**МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ**

**НАЦІОНАЛЬНИЙ ТЕХНІЧНИЙ УНІВЕРСИТЕТ УКРАЇНИ**

**„КИЇВСЬКИЙ ПОЛІТЕХНІЧНИЙ ІНСТИТУТ” ім. ІГОРЯ СІКОРСЬКОГО”**

**КАФЕДРА ІНФОРМАЦІЙНИХ СИСТЕМ І ТЕХНОЛОГІЙ**

**Пристрій обміну інформацією**

**АААА.123456.007**

**Курсова робота**

З дисципліни «МІКРОПРОЦЕСОРНІ СИСТЕМИ»

Керівник доц. Новацький А.О. Виконавець

„Допущений до захисту” ст. Тельє І.Н.

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ зал. книжка № АААА12

(Особистий підпис керівника) \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

(Особистий підпис виконавця)

«\_\_\_\_» \_\_\_\_\_\_\_\_\_\_\_\_\_ 2022 р.

«\_\_\_» \_\_\_\_\_\_\_\_\_\_\_\_ 2022р.

Захищений з оцінкою

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

(оцінка)

«\_\_\_\_» \_\_\_\_\_\_\_\_\_\_\_\_\_ 2022 р.

Члени комісії :

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

(Особистий підпис) (Розшифровка підпису)

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

(Особистий підпис) (Розшифровка підпису)

Київ 2022

НАЦІОНАЛЬНИЙ ТЕХНІЧНИЙ УНІВЕРСИТЕТ УКРАЇНИ “КПІ” ім. Ігоря Сікорського

Кафедра ІНФОРМАЦІЙНИХ СИСТЕМ ТА ТЕХНОЛОГІЙ \_

Дисципліна Мікропроцесорні системи \_

Курс 2 Група АААА Семестр 4 \_

ЗАВДАННЯ

на курсову роботу студента

|  |
| --- |
| \_\_\_\_\_\_Тельє Ніка Інновича |

(прізвище, ім'я, по батькові)

1.Тема проекту (роботи) \_\_ Пристрій обміну інформацією між модулем УАПП мікроконтролера та віртуальним терміналом

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

2.Строк здачі студентом закінченого проекту (роботи) 16.06.2022р. \_

3.Вихідні дані до проекту (роботи) швидкість передачі: = 4800 бод; тактова частота кварцового резонатора: = 8 МГц; для обміну використовувати модуль UART; режим передачі – асинхронний; розмір слова даних – 7 розрядів; використовувати перевірку на непарність; кількість стоп–бітів – 2; байт для передачі: 01101010b; символ для прийому від віртуального терміналу – n.

4.Зміст розрахунково-пояснювальної записки (перелік питань, які підлягають

розробці) Призначення та галузь застосування пристрою, його технічні характеристики,

вибір структурної схеми та розробка принципової схеми , розрахунки, \_

які підтверджують працездатність пристрою, розробка схеми алгоритму роботи,

керуючої програми та схеми моделювання \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

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

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

6.Дата видачі завдання 15.02.2022р.\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

**КАЛЕНДАРНИЙ ПЛАН**

|  |  |  |  |
| --- | --- | --- | --- |
| №  п/п | Назва етапів курсової роботи | Строк виконання  етапів роботи | Примітки |
| 1 | Отримання завдання на курсову роботу та |  |  |
|  | узгодження вихідних даних | 24.02.2022р. |  |
| 2 | Огляд існуючих рішень та вибір |  |  |
|  | структури пристрою | 04.06.2022р. |  |
| 3 | Виконання основних розрахунків, пов’язаних |  |  |
|  | з розробкою принципової схеми |  |  |
|  | пристрою | 11.06.2022р. |  |
| 4 | Розробка принципової схеми, схеми  алгоритму та керуючої програми | 01.06.2022р. |  |
| 5 | Моделювання пристрою | 11.06.2022р. |  |
| 6 | Оформлення текстової документації | 12.06.2022р. |  |
| 7 | Оформлення графічної конструкторської | 13.06.2022р. |  |
|  | документації |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |

Студент\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

(підпис)

Керівник \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_Анатолій НОВАЦЬКИЙ

*(підпис)* (ім'я та прізвище)

‹‹ 15 ›› 02 2022 р.

**Опис**

|  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| Номер рядка | Формат | Позначення | | | Найменування | | Кіл. листів | | | Номер  екзем. | Примітка | |
| *1* | *А4* | АААА.123456.007 ПЗ | | | Пояснювальна записка | | 29 | | |  |  | |
| *2* | *А4* | АААА.123456.007 ЕЗ | | | Схема електрична структурна | | 1 | | |  |  | |
| *3* |  |  | | |  | |  | | |  |  | |
| *4* |  |  | | |  | |  | | |  |  | |
| *5* |  |  | | |  | |  | | |  |  | |
| *6* |  |  | | |  | |  | | |  |  | |
| *7* |  |  | | |  | |  | | |  |  | |
| *8* |  |  | | |  | |  | | |  |  | |
| *9* |  |  | | |  | |  | | |  |  | |
| *10* |  |  | | |  | |  | | |  |  | |
| *11* |  |  | | |  | |  | | |  |  | |
| *12* |  |  | | |  | |  | | |  |  | |
| *13* |  |  | | |  | |  | | |  |  | |
| *14* |  |  | | |  | |  | | |  |  | |
| *15* |  |  | | |  | |  | | |  |  | |
| *16* |  |  | | |  | |  | | |  |  | |
| *17* |  |  | | |  | |  | | |  |  | |
| *18* |  |  | | |  | |  | | |  |  | |
| *19* |  |  | | |  | |  | | |  |  | |
| *20* |  |  | | |  | |  | | |  |  | |
| *21* |  |  | | |  | |  | | |  |  | |
| *22* |  |  | | |  | |  | | |  |  | |
| *23* |  |  | | |  | |  | | |  |  | |
| *24* |  |  | | |  | |  | | |  |  | |
| *25* |  |  | | |  | |  | | |  |  | |
| *26* |  |  | | |  | |  | | |  |  | |
|  |  |  |  |  | *АААА.123456.007 ОП* | | | | | | | |
|  |  |  |  |  |
|  | Лист | № докум. | Підпис | Дата |
| Розроб. | | Тельє |  |  | *Пристрій обміну інформацією*  *Опис* | Літ. | | | | Лист | | Листів |
| Перевір. | | Новацький |  |  | Т | |  |  | 1 | | 1 |
| Реценз. | |  |  |  | НТУУ “КПІ” ФІОТ  Група АААА | | | | | | |
| Н.контр. | |  |  |  |
| Затверд. | |  |  |  |

