 3

В даному розділі дипломного проекту зроблено аналіз умов праці, шкідливих та небезпечних чинників. Визначено параметри і характеристики приміщення для роботи над дипломним проектом, заходи, які потрібно зробити для того, щоб дане приміщення відповідало необхідним нормам і було комфортним і безпечним для роботи. Приведені рекомендації щодо організації робочого місця, електробезпеки та пожежної безпеки. Наведена схема, розміри приміщення та наведено значення температури, вологості й рухливості повітря, необхідна кількість і потужність ламп та інші параметри, значення яких впливає на умови праці, рекомендації з охорони праці, техніки безпеки при роботі на комп'ютері.

ВИСНОВКИ

В дипломній роботі виконано розроблення драйверу підтримки послідовної шини USB з використанням мікроконтролерів ATtiny2313 та ATmega8/48/88/168.

В розробленому пристрої реалізовані наступні функції:

- безпосереднє керування лінією вводу-виводу;
- перетворення USB – RS232.

Для реалізації таких функцій використана DLL-бібліотека AVR309.dll фірми Atmel.

Розроблені заходи щодо охорони праці та безпеки в надзвичайних ситуаціях.

ПЕРЕЛІК ПОСИЛАНЬ

1. Трамперт В. AVR–RISC мікроконтролери.: Пер. с нем. – К.: «МК– Пресс», 2006. – 464 с. , ил.
2. Королев Н. AVR: Программирование в среде AVR Studio. / Николай Королев, Дмитрий Королев / Компоненты и технологии. - № 3 – 2004. С. 15 –20.
3. Гумеров Р.И. Практикум по микропроцессорам. Часть первая: микроконтроллеры AVR. Руководство. – Казань: КГУ, 2009. – 37 с.
4. Хартов В.Я. Микроконтроллеры AVR. Практикум для начинающих. – М.: Изд–во МГТУ им. Н.Э. Баумана, 2007. – 240 с.: ил.
5. Интернет ресурс: www.atmel.com
6. Гребнев В.В. Микроконтроллеры семейства AVR фирмы Atmel / Гребнев В.В. – М.: ИП РадиоСофт, 2002. – 176 с.: ил.
7. Белов А.В. Самоучитель разработчика устройств на микроконтроллерах AVR / Белов А.В. – СПб.: Наука и техника, 2008. – 544 с.: ил.
8. Предко М. Руководство по микроконтроллерам. Том. II / Предко М. – М.: Постмаркет, 2001. – 488с.
9. Микушин А.В. Занимательно о микроконтроллерах. – СПб.: БХВ–Петербург, 2006. – 432 с.: ил.
10. Закон України «Про охорону праці».
11. НПАОП 0.00-6.03-93 «Порядок опрацювання та затвердження власником нормативних актів про охорону праці, що діють на підприємстві».
12. НПАОП 0.00-4.12-05. «Типове положення про порядок проведення навчання і перевірки знань з питань охорони праці».
13. ДСН 3.3.6.042-99 «Санітарні норми мікроклімату виробничих приміщень».
14. ДСанПіН 3.3.2.007-98 «Правила і норми роботи з візуальними дисплейними терміналами електронно-обчислювальних машин».
15. ДБН В.2.5-28:2015 Природне і штучне освітлення.
16. Закон України Про забезпечення санітарного та епідемічного благополуччя населення.
17. НПАОП 0.00-1.28-10 Правила охорони праці під час експлуатації електронно-обчислювальних машин.
18. НПАОП 0.00-8.24-05 Перелік робіт з підвищеною небезпекою.
19. НАПБ А. 01.001-2004 Правила пожежної безпеки України.

