

МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ
НАЦІОНАЛЬНИЙ ТЕХНІЧНИЙ УНІВЕРСИТЕТ УКРАЇНИ
„КИЇВСЬКИЙ ПОЛІТЕХНІЧНИЙ ІНСТИТУТ” ім. ІГОРЯ СІКОРСЬКОГО”
КАФЕДРА ІНФОРМАЦІЙНИХ СИСТЕМ І ТЕХНОЛОГІЙ

Пристрій обміну інформацією
AAAA.123456.007
Курсова робота

З дисципліни «МІКРОПРОЦЕСОРНІ СИСТЕМИ»

Керівник доц. Новацький А.О.
„Допущений до захисту”

(Особистий підпис керівника)

« ____ » _____ 2022 р.

Захищений з оцінкою

(оцінка)
« ____ » _____ 2022 р.

Члени комісії :

(Особистий підпис)

(Особистий підпис)

Виконавець

ст. Тельє І.Н.

зал. книжка № AAAA12

(Особистий підпис виконавця)

« ____ » _____ 2022р.

(Розшифровка підпису)

(Розшифровка підпису)

НАЦІОНАЛЬНИЙ ТЕХНІЧНИЙ УНІВЕРСИТЕТ УКРАЇНИ “КПІ” ім. Ігоря Сікорського

Кафедра ІНФОРМАЦІЙНИХ СИСТЕМ ТА ТЕХНОЛОГІЙ

Дисципліна Мікропроцесорні системи

Курс 2 Група AAAA Семестр 4

ЗАВДАННЯ

на курсову роботу студента

Тельє Ніка Інновича

(прізвище, ім'я, по батькові)

1.Тема проекту (роботи) Пристрій обміну інформацією між модулем УАПІ 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

[illegible]

Студент

(підпис)

Керівник

(niɔnuɔ)

« 15 » 02 2022 p.

Анатолій НОВАЦЬКИЙ
(ім'я та прізвище)

Опис

Номер рядка	Формат	Позначення	Найменування	Кіл. листів	Номер екзем.	Примітка	
1	A4	AAAA.123456.007 ПЗ	Пояснювальна записка	29			
2	A4	AAAA.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							
				AAAA.123456.007 ОП			
	Лист	№ докум.	Підпис				Дата
Розроб.	Тельє						
Перевір.	Новацький				Пристрій обміну інформацією Опис		
Реценз.							
Н.контр.							
Затверд.							
					Літ.	Лист	Листів
					Т	1	1
					НТУУ “КПІ” ФІОТ		
					Група AAAA		

Зміст

1	ВСТУП.....	6
2	ПРИЗНАЧЕННЯ ТА ОБЛАСТЬ ВИКОРИСТАННЯ.....	7
3	ТЕХНІЧНА ХАРАКТЕРИСТИКА.....	8
4	ОГЛЯД ІСНУЮЧИХ РІШЕНЬ ТА ОБГРУНТУВАННЯ ВИБОРУ СТРУКТУРИ ПРИСТРОЮ 10	
5	ОПИС РОБОТИ ПРИСТРОЮ ЗА СТРУКТУРНОЮ СХЕМОЮ	11
6	ВИБІР ТА ОБГРУНТУВАННЯ ОКРЕМИХ ВУЗЛІВ ЕЛЕМЕНТІВ	13
6.1	Мікроконтроллер	13
6.2	Інтерфейс УСАПП мікроконтролера АТМЕГА16А	16
7	ПРОЕКТУВАННЯ ПРИСТРОЮ	17
7.1	Розробка та опис робочої моделі в пакеті Proteus	17
7.2	Розрахунок модуля УСАПП	18
8	РОЗРОБКА І ОПИС СХЕМИ АЛГОРИТМУ ТА КЕРУЮЧОЇ ПРОГРАМИ	23
8.1	Схема алгоритму роботи пристрою	23
8.2	Керуюча програма мовою Асемблер	24
9	МОДЕЛЮВАННЯ ПРИСТРОЮ В ПАКЕТІ PROTEUS	27
10	ВИСНОВОК	34
11	СПИСОК ЛІТЕРАТУРИ	35

					AAAA.123456.007 ПЗ						
Зм.	Лист	№ докум.	Підп.	Дата							
Розроб.	Тельс				Пристрій обміну інформацією Пояснювальна записка			Літ.	Лист	Листів	
Перевір.	Новацький							Т		І	29
Н. контр.											
Затв.											
					НТУУ(КП) ФІОТ			група AAAA			

1 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 проекти.