|  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| Зміст  [1 ВСТУП 6](#_Toc106016605)  [2 ПРИЗНАЧЕННЯ ТА ОБЛАСТЬ ВИКОРИСТАННЯ 7](#_Toc106016606)  [3 ТЕХНІЧНА ХАРАКТЕРИСТИКА 8](#_Toc106016607)  [4 ОГЛЯД ІСНУЮЧИХ РІШЕНЬ ТА ОБГРУНТУВАННЯ ВИБОРУ СТРУКТУРИ ПРИСТРОЮ 10](#_Toc106016608)  [5 ОПИС РОБОТИ ПРИСТРОЮ ЗА СТРУКТУРНОЮ СХЕМОЮ 11](#_Toc106016609)  [6 ВИБІР ТА ОБГРУНТУВАННЯ ОКРЕМИХ ВУЗЛІВ ЕЛЕМЕНТІВ 13](#_Toc106016610)  [6.1 Мікроконтроллер 13](#_Toc106016611)  [6.2 Інтерфейс УСАПП мікроконтролера ATMEGA16A 16](#_Toc106016612)  [7 ПРОЕКТУВАННЯ ПРИСТРОЮ 17](#_Toc106016613)  [7.1 Розробка та опис робочої моделі в пакеті Proteus 17](#_Toc106016614)  [7.2 Розрахунок модуля УСАПП 18](#_Toc106016615)  [8 РОЗРОБКА І ОПИС СХЕМИ АЛГОРИТМУ ТА КЕРУЮЧОЇ ПРОГРАМИ 23](#_Toc106016616)  [8.1 Схема алгоритму роботи пристрою 23](#_Toc106016617)  [8.2 Керуюча програма мовою Асемблер 24](#_Toc106016618)  [9 МОДЕЛЮВАННЯ ПРИСТРОЮ В ПАКЕТІ PROTEUS 27](#_Toc106016619)  [10 ВИСНОВОК 34](#_Toc106016620)  [11 СПИСОК ЛІТЕРАТУРИ 35](#_Toc106016621) | | | | | | | | | | |
|  |  |  |  |  | АААА.123456.007 ПЗ | | | | | |
|  |  |  |  |  |
| Зм. | Лист | № докум. | Підп. | Дата |
| Розроб. | | Тельє |  |  | Пристрій обміну інформацією  Пояснювальна записка | Літ. | | | Лист | Листів |
| Перевір. | | Новацький |  |  | T |  |  | 1 | 29 |
|  | |  |  |  | НТУУ(КПІ) ФІОТ група АААА | | | | |
| Н. контр. | |  |  |  |
| Затв. | |  |  |  |