20. ГОСТ 12.1.044-89 ССБТ. Пожаровзрывоопасность веществ и материалов. Номенклатура показателей и методы их определения.
21. НАПБ.Б.06.004-2005 Правила пожарной безопасности в Украине.
22. НПАОП 40.1-1.01-97 Правила безопасной эксплуатации электроустановок.
23. НПАОП 40.1-1.21-98 Правила безпечної експлуатації електроустановок споживачів.
24. ГОСТ 12.1.018-93 ССБТ.Пожаровзрывобезопасность статического электричества. Общие требования.
25. ГОСТ 12.1.030-81 ССБТ. Электробезопасность. Защитное заземление. Зануление.
26. ДСН 3.3.6.037-99 «Санітарні норми виробничого шуму, ультразвуку та інфразвуку».
27. НАПБ Б.03.002-2007. Норми визначення категорій приміщень, будинків та зовнішніх установок за вибухопожежною та пожежною безпекою

ДОДАТОК А

USB ДРАЙВЕР ДЛЯ СІМЕЙСТВА МІКРОКОНТРОЛЕРІВ AVR

```

;*****
;* USB driver for AVR FAMILY USB stack
;*
;* File Name      : "AVR309.sys"
;* Title          : AVR309:USB driver for "USB to UART protocol converter"
;* Date           : 22.05.2017
;* Version        : 1.0
;* Target Platform : Windows 7
;* AUTHOR         : Boychuk Artur
;*               : Ukraine
;* DESCRIPTION:
;* USB driver for device "USB stack in AVR microcontrollers" (USB protocol
;* implementation into MCU with noUSB interface)
;* Device:
;* Universal USB interface (3x8-bit I/O port + RS232 serial line + EEPROM)
;* + added RS232 FIFO buffer
;*
;* DRIVER CUSTOMIZATION AND MODIFICATION:
;* This driver is modified example "isusb.sys" from
;* Microsoft Windows 2000 DDK (Driver Development Kit):
;* original "isusb.sys" is located in DDK sample directory:
;* "\NTDDK\src\wdm\usb\isusb"
;*
;* Modified part of "isusb" files is described in document:
;* "\PDF doc\Differences to isusb DDK.pdf"
;*
;* To customize device driver to your purpose:
;* - change driver link name "AVR309USB_0" in files "isopnp.c" and "isusb.c"
;*   to your name
;* - set target SYS filename "TARGETNAME=" to your name in file "sources"
;* - modify driver information in file "AVR309.rc"

```

;* - modify registry path "ISOUSB_REGISTRY_PARAMETERS_PATH" in file "iusbdbg.h"

;*

Module Name:

IsoUsb.c

Abstract:

Isochronous USB device driver for Intel 82930 USB test board

Main module

Environment:

kernel mode only

Notes:

THIS CODE AND INFORMATION IS PROVIDED "AS IS" WITHOUT WARRANTY OF ANY

KIND, EITHER EXPRESSED OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE

IMPLIED WARRANTIES OF MERCHANTABILITY AND/OR FITNESS FOR A PARTICULAR

PURPOSE.

11/06/13: created

#include "wdm.h"

#include "stdarg.h"

#include "stdio.h"

#include "usbdi.h"

#include "usbdlb.h"

#include "Iso82930.h"

#include "GUIDISO.h"

NTSTATUS

DriverEntry(

IN PDRIVER_OBJECT DriverObject,

IN PUNICODE_STRING RegistryPath

Routine Description:

Installable driver initialization entry point.

This entry point is called directly by the I/O system.

Arguments:

DriverObject - pointer to the driver object

RegistryPath - pointer to a unicode string representing the path
to driver-specific key in the registry

Return Value:

STATUS_SUCCESS if successful,
STATUS_UNSUCCESSFUL otherwise

```
{
    NTSTATUS ntStatus = STATUS_SUCCESS;
    PDEVICE_OBJECT deviceObject = NULL;
    BOOLEAN fRes;
#if DBG
    // should be done before any debug output is done.
    // read our debug verbosity level from the registry
    IsoUsb_GetRegistryDword( ISOUSB_REGISTRY_PARAMETERS_PATH, //absolute
registry path
                                L"DebugLevel", // REG_DWORD ValueName
                                &gDebugLevel ); // Value receiver
#endif
    ISOUSB_KdPrint( DBGLVL_MINIMUM ,("Entering DriverEntry(), RegistryPath=\n
%ws\n", RegistryPath->Buffer ));
    //
    // Create dispatch points for create, close, unload
    DriverObject->MajorFunction[IRP_MJ_CREATE] = IsoUsb_Create;
    DriverObject->MajorFunction[IRP_MJ_CLOSE] = IsoUsb_Close;
    DriverObject->DriverUnload = IsoUsb_Unload;
    // User mode DeviceIoControl() calls will be routed here
    DriverObject->MajorFunction[IRP_MJ_DEVICE_CONTROL] =
IsoUsb_ProcessIOCTL;
    // routines for handling system PNP and power management requests
    DriverObject->MajorFunction[IRP_MJ_SYSTEM_CONTROL] =
IsoUsb_ProcessSysControlIrp;
    DriverObject->MajorFunction[IRP_MJ_PNP] = IsoUsb_ProcessPnPirp;
    DriverObject->MajorFunction[IRP_MJ_POWER] = IsoUsb_ProcessPowerIrp;
    // The Functional Device Object (FDO) will not be created for PNP devices until
    // this routine is called upon device plug-in.
    DriverObject->DriverExtension->AddDevice = IsoUsb_PnPAddDevice;
```

```

    ISOUSB_KdPrint( DBGLVL_DEFAULT,("exiting DriverEntry (%x)\n", ntStatus));
    return ntStatus;
}

```

NTSTATUS

```

IsoUsb_ProcessSysControlIrp(
    IN PDEVICE_OBJECT DeviceObject,
    IN PIRP          Irp
)

```

/*++

Routine Description:

Main dispatch table routine for IRP_MJ_SYSTEM_CONTROL

We basically just pass these down to the PDO

Arguments:

DeviceObject - pointer to FDO device object

Irp - pointer to an I/O Request Packet

Return Value:

Status returned from lower driver

PIO_STACK_LOCATION irpStack;

PDEVICE_EXTENSION deviceExtension;

NTSTATUS ntStatus = STATUS_SUCCESS;

NTSTATUS waitStatus;

PDEVICE_OBJECT stackDeviceObject;

Irp->IoStatus.Status = STATUS_SUCCESS;

Irp->IoStatus.Information = 0;

// Get a pointer to the current location in the Irp. This is where

// the function codes and parameters are located.

irpStack = IoGetCurrentIrpStackLocation (Irp);

//

// Get a pointer to the device extension

deviceExtension = DeviceObject->DeviceExtension;

stackDeviceObject = deviceExtension->TopOfStackDeviceObject;

ISOUSB_KdPrint(DBGLVL_HIGH, ("enter IsoUsb_ProcessSysControlIrp()\n"));

IsoUsb_IncrementIoCount(DeviceObject);

ISOUSB_ASSERT(IRP_MJ_SYSTEM_CONTROL == irpStack->MajorFunction);

IoCopyCurrentIrpStackLocationToNext(Irp);

```

        ntStatus = IoCallDriver(stackDeviceObject,
                                Irp);
        IsoUsb_DecrementIoCount(DeviceObject);
        ISOUSB_KdPrint(         DBGLVL_HIGH,("IsoUsb_ProcessSysControlIrp()      Exit
IsoUsb_ProcessSysControlIrp %x\n", ntStatus));
        return ntStatus;

```

```

}VOID

```

```

IsoUsb_Unload(
    IN PDRIVER_OBJECT DriverObject
)

```

```

/*++

```

Routine Description:

Free all the allocated resources, etc.