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)

					AAAA.123456.007 ПЗ	Лист
Зм	Лист	№ докум.	Підп.	Дата		1.

2 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)

					AAAA.123456.007 ПЗ	Лист
Зм	Лист	№ докум.	Підп.	Дата		2.

3 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]:

- High Endurance Non-volatile Memory segments
 - 16KBytes of In-System Self-programmable Flash program memory
 - 512Bytes EEPROM
 - 1KByte Internal SRAM
 - Write/Erase Cycles: 10,000 Flash/100,000 EEPROM
 - Data retention: 20 years at 85°C/100 years at 25°C⁽¹⁾
 - Optional Boot Code Section with Independent Lock Bits
 - In-System Programming by On-chip Boot Program
 - True Read-While-Write Operation
 - Programming Lock for Software Security
- Peripheral Features
 - Two 8-bit Timer/Counters with Separate Prescalers and Compare Modes
 - One 16-bit Timer/Counter with Separate Prescaler, Compare Mode, and Capture Mode
 - Real Time Counter with Separate Oscillator
 - Four PWM Channels
 - 8-channel, 10-bit ADC
 - 8 Single-ended Channels
 - 7 Differential Channels in TQFP Package Only
 - 2 Differential Channels with Programmable Gain at 1x, 10x, or 200x
 - Byte-oriented Two-wire Serial Interface
 - Programmable Serial USART
 - Master/Slave SPI Serial Interface
 - Programmable Watchdog Timer with Separate On-chip Oscillator
 - On-chip Analog Comparator

Модуль УСАПП мікроконтролера має наступні властивості [4]:

- Можливість працювати в різних режимах: повний дуплекс, напівдуплекс, RX, TX.
- Має синхронний та асинхронний режими

					AAAA.123456.007 ПЗ	Лист
Зм	Лист	№ докум.	Підп.	Дата		3.

- 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

					AAAA.123456.007 ПЗ	Лист
Зм	Лист	№ докум.	Підп.	Дата		4.

4 ОГЛЯД ІСНУЮЧИХ РІШЕНЬ ТА ОБГРУНТУВАННЯ ВИБОРУ СТРУКТУРИ ПРИБРОЮ

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 «прикладях».

					AAAA.123456.007 ПЗ	Лист
Зм	Лист	№ докум.	Підп.	Дата		5.

5 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 інформації.

За допомогою восьми кнопок кодується байт для передачі, тобто вхідний сигнал. Через паралельний порт введення мікроконтролер отримує цей сигнал.

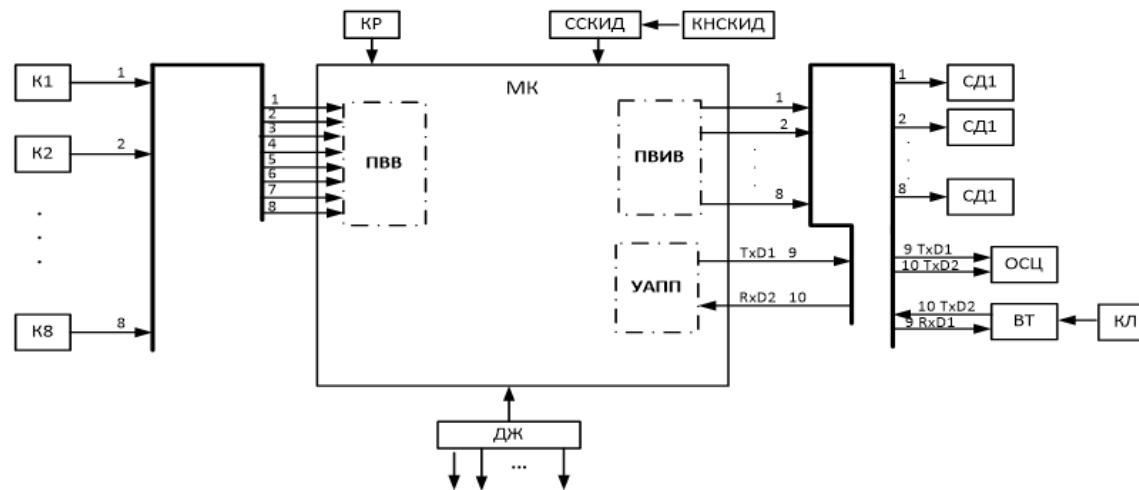
Через порт виведення вхідний сигнал передається до світлодіодів. Через модуль УСАПП мікроконтролера байт, який закодовано кнопками, з виходу TxD1 передається на вхід RxD1 віртуального термінала у послідовному асинхронному старт-стопному режимі. Також вхідний сигнал відображається на екрані осцилографа.