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| ВСТУП З настанням нової ери електронних та вбудованих пристроїв знайти мікроконтролер можна майже усюди: він доступний і дітям у вигляді DIY kit конструкторів, і школярам у вигляді Arduino / Raspberry PI, і, звичайно, фахівцям з розробки на Embedded та Embedded ентузіастам.  Паралельно зі зростанням доступності мікроконтролерів зростала й доступність різних дотичних до них пристроїв периферії: сенсорів, екранів, датчиків тощо. Завдяки такому стрибку в розвитку та доступності різних пристроїв, звичайний користувач отримав можливість прототипувати та тестувати практично будь-які ідеї та проєкти.  Наприклад, мікроконтролери використовують у побутовій техніці, медичних приладах, системах керування ліфтами, телефонах, раціях та інших засобах зв’язку, електронних музичних інструментах та автомагнітолах, комп’ютерній периферії (клавіатурах, джойстиках, принтерах, тощо), світлофорах, автоматичних воротах та шлагбаумах, інтерактивних дитячих іграшках, автомобілях, локомотивах та літаках, роботах та промислових верстатах. Мікроконтролери також широко застосовуються в автомобільній електроніці. Наприклад, автомобіль «Peugeot 206» має на борту 27 мікроконтролерів, а в автомобілях високого класу, як наприклад «BMW» сьомої серії, використовується понад 60 мікроконтролерів. Вони керують вприском палива, жорсткістю адаптивної підвіски, світлотехнікою, двигунами двірників, склопідіймачів та дзеркал заднього виду, тощо. (1) | | | | | | |
|  |  |  |  |  | АААА.123456.007 ПЗ | *Лист* |
|  |  |  |  |  |
| 1. *1* |
| *Зм* | *Лист* | *№ докум.* | *Підп.* | *Дата* |
| ПРИЗНАЧЕННЯ ТА ОБЛАСТЬ ВИКОРИСТАННЯ UART (англ. universal asynchronous receiver/transmitter — універсальний асинхронний приймач/передавач) — тип асинхронного приймача-передавача, компонентів комп'ютерів та периферійних пристроїв, що передає дані між паралельною та послідовною формами. UART звичайно використовується спільно з іншими комунікаційними стандартами, такими як EIA RS-232.  UART – це зазвичай окрема мікросхема чи частина мікросхеми, що використовується для з'єднання через комп'ютерний чи периферійний послідовний порт. UART нині загалом включені в мікроконтролери. Здвоєний UART (Dual UART або DUART) об'єднує двоє UART в одній мікросхемі. Багато сучасних мікросхем сьогодні випускаються з можливістю комунікації в синхронному режимі, такі прилади називають USART. Зазвичай UART не отримує і не генерує зовнішні сигнали, які подорожують між різними частинами обладнання. Як правило, для перетворення логічного рівня UART в та з зовнішнього рівня сигналів використовується окремий інтерфейсний блок. (8) | | | | | | |
|  |  |  |  |  | АААА.123456.007 ПЗ | *Лист* |
|  |  |  |  |  |
|  |
| *Зм* | *Лист* | *№ докум.* | *Підп.* | *Дата* |
| ТЕХНІЧНА ХАРАКТЕРИСТИКА Швидкість передачі: = 4800 бод;  Тактова частота кварцового резонатора: = 8МГц;  Для виконання курсової роботи використано один з мікроконтролерів сімейства AVR – ATMEGA16A  ATmega16A має наступні властивості[3]:      Модуль УСАПП мікроконтролера має наступні властивості [4]:   * Можливість працювати в різних режимах: повний дуплекс, напівдуплекс, RX, TX. * Має синхронний та асинхронний режими | | | | | | |
|  |  |  |  |  | АААА.123456.007 ПЗ | *Лист* |
|  |  |  |  |  |
|  |
| *Зм* | *Лист* | *№ докум.* | *Підп.* | *Дата* |
| * Наявна можливість задавати швидкість передачі даних з високою роздільною здатністю * Може підтримувати кадри з 5, 6, 7, 8, або 9 інформаційними бітами та 1 або 2 стопових біта * Наявна апаратна перевірка на парність та перевірка «помилки кадру» * Виявлення перевищення (втрат) даних * Виявлення помилки синхронізації кадру * Фільтрація шуму, що включає виявлення «невірного старту» та цифровий фільтр низьких частот * Три джерела переривань:   + після закінчення прийому   + після закінчення передачі   + по спустошенню регістру даних | | | | | | |
|  |  |  |  |  | АААА.123456.007 ПЗ | *Лист* |
|  |  |  |  |  |
|  |
| *Зм* | *Лист* | *№ докум.* | *Підп.* | *Дата* |
| ОГЛЯД ІСНУЮЧИХ РІШЕНЬ ТА ОБГРУНТУВАННЯ ВИБОРУ СТРУКТУРИ ПРИСТРОЮ Очевидним є той факт, що УАПП сам по собі є лише пристроєм для прийому-передачі даних. З іншого боку, прийом-передача даних є надзвичайно важливою складовою багатьох проєктів, зокрема й тих, що використовуються в промисловості.  Прикладами пристроїв, що використовують УАПП можуть бути:   * <https://hackaday.io/projects?tag=UART> * <https://hackaday.io/projects?page=2&tag=UART> * <https://hackaday.io/projects?page=3&tag=UART>   Таким чином, неважко обрати приклад, схожий на завдання даної курсової (або, звичайно і очевидно, навіть більш складний).  Структура пристрою, наведена в курсовій роботі є доволі базовою, і, тим не менш, діючою для демонстрації основних принципів роботи з УАПП.  Таким чином, я вважаю, що вибір саме такої конфігурації пристроїв в моделі є обґрунтованим та підкріпленим попереднім досвідом застосувань УАПП; що, зокрема, прослідковується в кількості проєктів, наведених двома параграфами вище в «прикладах». | | | | | | |
|  |  |  |  |  | АААА.123456.007 ПЗ | *Лист* |
|  |  |  |  |  |
|  |
| *Зм* | *Лист* | *№ докум.* | *Підп.* | *Дата* |
| ОПИС РОБОТИ ПРИСТРОЮ ЗА СТРУКТУРНОЮ СХЕМОЮ На рисунку 5.1 наведено структурну схему пристрою обміну інформацією між модулем УСАПП мікроконтролера ATmega16A та віртуальним терміналом.  Основним вузлом структури є мікроконтролер, який керує процесом обміну інфорації.  За допомогою восьми кнопок кодується байт для передачі, тобто вхідний сигнал. Через паралельний порт введення мікроконтролер отримує цей сигнал.  Через порт виведення вхідний сигнал передається до світлодіодів. Через модуль УСАПП мікроконтролера байт, який закодовано кнопками, з виходу TxD1 передається на вхід RxD1 віртуального термінала у послідовному асинхронному старт-стопному режимі. Також вхідний сигнал відображається на екрані осцилографа.  З виходу TxD2 віртуального терміналу у послідовному асинхронному старт-стопному режимі на вхід RxD2 модуля УАПП подається ASCII-код символу, який вводиться з клавіатури. Цей код також відображається на екрані осцилографа.  Кварцовий резонатор (КР) визначає частоту високочастотного генератора, схему якого вбудовано у мікроконтролер.  Схема скидання (ССКИД) формує сигнал «RESET» (Скидання), який запускає виконання робочої програми спочатку. Цей сигнал формується автоматично при включенні живлення, або від зовнішньої кнопки КСКИД. | | | | | | |
|  |  |  |  |  | АААА.123456.007 ПЗ | *Лист* |
|  |  |  |  |  |
|  |
| *Зм* | *Лист* | *№ докум.* | *Підп.* | *Дата* |

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  | | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | |  |  |  |  |  | АААА.123456.007 ЕЗ | | | | | | | | | | |  |  |  |  |  | |  |  |  |  |  | |  |  |  |  |  | Пристрій обміну інформацією між модулем УАПП та віртуальним терміналом  Схема електрична структурна | | Літера | | | | Маса | | | Масштаб | | Зм. | Лист | № докум. | Підпис | Дата |  |  |  | |  | | |  | | Розроб. | |  |  |  | Т |  |  |  |  | |  |  | | Перевір. | |  |  |  |  |  |  | | Т.контр. | |  |  |  | Лист | | | | |  | Листів 1 | | |  | |  |  |  |  | НТУУ “КПІ” ФІОТ  Група АААА | | | | | | | | | | Н.контр. | |  |  |  | | Затверд. | |  |  |  |     К1. К2,…,К8 - кнопки  КР - кварцовий резонатор  ССКИД - схема скидання  КНСКИД - кнопка скидання  МК - мікроконтролер  ПВВ - порт введення  ПВИВ - порт виведення  УАПП - універсально асинхронний приймач-передавач  СД1,СД2,…,СД8 - світлодіоди  ОСЦ - осцилограф  ВТ - віртуальний термінал  ДЖ - джерело живлення  КЛ - клавіатура |

Рисунок 5.1 - Схема електрична структурна пристрою обміну інформацією між модулем УАПП мікроконтролера та віртуальним терміналом