Arguments:

DriverObject - pointer to a driver object

Return Value:

```

--*/

```

```

{
    ISOUSB_KdPrint( DBGLVL_HIGH,("enter IsoUsb_Unload\n"));
    //
    // Free any global resources allocated
    // in DriverEntry.
    // We have few or none because for a PNP device, almost all
    // allocation is done in PnpAddDevice() and all freeing
    // while handling IRP_MN_REMOVE_DEVICE:
    //
    ISOUSB_KdPrint( DBGLVL_DEFAULT,("exit IsoUsb_Unload\n"));
}

```

NTSTATUS

```

IsoUsb_SymbolicLink(
    IN PDEVICE_OBJECT DeviceObject,
    IN OUT PUNICODE_STRING deviceLinkUnicodeString
)

```

```

/*++

```

Routine Description:

This routine is called to create and initialize a GUID-based symbolic link to our device to be used to open/create instances of us from user mode.

Called from IsoUsb_CreateDeviceObject() to create the link.

Arguments:

DeviceObject - pointer to our Physical Device Object (PDO)

deviceLinkUnicodeString - Points to a unicode string structure allocated by the caller.

If this routine is successful, it initializes the unicode string and allocates the string buffer containing the kernel-mode path to the symbolic link for this device interface.

Return Value:

STATUS_SUCCESS if successful,

STATUS_UNSUCCESSFUL otherwise

--*/{

```
NTSTATUS ntStatus = STATUS_SUCCESS;
```

```
// Create the symbolic link
```

```
// IoRegisterDeviceInterface registers device functionality (a device interface)
```

```
// that a driver will enable for use by applications or other system components.
```

```
ntStatus = IoRegisterDeviceInterface(
```

```
    DeviceObject,
```

```
    (LPGUID)&GUID_CLASS_I82930_ISO,
```

```
    NULL,
```

```
    deviceLinkUnicodeString);
```

```
ISOUSB_KdPrintCond( DBGLVL_MEDIUM, (!(NT_SUCCESS(ntStatus))),
```

```
    ("FAILED to IoRegisterDeviceInterface()\n"));
```

```
if (NT_SUCCESS(ntStatus)) {
```

```
    // IoSetDeviceInterfaceState enables or disables a previously
```

```
    // registered device interface. Applications and other system components
```

```
    // can open only interfaces that are enabled.
```

```
ntStatus = IoSetDeviceInterfaceState(deviceLinkUnicodeString, TRUE);
```

```
ISOUSB_KdPrintCond( DBGLVL_MEDIUM,
```

```
    (!(NT_SUCCESS(ntStatus))),
```

```
    ("FAILED to IoSetDeviceInterfaceState()\n"));
```

```
ISOUSB_KdPrintCond( DBGLVL_MEDIUM,
```

```
    ((NT_SUCCESS(ntStatus))),
```

```

        ("SUCCEEDED IoSetDeviceInterfaceState()\n"));
    }
    return ntStatus;
}

```

NTSTATUS

```

IsoUsb_CreateDeviceObject(
    IN PDRIVER_OBJECT DriverObject,
    IN PDEVICE_OBJECT PhysicalDeviceObject,
    IN PDEVICE_OBJECT *DeviceObject
)

```

/*++

Routine Description:

Creates a Functional DeviceObject

Arguments:

DriverObject - pointer to the driver object for device

DeviceObject - pointer to DeviceObject pointer to return
created device object.

Instance - instance of the device create.

Return Value:

STATUS_SUCCESS if successful,

STATUS_UNSUCCESSFUL otherwise

--*/

```

{
    NTSTATUS ntStatus;
    UNICODE_STRING deviceLinkUnicodeString;
    PDEVICE_EXTENSION deviceExtension;
    USHORT i;
    UNICODE_STRING pdoUniCodeName;
    WCHAR      pdoName[] = L"\\Device\\AVR309USB_0";
    UNICODE_STRING DeviceLinkUniCodeString;
    WCHAR      DeviceLinkName[] = L"\\DosDevices\\AVR309USB_0";
    RtlInitUnicodeString (&pdoUniCodeName, pdoName); // Creates Unicode Name
    ISOUSB_KdPrint( DBGLVL_DEFAULT, ("enter IsoUsb_CreateDeviceObject() \n"));
    ntStatus = IsoUsb_SymbolicLink( PhysicalDeviceObject, &deviceLinkUnicodeString );
    ISOUSB_KdPrintCond( DBGLVL_DEFAULT,