З виходу TxD2 віртуального термінала 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.

					AAAA.123456.007 ПЗ	Лист
Зм	Лист	№ докум.	Підп.	Дата		6.



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

					AAAA.123456.007 ЕЗ					
Зм.	Лист	№ докум.	Підпис	Дата	Пристрій обміну інформацією між модулем УАПП та віртуальним терміналом Схема електрична структурна	Літера		Маса	Масштаб	
Розроб.						T				
Перевір.										
Т.контр.						Лист	Листів 1			
						НТУУ “КПІ” ФІОТ Група AAAA				
Н.контр.										
Затверд.										

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

6 ВИБІР ТА ОБГРУНТУВАННЯ ОКРЕМИХ ВУЗЛІВ ЕЛЕМЕНТІВ

6.1 Мікроконтролер

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.

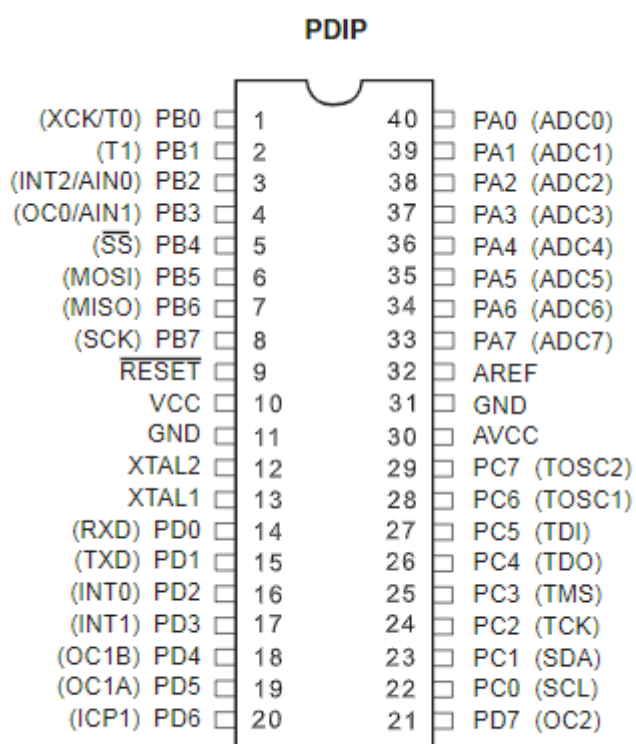


Рисунок 6.1

Цей мікроконтролер повністю задовольняє вимоги до курсової роботи, оскільки він простий в використанні, дешевий та досить потужний.

Зм	Лист	№ докум.	Підп.	Дата

AAAA.123456.007 ПЗ

Лист

7.