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| ВИБІР ТА ОБГРУНТУВАННЯ ОКРЕМИХ ВУЗЛІВ ЕЛЕМЕНТІВМікроконтроллер ATmega16A – 8- розрядний мікроконтролер, що базується на посиленій AVR RISC архітектурі. Цей мікроконтролер забезпечує продуктивність 1 млн. операцій у секунду на 1 МГц синхронізації за рахунок виконання більшості інструкцій за один машинний цикл і дозволяє оптимізувати споживання енергії за рахунок зміни частоти синхронізації.    Рисунок 6.1  Цей мікроконтролер повністю задовольняє вимоги до курсової роботи, оскільки він простий в використанні, дешевий та досить потужний. | | | | | | |
|  |  |  |  |  | АААА.123456.007 ПЗ | *Лист* |
|  |  |  |  |  |
|  |
| *Зм* | *Лист* | *№ докум.* | *Підп.* | *Дата* |
| Рисунок 6.2 - Схема ATMEGA16A | | | | | | |
|  |  |  |  |  | АААА.123456.007 ПЗ | *Лист* |
|  |  |  |  |  |
|  |
| *Зм* | *Лист* | *№ докум.* | *Підп.* | *Дата* |
| Таблиця 6.1 - Порівняльна таблиця мікроконтролерів AVR   |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  | | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | | **Модель** | **Flash(Кб)** | **СППЗУ, кБ** | **ОЗУ (байт)** | **I/O** | **Fmax,Мгц** | **Vcc, В** | **Таймер 1б-біт** | **Таймер 8-біт** | **ШИМ, каналів** | **RTC** | **SPI** | **UART** | **TWI** | **ISP** | **10-біт АЦП, кан.** | **Компаратори (ан.)** | **Детектор Vc** | **WD** | **Osc.** | **Hardware Multiplier** | **Корпус** | | AT90S1200 | 1 | 0.0625 |  | 15 | 12 | 2.7-6.0 |  | 1 |  |  |  |  |  | + |  | + |  | + | + |  | PDIP20, S0IC20, SS0P20 | | AT90S2313 | 2 | 0.125 | 128 | 15 | 10 | 2.7-6.0 | 1 | 1 | 1 |  |  | 1 |  | + |  | + |  | + |  |  | PDIP20, S0IC20 | | ATmega128 | 128 | 4 | 4096 | 53 | 16 | 4.5-5.5 | 2 | 2 | 8 | + | 1 | 2 | + | + | 8 | + | + | + | + | + | TQFP64 | | ATmegal28L | 128 | 4 | 4096 | 53 | 8 | 2.7-5.5 | 2 | 2 | 8 | + | 1 | 2 | + | + | 8 | + | + | + | + | + | TQFP64 | | ATmega16 | 16 | 0.5 | 1024 | 32 | 16 | 4.5-5.5 | 1 | 2 | 3 | + | 1 | 1 | + | + | 8 | + | + | + | + | + | PDIP40.TQFP44, MLF44 | | ATmega162 | 16 | 0.5 | 1024 | 35 | 16 | 4.5-5.5 | 2 | 2 | 4 | + | 1 | 2 |  | + |  | + | + | + | + | + | PDIP40, TQFP44 | | ATmega162L | 16 | 0.5 | 1024 | 35 | 8 | 2.7-5.5 | 2 | 2 | 4 | + | 1 | 2 |  | + |  | + | + | + | + | + | PDIP40, TQFP44 | | ATmega162V | 16 | 0.5 | 1024 | 35 | 1 | 1.8-3.6 | 2 | 2 | 4 | + | 1 | 2 |  | + |  | + | + | + | + | + | PDIP40, TQFP44 | | ATmega165 | 16 | 0.5 | 1024 | 54 | 16 | 4.5-5.5 | 1 | 2 | 4 | + | 1+USI | 1 | + | + | 8 | + | + | + | + | + | TQFP64 | | ATmega165V | 16 | 0.5 | 1024 | 54 | 8 | 1.8-5.5 | 1 | 2 | 4 | + | 1+USI | 1 | + | + | 8 | + | + | + | + | + | TQFP64 | | ATmega168 | 16 | 0.5 | 1024 | 23 | 20 | 1.8-5.5 | 1 | 2 | 3 | + | 1+USART | 1 | + | + | 8 | + | + | + | + | + | PDIP28, TQFP32, MLF32 | | ATmega169 | 16 | 0.5 | 1024 | 54 | 16 | 4.5-5.5 | 1 | 2 | 4 | + | 1+USI | 1 | + | + | 8 | + | + | + | + | + | TQFP64 | | ATmegal69V | 16 | 0.5 | 1024 | 54 | 1 | 1.8-5.5 | 1 | 2 | 4 | + | 1+USI | 1 | + | + | 8 | + | + | + | + | + | TQFP64 | | ATmega16L | 16 | 0.5 | 1024 | 32 | 8 | 2.7-5.5 | 1 | 2 | 3 | + | 1 | 1 | + | + | 8 | + | + | + | + | + | PDIP40, TQFP44 | | ATmega32 | 32 | 1 | 2048 | 32 | 16 | 4.0-5.5 | 2 | 2 | 4 | + | 1 | 1 | + | + | 8 | + | + | + | + | + | PDIP40, TQFP44, MLF44 | | ATmega32L | 32 | 1 | 2048 | 32 | 8 | 2.7-5.5 | 1 | 2 | 4 | + | 1 | 1 | + | + | 8 | + | + | + | + | + | PDIP40, TQFP44, MLF44 | | ATmega48 | 4 | 0.256 | 512 | 23 | 20 | 1.8-5.5 | 1 | 2 | 3 | + | 1+USART | 1 | + | + | 8 | + | + | + | + | + | PDIP28, TQFP32, MLF32 | | ATmega64 | 64 | 2 | 4096 | 53 | 16 | 4.5-5.5 | 2 | 2 | 8 | + | 1 | 2 | + | + | 8 | + | + | + | + | + | TQFP64 | | ATmega64L | 64 | 2 | 4096 | 53 | 8 | 2.7-5.5 | 2 | 2 | 8 | + | 1 | 2 | + | + | 8 | + | + | + | + | + | TQFP64 | | ATmega8 | 8 | 0.5 | 1024 | 23 | 16 | 4.5-5.5 | 1 | 2 | 3 | + | 1 | 1 | + | + | 8 | + | + | + | + | + | PDIP28, TQFP32, MLF32 | | ATmega16A | 8 | 0.5 | 512 | 35 | 16 | 4.5-5.5 | 1 | 1 | 3 |  | 1 | 1 |  | + |  |  | + | + | + | + | PDIP40, TQFP44 | | ATmega16AL | 8 | 0.5 | 512 | 35 | 8 | 2.7-5.5 | 1 | 1 | 3 | + | 1 | 1 |  | + |  |  | + | + | + | + | PDIP40, TQFP44 | | ATmega8535 | 8 | 0.5 | 512 | 32 | 16 | 4.5-5.5 | 1 | 2 | 4 |  | 1 | 1 | + | + | 8 | + | + | + | + | + | PDIP40, TQFP44 | | ATmega8535L | 8 | 0.5 | 512 | 32 | 8 | 2.7-5.5 | 1 | 2 | 4 | + | 1 | 1 | + | + | 8 | + | + | + | + | + | PDIP40, TQFP44 | | ATmega88 | 8 | 0.5 | 1024 | 23 | 20 | 1.8-5.5 | 1 | 2 | 3 | + | 1+USART | 1 | + | + | 8 | + | + | + | + | + | PDIP28, TQFP32, MLF32 | | ATmega8L | 8 | 0.5 | 1024 | 23 | 8 | 2.7-5.5 | ! | 2 | 3 | + | 1 | 1 | + | + | 8 | + | + | + | + | + | PDIP28,TQFP32, MLF32 | | ATtiny11 | 1 |  |  | 6 | 6 | 2.7-5.5 |  | 1 |  |  |  |  |  |  |  | + |  | + | + |  | PDIP8, S0IC8 | | ATtiny12 | 1 | 0.0625 |  | 6 | 8 | 1.8-5.5 |  | 1 |  |  |  |  |  | + |  | + | + | + | + |  | PDIP8, S0IC8 | | ATtiny13 | 1 | 0.064 | 64 | 6 | 20 | 1.8-5.5 |  | 1 | 2 |  |  |  |  | + | 4 | + | + | + | + |  | PDIP8, S0IC8 | | ATtinyl5L | 1 | 0.0625 |  | 6 | 1,6 | 2.7-5.5 |  | 2 | 1 |  |  |  |  | + | 4 | + | + | + | + |  | PDIP8, S0IC8 | | ATtiny2313 | 2 | 0.128 | 128 | 18 | 20 | 1.8-5.5 | 1 | 1 | 4 |  | USI | 1 |  | + |  | + | + | + | + |  | PDIP20, SOIC20 | | ATtiny26 | 2 | 0.125 | 128 | 16 | 16 | 4.5-5.5 |  | 2 | 2 |  | USI |  |  | + | 11 | + | + | + | + |  | PDIP20, S0IC20, MLF32 | | ATtiny26L | 2 | 0.125 | 128 | 16 | 8 | 2.7-5.5 |  | 2 | 2 |  | USI |  |  | + | 11 | + | + | + | + |  | PDIP20, S0IC20, MLF32 | | ATtiny28L | 2 |  | 32 | 11 | 4 | 2.7-5.5 |  | 1 |  |  |  |  |  |  |  | + |  | + | + |  | PDIP28, TQFP32, MLF32 | | ATtiny28V | 2 |  | 32 | 11 | 1 | 1.8-5.5 |  | 1 |  |  |  |  |  |  |  | + |  | + | + |  | PDIP28, TQFP32, MLF32 | | | | | | | |
|  |  |  |  |  | АААА.123456.007 ПЗ | *Лист* |
|  |  |  |  |  |
|  |
| *Зм* | *Лист* | *№ докум.* | *Підп.* | *Дата* |
| Інтерфейс УСАПП мікроконтролера ATMEGA16A Інтерфейс УСАПП - складова мікроконтролера ATMEGA16A. Схема підключення модуля УСАПП зображена на рисунку 6.3.  УАПП широко застосовувався раніше, але з початку 2000х його почали заміняти інші протоколи, наприклад, USB. Тим не менш, в широкому доступі існують, наприклад, перехідники USB/USART, тобто вибір протокола УАПП в цій роботі не завадить демонстрації основних принципів моделі.  Окрім USB, існують, наприклад, й інші протоколи передачі даних, як-от SPI та I2C.  USB-UART / USB-TTL конвертер на чипе CP2102  Рисунок 6.3 - перехідник USB/USART  UART  Рисунок 6.4 - схема підключення УАПП | | | | | | |
|  |  |  |  |  | АААА.123456.007 ПЗ | *Лист* |
|  |  |  |  |  |
|  |
| *Зм* | *Лист* | *№ докум.* | *Підп.* | *Дата* |
| ПРОЕКТУВАННЯ ПРИСТРОЮРозробка та опис робочої моделі в пакеті Proteus Схема робочої моделі зображена на рисунку 7.1. Зліва знаходяться вісім кнопок, за допомогою яких задається байт даних, який передається через модуль УСАПП  Кнопки підключено до восьми ліній порту А мікроконтролера: PА.0, PА.1, … , PА.7. Якщо якась із кнопок відпущена, то через резистори R1...R8 на відповідну лінію порту А подається логічна 1. Якщо кнопка нажата, то вводиться логічний нуль. Праворуч зображено вісім світлодіодів, які підключено до ліній порту С: PС.0, PС.1, … , PС.7. Катоди світлодіодів підключено до спільного провода (землі), а на аноди через резистори R9…R16, які обмежують струм, із виходів порту С подаються логічні одиниці, коли треба засвітити відповідний світлодіод.  До ліній TxD – вихід передавача УСАПП та RxD – вхід приймача підключено осцилограф, та віртуальний термінал «Virtual Terminal». На ньому відображаються повідомлення, які вводиться та виводяться через модуль УСАПП в/з мікроконтролера.  До виводів XTAL1 та XTAL2 в реальних схемах підключають кварцовий резонатор частотою, яка визначає тактову частоту генератора мікроконтролера Також підключають конденсатори C1 та C2, ємністю 30пФ, які призначені для підвищення стабільності роботи системного генератора. Резонатор та конденсатори потрібні лише в практичній схемі.  В якості мікроконтролера в моделі використано AVR-мікроконтролер ATmega16A. | | | | | | |
|  |  |  |  |  | АААА.123456.007 ПЗ | *Лист* |
|  |  |  |  |  |
|  |
| *Зм* | *Лист* | *№ докум.* | *Підп.* | *Дата* |
| Рисунок 7.1 - Схема робочої моделі Розрахунок модуля УСАПП Розрахунок значення регістра UBRR  При роботі в асинхронному режимі швидкість обміну визначається не тільки вмістом регістра UBRR, але й станом розряду U2X (U2Xn) регістра UCSRA. Якщо цей розряд встановлено в «1», коефіцієнт ділення попереднього дільника зменшується у два рази, а швидкість обміну відповідно подвоюється. При роботі у синхронному режимі цей розряд має бути скинуто.  Отже, швидкість обміну визначається наступними формулами :   * асинхронний режим (звичайний, U2Xn= «0»):   BAUD= /(16(UBRR+1)); | | | | | | |
|  |  |  |  |  | АААА.123456.007 ПЗ | *Лист* |
|  |  |  |  |  |
|  |
| *Зм* | *Лист* | *№ докум.* | *Підп.* | *Дата* |
| * асинхронний режим (пришвидшений, U2Xn= «1»):   BAUD= /(8(UBRR+1))  BAUD – швидкість передачі у бодах, тактова частота мікроконтролера, UBRR- вміст регістра контролера швидкості передачі (0…4095).  Згідно з завданням до курсової роботи, швидкість передачі становить 4800 бод, тактова частота кварцового резонатора – 8МГц. При таких даних UBRR=103 при U2X = 0, або UBRR=207 при U2X = 1. Похибка відносно стандартної швидкості в першому випадку становить 0.2%, в другому – 0.2%.  Обираємо UBRR=207=0xCF при U2X = 1.  Програмування регістра контролера швидкості передачі UBRR  Для нашого мікроконтролера регістр UBRR є 12-розрядним та фізично розміщений у двох регістрах введення/виведення. Ці регістри мають адреси: UBRRН=$0040; UBRRL=$0029.  Згідно з розрахунком, наведеним вище, UBRR=207= $00CF  Тоді програма має вид:  LDI R18, $00; R18 $00;  LDI R31, $00; R31$00;  LDI R30, $40; R30$40; R31, R30 (Z)$0040 (адреса UBRRН);  ST Z, R18; UBRRH $00 (старший байт UBRR).  LDI R19, $03; R19 $CF;  LDI R27, $00; R27$00;  LDI R26, $29; R26$29; R27, R26 (X)$0029 (адреса UBRR0L);  ST X, R19; UBRR0L = $CF (молодший байт UBRR). | | | | | | |