```

```

        (NT_SUCCESS(ntStatus)),
        ("IsoUsb_CreateDeviceObject() SUCCESS Create GUID_CLASS_IsoUsb_ISO-
based Device name\n %ws\n Length decimal %d, MaximumLength decimal %d\n",
        deviceLinkUnicodeString.Buffer,
        deviceLinkUnicodeString.Length,
        deviceLinkUnicodeString.MaximumLength));
    ISOUSB_KdPrintCond( DBGLVL_DEFAULT,
        (!(NT_SUCCESS(ntStatus))),
        ("IsoUsb_CreateDeviceObject() FAILED to Create GUID_CLASS_ISO-based
Device name\n"));
    if (NT_SUCCESS(ntStatus)) {
        ntStatus = IoCreateDevice (DriverObject,
            sizeof (DEVICE_EXTENSION),
            &pdoUniCodeName,
            FILE_DEVICE_UNKNOWN,
            0,
            FALSE,
            DeviceObject);
        if (NT_SUCCESS(ntStatus)) {
            deviceExtension = (PDEVICE_EXTENSION) ((*DeviceObject)-
>DeviceExtension);
        }
        ISOUSB_KdPrintCond( DBGLVL_DEFAULT,
            (!(NT_SUCCESS(ntStatus))),
            ("IsoUsb_CreateDeviceObject() IoCreateDevice() FAILED\n"));
        ISOUSB_KdPrintCond( DBGLVL_DEFAULT,
            ((NT_SUCCESS(ntStatus))),
            ("IsoUsb_CreateDeviceObject() IoCreateDevice() SUCCESS\n"));
        if (!NT_SUCCESS(ntStatus)) {
            return ntStatus;
        }
        RtlInitUnicodeString (&DeviceLinkUniCodeString, DeviceLinkName);
        ntStatus = IoCreateSymbolicLink(&DeviceLinkUniCodeString,
&pdoUniCodeName);
        //default maximum transfer size per io request

```

```

deviceExtension->MaximumTransferSize = ISO_MAX_TRANSFER_SIZE ;
#if DBG
    // may be overridden in registry
    IsoUsb_GetRegistryDword( ISOUSB_REGISTRY_PARAMETERS_PATH,
        L"MaximumTransferSize",
        &(deviceExtension->MaximumTransferSize) );
#endif
// Name buffer for our named Functional device object link
// The name is generated based on the driver's class GUID
RtlCopyMemory(deviceExtension->DeviceLinkNameBuffer,
    deviceLinkUnicodeString.Buffer,
    deviceLinkUnicodeString.Length);
// this event is triggered when there is no pending io of any kind and device is removed
KeInitializeEvent(&deviceExtension->RemoveEvent, NotificationEvent, FALSE);
// this event is triggered when self-requested power irps complete
KeInitializeEvent(&deviceExtension->SelfRequestedPowerIrpEvent, NotificationEvent,
FALSE);
// this event is triggered when there is no pending io (pending io count == 1 )
KeInitializeEvent(&deviceExtension->NoPendingIoEvent, NotificationEvent, FALSE);
    //free buffer from unicode string we used to init interface
    RtlFreeUnicodeString( &deviceLinkUnicodeString );
}
ISOUSB_KdPrintCond( DBGLVL_DEFAULT,
    ((NT_SUCCESS(ntStatus))),
    ("Exit IsoUsb_CreateDeviceObject() IoCreateDevice() SUCCESS\n"));
return ntStatus;

```