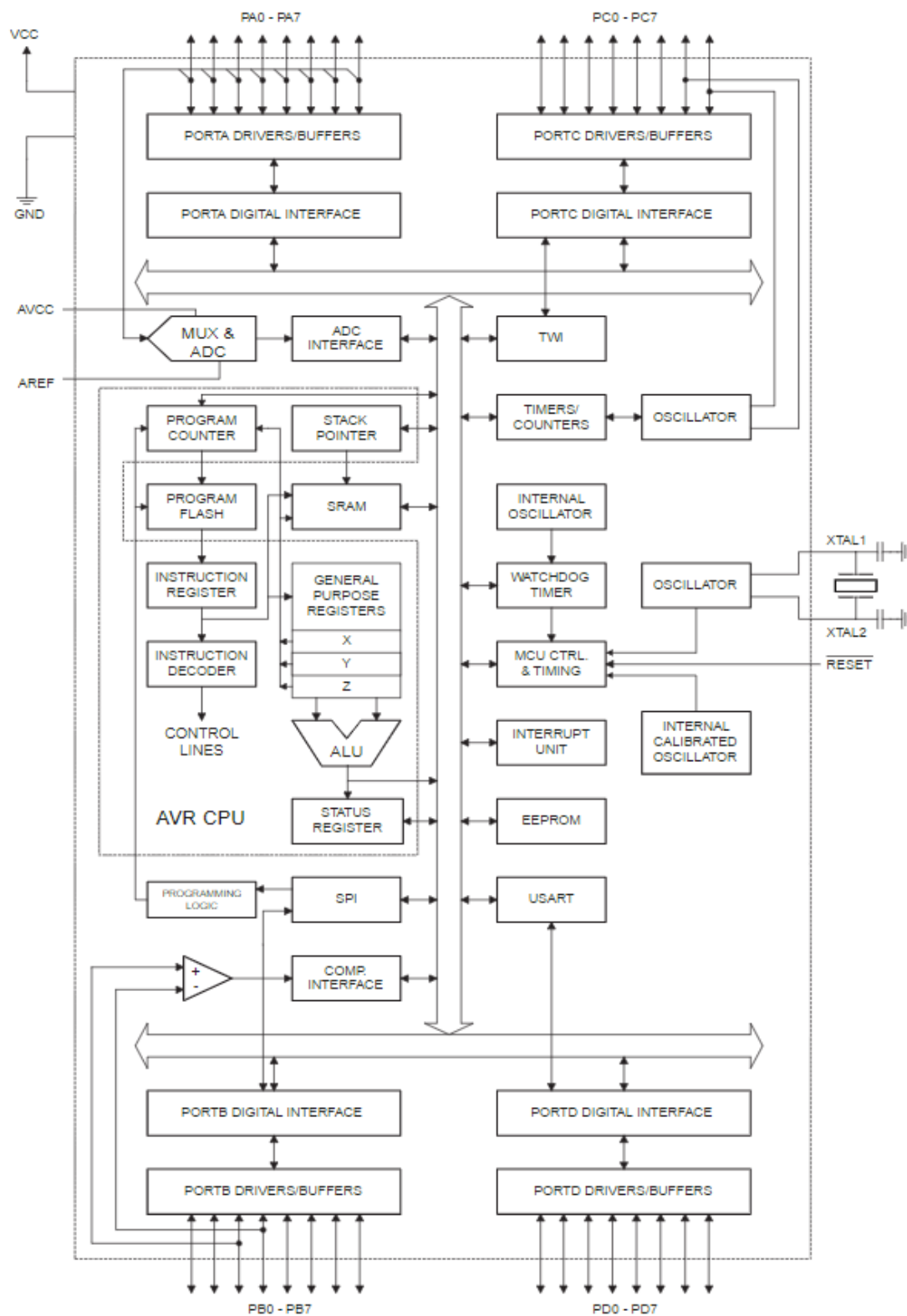


Рисунок 6.2 - Схема ATMEGA16A

3м	Лист	№ докум.	Підп.	Дата

AAAA.123456.007 ПЗ

Лист

8.