|  |  |  |  |  | АААА.123456.007 ПЗ | *Лист* |
|  |  |  |  |  |
|  |
| *Зм* | *Лист* | *№ докум.* | *Підп.* | *Дата* |
| Програмування регістра UCSRA  При програмуванні регістра потрібно встановити біт U2X в 0 та встановити біт UDRE в одиницю. Після скидання: біт UDRE автоматично встановлюється у одиницю, а біт U2X скидається в нуль. Інші біти регістра мають початкові нульові значення. Тоді керуюче слово КС1 має вид:  7р 6р 5р 4р 3р 2р 1р 0р   |  |  |  |  |  |  |  |  | | --- | --- | --- | --- | --- | --- | --- | --- | | 0 | 0 | 0 | 0 | 0 | 0 | 1 | 0 |   UCSRA B=$02  Регістр UCSRA має адресу у просторі резидентної пам’яті даних мікроконтролера АТ mega8515: $002В.  Тоді програма має вигляд:  LDI R16, $22; R16KC1=$02,  LDI R27, $00; R27$00,  LDI R26, $2В; R26$2В; R27, R26 (X) $002В,  ST X, R16; UCSR0A R16= КС1=$02  Програмування регістра UCSRC  Регістр UCSRC - регістр даних УСАПП. Для мікроконтролера ATmega16A розряд UМSEL регістру програмує режим роботи модуля USАRТ. Оскільки у завданні задано асинхронний режим, то цей біт повинен мати нульове значення.  Біти UPM1 та UPM0 керують контролем парності. В нашому випадку необхідно запрограмувати: UPM1=1; UPM0=0 (перевірка на непарність).  Біт USBS повинен дорівнювати 1, тому що ми повинні передавати 2 стоп-біти.  Біти UCSZ1, UCSZ0 разом з бітом UCSZ2 регістра UCSRВ програмують розмір | | | | | | |
|  |  |  |  |  | АААА.123456.007 ПЗ | *Лист* |
|  |  |  |  |  |
|  |
| *Зм* | *Лист* | *№ докум.* | *Підп.* | *Дата* |
| слова даних. Для нашого прикладу треба запрограмувати: UCSZ1=1; UCSZ0=0, а UCSZ2=0 (передається 7 бітів даних).  Розряд UCPOL в асинхронному режимі не використовуються, тому запишемо у нього 0. Біт URSEL= 1, тому, що запис виконується в регістр UCSRC.  Тоді керуюче слово КС2 має вид:  7р 6р 5р 4р 3р 2р 1р 0р   |  |  |  |  |  |  |  |  | | --- | --- | --- | --- | --- | --- | --- | --- | | 1 | 0 | 1 | 0 | 1 | 1 | 0 | 0 |   UCSRC 0xAC  Регістр UCSRC для заданого МК-ра має адресу: $0040.  Тоді програма має вид:  LDI R17, $AC; R17 $AC,  LDI R29, $00; R29$00,  LDI R28, $40; R28$40, R29, R28 (Y) $0040,  ST Y , R17; UCSR0С R17 = КС2 = $AC.  Програмування регістра UCSRB  UCSRB - регістр статусу УСАПП, відображає стан модуля УСАПП  Необхідно запрограмувати: TXCIE=1 (дозвіл на переривання «передачу завершено»); UDRIE=1 (дозвіл на переривання «регістр даних пустий».); UCSZ02=0; TXEN=1 (дозвіл роботи передавача UART/USART); RXEN=1. Інші біти не використовуються, тому пишемо в них нулі.  Тоді керуюче слово КС3 має вид:  7р 6р 5р 4р 3р 2р 1р 0р   |  |  |  |  |  |  |  |  | | --- | --- | --- | --- | --- | --- | --- | --- | | 0 | 1 | 1 | 1 | 1 | 0 | 0 | 0 |   UCSRВ 0x78  Регістр UCSRB має адресу: $002А. | | | | | | |
|  |  |  |  |  | АААА.123456.007 ПЗ | *Лист* |
|  |  |  |  |  |
|  |
| *Зм* | *Лист* | *№ докум.* | *Підп.* | *Дата* |
| Тоді програма має вид:  LDI R19, $78; R19← КС3=$78,  LDI R29, $00; R29←$00,  LDI R28, $2А; R28←$2А R29, R28 (Y)←$002А,  ST Y, R19; UCSRВ ←R19 = КС3= $78.  Програмування регістра даних UDR  Регістр даних UDR нашою МК-ра має адресу $002C.  Тоді програма має вид:  LDI R22, $E4; R22← $E4;( $E4 = 11100100b)  LDI R31, $00; R31 $00;  LDI R30, $2С; R30$2С; R31, R30 (Z) $002C;  ST Z, R22; UDR0 R22 (завантаження байту для передачі).  Після завантаження байта з регістра R22 в регістр UDR (буфер передавача) інформація у послідовному двійковому коді у старт-стопному форматі передається на лінію TXD = PD1 мікроконтролера – АТmega16A. | | | | | | |
|  |  |  |  |  | АААА.123456.007 ПЗ | *Лист* |
|  |  |  |  |  |
|  |
| *Зм* | *Лист* | *№ докум.* | *Підп.* | *Дата* |
| РОЗРОБКА І ОПИС СХЕМИ АЛГОРИТМУ ТА КЕРУЮЧОЇ ПРОГРАМИСхема алгоритму роботи пристрою Схему алгоритму роботи пристрою наведено на рисунку 8.1    Рисунок 8.1 - Схема алгоритму роботи пристрою | | | | | | |
|  |  |  |  |  | АААА.123456.007 ПЗ | *Лист* |
|  |  |  |  |  |
|  |
| *Зм* | *Лист* | *№ докум.* | *Підп.* | *Дата* |