Таблиця 6.1 - Порівняльна таблиця мікроконтролерів AVR

Модель	Flash(Кб)	СППЗУ, кБ	ОЗУ (байт)	I/O	Fmax,MHz	Vcc, В	Таймер 16-біт	Таймер 8-біт	ІІММ, каналів	RTC	SPI	UART	TWI	ISP	10-біт АЦП, кан.	Компаратори (ан.)	Детектор Vcc	WD	Osc.	Hardware Multiplier	Корпус
AT90S1200	1	0.0625		15	12	2.7-6.0		1						+		+		+			PDIP20, S0IC20, SSOP20
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
ATmega128L	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+USAR T	1		+	8	+	+	+	+	+	PDIP28, TQFP32, MLF32
ATmega169	16	0.5	1024	54	16	4.5-5.5	1	2	4	+	1+USI	1		+	8	+	+	+	+	+	TQFP64
ATmega169V	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+USAR T	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+USAR T	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
ATtiny15L	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, S0IC20
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

6.2 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 замінювати 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.

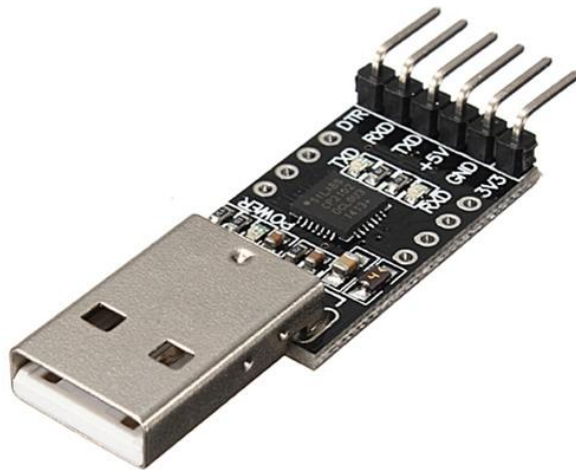


Рисунок 6.3 - перехідник USB/USART

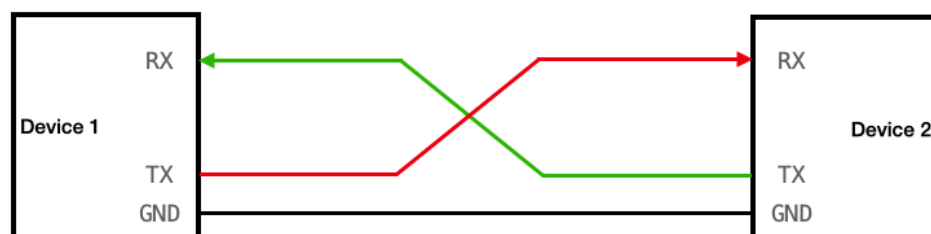


Рисунок 6.4 - схема підключення УАПП

Зм	Лист	№ докум.	Підп.	Дата

AAAA.123456.007 ПЗ

Лист

10.

7 ПРОЕКТУВАННЯ ПРИСТРОЮ

7.1 Розробка 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: РА.0, РА.1, ... , РА.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: РС.0, РС.1, ... , РС.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.

					AAAA.123456.007 ПЗ	Лист
Зм	Лист	№ докум.	Підп.	Дата		11.