| Керуюча програма мовою Асемблер .nolist ; директива відключає генерацію коду у лістинг,  ;тобто далі у файлі \*.lss не буде фіксуватися асемблерний код  .include "m8515def.inc" ;підключення стандартного заголовочного файлу для ATmega8  .list ;директива включає генерацію коду у лістинг, тобто далі у файлі \*.lss буде фіксуватися асемблерний код  .equ fCK = 8000000 ;clock frequency in Hz  .equ BAUD = 4800 ;UART baudrate  .equ UBRR\_value = (fCK / (BAUD \* 16)) - 1 ; розрахунок значення для регістра UBRR  .cseg ;директива визначення, що далі іде код програми  .org 0 ;директива визначення, що код програми у FLASH буде розміщено з нульової адреси  main:  rcall init\_USART ; (блок 1) відносний виклик підпрограми    ;set port C as OUT  LDI R16, 0xFF ; (блок 2) R16 <-0xFF  OUT DDRC, R16 ; (блок 2) DDRC <- R16 | | | | | | |
|  |  |  |  |  | АААА.123456.007 ПЗ | *Лист* |
|  |  |  |  |  |
|  |
| *Зм* | *Лист* | *№ докум.* | *Підп.* | *Дата* |
| rcall USART\_receive ; receive from UART [to R16]    ;OUT the received data from port C (LEDs connected)  OUT PORTC, R16 ; (блок 4) PORTC <-R16    ;set port A as IN  LDI R16, 0x00 ; (блок 5) R16<-0x00  OUT DDRA, R16 ; (блок 5) DDRА<- R16      ; read buttons into A  in R16, PINA ; (блок 6) R16<- PINA  ; out read to C (LEDs)  OUT PORTC, R16 ; (блок 7) PORTC<-R16  rcall USART\_send ; send read data (R16) by UART  init\_USART:  ldi R16, high(UBRR\_value) ; (блок 1) R16 <-старший байт UBRR\_value  out UBRRH, R16 ; (блок 1) UBRRH <-R16  ldi R16, low(UBRR\_value) ; (блок 1) R16 <-молодший байт UBRR\_value  out UBRRL, R16 ; (блок 1) UBRRL <-R16  ldi R16, (1<<RXEN)|(1<<TXEN)|(0<<RXCIE)|(1<<TXCIE)|(1<<UDRIE)  ; RXEN - UART Receiver enable  ; TXEN - UART Transmitter enable  ; RXCIE - UART RX Complete Interrupt Enable (0 - disabled)  ; TXCIE - UART TX Complete Interrupt Enable  ; UDRIE - UART Data Register Empty Interrupt Enable  out UCSRB, R16 ; write settings above to their corresponding UCSRB register | | | | | | |
|  |  |  |  |  | АААА.123456.007 ПЗ | *Лист* |
|  |  |  |  |  |
|  |
| *Зм* | *Лист* | *№ докум.* | *Підп.* | *Дата* |
| ; 7 бітів даних, перевірка на непарність, 2 стоп-біти  ldi R16, (1<< URSEL)|(1<<UPM1)|(1<<UPM0)|(1<< UCSZ1)|(0<< UCSZ0)|(1<<USBS)  ; URSEL 1 means we write to UCSRC (not UBRRH)  ; UPM1,1 set to 1,0 - even parity (outputs 1 if ODD | 0 if EVEN)  ; UCSZ2,1,0 set to 0,1,0 - 7 data bits  ; USBS set to 1 - 2 stop bits  out UCSRC, R16 ; (блок 1) UCSRC <- R16    ret ; return [to main]  ; передача через модуль USART  USART\_send:  out UDR, R16 ; populate UDR (UART Data register) with data from R16    sending: ;loop while TXC (USART Transmit Complete Flag) != 1 (1 would mean send is complete)  sbis UCSRA, TXC ; (блок 8) якщо біт TXC = 1, то наступна ;команда пропускається, інакше – наступна команда  rjmp sending ; loop  ret ; return  USART\_receive:  sbis UCSRA, RXC ; loop while receiving  rjmp USART\_receive ; loop  in R16, UDR ; populate R16 with read data  ret ; return | | | | | | |
|  |  |  |  |  | АААА.123456.007 ПЗ | *Лист* |
|  |  |  |  |  |
|  |
| *Зм* | *Лист* | *№ докум.* | *Підп.* | *Дата* |
| МОДЕЛЮВАННЯ ПРИСТРОЮ В ПАКЕТІ PROTEUS Перш за все, потрібно пересвідчитись, що Terminal налаштовано правильно.  Налаштування віртуального терміналу зображені на рисунку 9.1 і відповідають завданню курсової роботи (швидкість передачі – 4800 бод, 7 бітів даних, 2 стоп-біти, перевірка на непарність (ODD parity))    Рисунок 9.1 - Налаштування віртуального термінала  Натисканням Start VSM Debugging запускається процес моделювання У вікні Virtual Terminal натисканням правої кнопки обираються: Hex Display Mode та Echo Typed Characters (рисунок 9.2).  У вкладці Source Code встановлюємо дві точки зупинки( рисунок 9.3)  Запускаємо процес симуляції. | | | | | | |
|  |  |  |  |  | АААА.123456.007 ПЗ | *Лист* |
|  |  |  |  |  |
|  |
| *Зм* | *Лист* | *№ докум.* | *Підп.* | *Дата* |
| Рисунок 9.2 - Налаштування віртуального термінала    Рисунок 9.3 - Встановлення двох точок зупинки | | | | | | |
|  |  |  |  |  | АААА.123456.007 ПЗ | *Лист* |
|  |  |  |  |  |
|  |
| *Зм* | *Лист* | *№ докум.* | *Підп.* | *Дата* |
| Вводиться символ, який надсилається від віртуального терміналу до мікроконтролера через УАПП. При введенні символу у вікні терміналу повинен стояти курсор. Символ, який вводиться з клавіатури, віртуальний термінал перетворює в ASCII-код. Наприклад, вводиться символ n, ASCII код якоro 0x6E = 0b01101110. Програма перейде на рядок, де встановлено першу точку зупину.  У вкладці зі схемою Schematic Capture на світлодіодах та осцилографі відображається символ, який прийнято від термінала через виходи порту С (рисунок 9.4).  Світлодіоди та осцилограф відображають бінарний код символу, який було прийнято від віртуального терміналу.    Рисунок 9.4 - Встановлення переданого від терміналу символу на світлодіодах та осцилографі | | | | | | |
|  |  |  |  |  | АААА.123456.007 ПЗ | *Лист* |
|  |  |  |  |  |
|  |