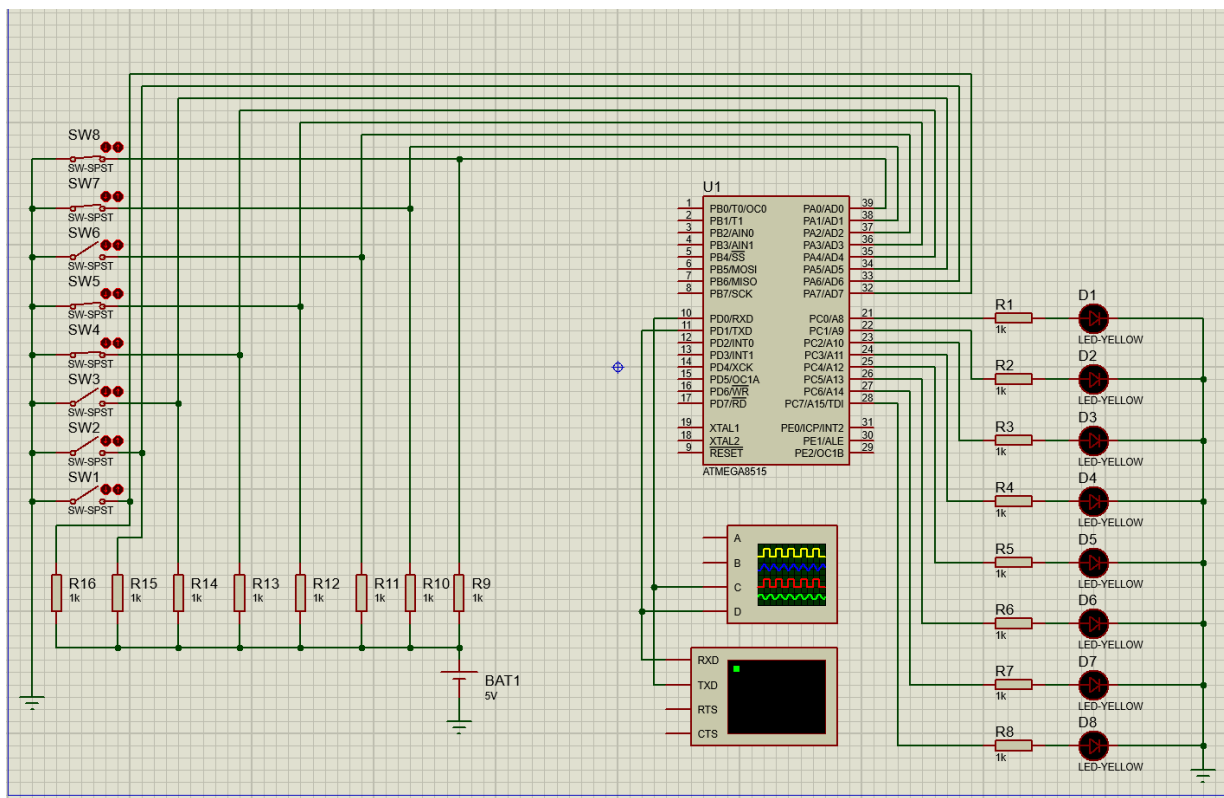


Рисунок 7.1 - Схема робочої моделі

7.2 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»):

$$BAUD = f_{CLK} / (16(UBRR + 1));$$

Зм	Лист	№ докум.	Підп.	Дата

AAAA.123456.007 ПЗ

Лист

12.

- асинхронний режим (пришвидшений, U2Xn= «1»):

$$BAUD = f_{CLK} / (8(UBRR+1))$$

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 f_{CLK} – 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: UBRRH=\$0040; UBRR0L=\$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 UBRRH);

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).

					AAAA.123456.007 ПЗ	Лист
Зм	Лист	№ докум.	Підп.	Дата		13.

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 KC1 має вид:

	7p	6p	5p	4p	3p	2p	1p	0p	
UCSRA	0	0	0	0	0	0	1	0	B=\$02

Регістр UCSRA має адресу у просторі резидентної пам'яті даних мікроконтролера AT mega8515: \$002B.

Тоді програма має вигляд:

LDI R16, \$22; R16←KC1=\$02,

LDI R27, \$00; R27←\$00,

LDI R26, \$2B; R26←\$2B; R27, R26 (X) ←\$002B,

ST X, R16; UCSR0A ←R16= KC1=\$02

Програмування регістра UCSRC

Регістр UCSRC - регістр даних УСАПП. Для мікроконтролера ATmega16A розряд UMSEL регістру програмує режим роботи модуля USART. 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 UCSRB програмують розмір

слова даних. Для нашого прикладу треба запрограмувати: UCSZ1=1; UCSZ0=0, а UCSZ2=0 (передається 7 бітів даних).

Розряд UCPOLE в асинхронному режимі не використовуються, тому запишемо у нього 0. Біт URSEL= 1, тому, що запис виконується в регістр UCSRC.

Тоді керуюче слово KC2 має вид:

	7p	6p	5p	4p	3p	2p	1p	0p	
UCSRC	1	0	1	0	1	1	0	0	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; UCSR0C ←R17 = KC2 = \$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 KC3 має вид:

	7p	6p	5p	4p	3p	2p	1p	0p
UCSRB 0x78	0	1	1	1	1	0	0	0

Регістр UCSRB має адресу: \$002A.

					AAAA.123456.007 ПЗ	Лист
Зм	Лист	№ докум.	Підп.	Дата		15.