| *Зм* | *Лист* | *№ докум.* | *Підп.* | *Дата* |
| Символ для передачі мікроконтролером через УСАПП кодується кнопками: замкнена кнопка – нуль, а розімкнена – одиниця (рисунок 9.5). Якщо дивитися на кнопки знизу доверху, то на рисунку 9.5 закодовано число: 0b01101010 = 0x6A .    Рисунок 9.5 - Кодування кнопками символу для передачі  Щоб побачити переданий сигнал на осцилографі, потрібно натиснути кнопку на опції One-Shot (один кадр)( Рисунок 9.6)    Рисунок 9.6 - Налаштування осцилографа | | | | | | |
|  |  |  |  |  | АААА.123456.007 ПЗ | *Лист* |
|  |  |  |  |  |
|  |
| *Зм* | *Лист* | *№ докум.* | *Підп.* | *Дата* |
| Для продовження виконання програми знову натискається Run Simulation. Програма переходить до другої точки зупину. Отримуємо дані на осцилографі. ( рисунок 9.7)    Рисунок 9.7 - Відображення сигналів на осцилографі: від терміналу – червоний; від МК (кнопок) – зелений  Осцилограф відображає сигнали за двома променями: нижній (червоний) – сигнал, який приймається від терміналу, верхній (зелений) – сигнал, який передається від кнопок.  Коментуючи червоний сигнал (рисунок 9.8), можна сказати, що перший перепад сигналу із високого рівня у низький – старт-біт. Потім знаходяться сім інформаційних бітів, які відображені на віртуальному терміналі та осцилографі, як 0b1101110 = 0x6E  Далі іде 8-й біт доповнення до непарності Podd, який дорівнює 1 та два стоп-біти. | | | | | | |
|  |  |  |  |  | АААА.123456.007 ПЗ | *Лист* |
|  |  |  |  |  |
|  |
| *Зм* | *Лист* | *№ докум.* | *Підп.* | *Дата* |
| Коментуючи зелений сигнал (рисунок 9.9), можна сказати, що перший перепад сигналу із високого рівня у низький – старт-біт. Потім сім інформаційних бітів, які відображені на віртуальному терміналі та осцилографі, як 0b1101010 - сигнал, прийнятий від кнопок. Далі йде Podd, який дорівнює 0 та два стоп-біти.    Рисунок 9.8 - Пояснення до сигналу    Рисунок 9.9 - Пояснення до сигналу | | | | | | |
|  |  |  |  |  | АААА.123456.007 ПЗ | *Лист* |
|  |  |  |  |  |
|  |
| *Зм* | *Лист* | *№ докум.* | *Підп.* | *Дата* |
| Для визначення отриманої швидкості передачі, яку в даному прикладі було обрано – 4800 біт/сек = 4800 бод, налаштовується така розгортка сигналу на осцилографі, щоб сумістити бічні сторони імпульсу із вертикальними лініями сітки осцилографа (рисунок 9.10).    Рисунок 9.10 - Визначення отриманої швидкості передачі на осцилографі  Як видно з рисунку, тривалість одного імпульсу дорівнює приблизно 0.22ms. Якщо розділити один біт на цей час, та помножити на 1000, отримується задана швидкість:  V = . | | | | | | |
|  |  |  |  |  | АААА.123456.007 ПЗ | *Лист* |
|  |  |  |  |  |
|  |
| *Зм* | *Лист* | *№ докум.* | *Підп.* | *Дата* |
| ВИСНОВОК В даній курсовій роботі я успішно модифікував надану схему взаємодії модуля УАПП мікроконтролера з віртуальним УАПП.  Параметри майбутнього виробу, налаштовані мною в середовищі моделювання, повністю збігаються з наданими в технічному завданні, якому відповідає мій варіант курсової роботи.  Стосовно рекомендацій щодо використання та впровадження  запропонованої моделі УАПП у промисловості, зокрема на  потужностях ДП «Укроборонпром», я б рекомендував переглянути схему взаємодії повністю, оскільки вважаю, що механізм вводу та відображення отриманих даних є суто навчальним та громіздким для застосування на практиці. Техніко-економічна ефективність моделі повністю збігається з техніко-економічною ефективністю звичайного модуля УАПП та техніко-економічною ефективністю будь-якого середньостатистичного мікроконтролера AVR. | | | | | | |
|  |  |  |  |  | АААА.123456.007 ПЗ | *Лист* |
|  |  |  |  |  |
|  |
| *Зм* | *Лист* | *№ докум.* | *Підп.* | *Дата* |
| СПИСОК ЛІТЕРАТУРИ  1. UART - <https://maxembedded.com/2013/09/the-usart-of-the-avr/#UBRR> 2. ATmega16A Datasheet - <https://ww1.microchip.com/downloads/en/DeviceDoc/Atmel-8154-8-bit-AVR-ATmega16A_Datasheet%20Summary.pdf> 3. ASCII table - <https://www.google.com/search?q=ascii+table&sxsrf=ALiCzsZXLKOVabxci1CUTiFZchnnCC4LPg:1655026273863&source=lnms&tbm=isch&sa=X&ved=2ahUKEwjB1Zapzaf4AhWFyoKHTMrCpIQ_AUoAXoECAEQAw&biw=1920&bih=1049&dpr=1#imgrc=5Uw4K1lHaT_sHM> 4. AVR Instruction Set - <https://ww1.microchip.com/downloads/en/DeviceDoc/AVR-Instruction-Set-Manual-DS40002198A.pdf> 5. <https://www.microchip.com/en-us/product/ATmega16A#document-table> 6. Модуль SPI - <https://uk.wikipedia.org/wiki/Serial_Peripheral_Interface> 7. Модуль I2C - <https://uk.wikipedia.org/wiki/I%C2%B2C> 8. Модуль UART - <https://en.wikipedia.org/wiki/Universal_asynchronous_receiver-transmitter> 9. Вступ - https://elprivod.nmu.org.ua/ua/interesting/what\_is\_mp\_mc\_plc.php | | | | | | |
|  |  |  |  |  |  | *Лист* |
|  |  |  |  |  |
|  |
| *Зм* | *Лист* | *№ докум.* | *Підп.* | *Дата* |