Тоді програма має вид:

LDI R19, \$78; R19← KC3=\$78,

LDI R29, \$00; R29←\$00,

LDI R28, \$2A; R28←\$2A R29, R28 (Y)←\$002A,

ST Y, R19; UCSRB ←R19 = KC3= \$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, \$2C; R30←\$2C; 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 – ATmega16A.

					AAAA.123456.007 ПЗ	Лист
Зм	Лист	№ докум.	Підп.	Дата		16.

8 РОЗРОБКА І ОПИС СХЕМИ АЛГОРИТМУ ТА КЕРУЮЧОЇ ПРОГРАМИ

8.1 Схеми алгоритму роботи пристрою

Схему алгоритму роботи пристрою наведено на рисунку 8.1

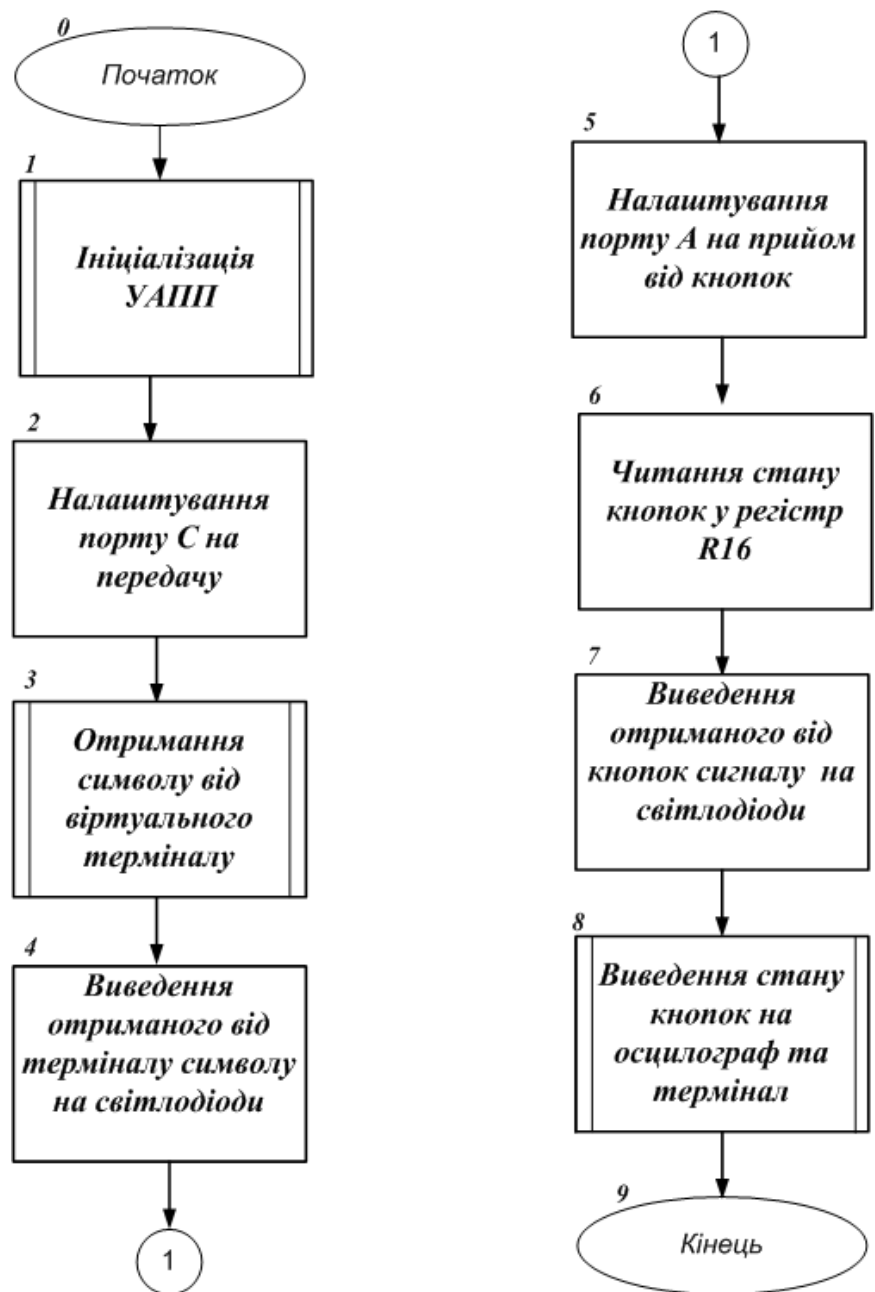


Рисунок 8.1 - Схеми алгоритму роботи пристрою

8.2 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) $R16 \leftarrow 0xFF$

`OUT DDRC, R16` ; (блок 2) $DDRC \leftarrow R16$

					AAAA.123456.007 ПЗ	Лист
Зм	Лист	№ докум.	Підп.	Дата		18.

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) DDRA<- 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

					AAAA.123456.007 ПЗ	Лист
						19.
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

```

9 МОДЕЛЮВАННЯ ПРИСТРОЮ В ПАКЕТІ PROTEUS

Перш за все, потрібно пересвідчитись, що Terminal налаштовано правильно.

Налаштування віртуального термінала 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))

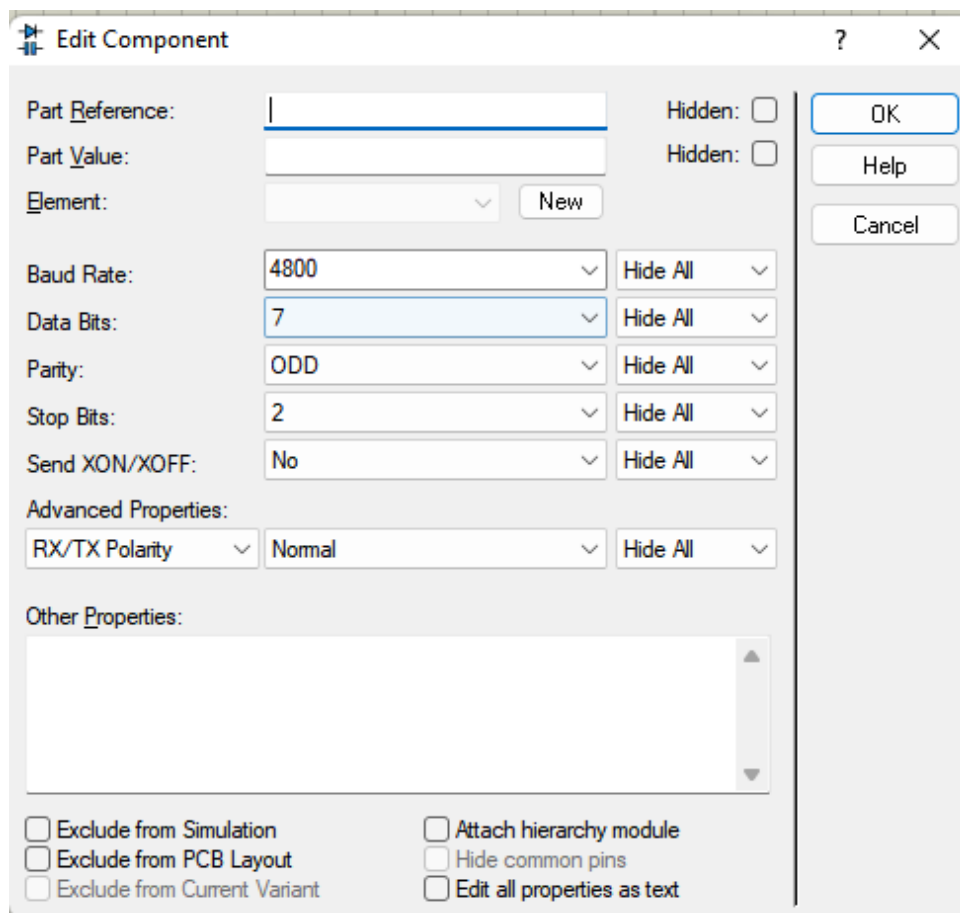


Рисунок 9.1 - Налаштування віртуального термінала

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.

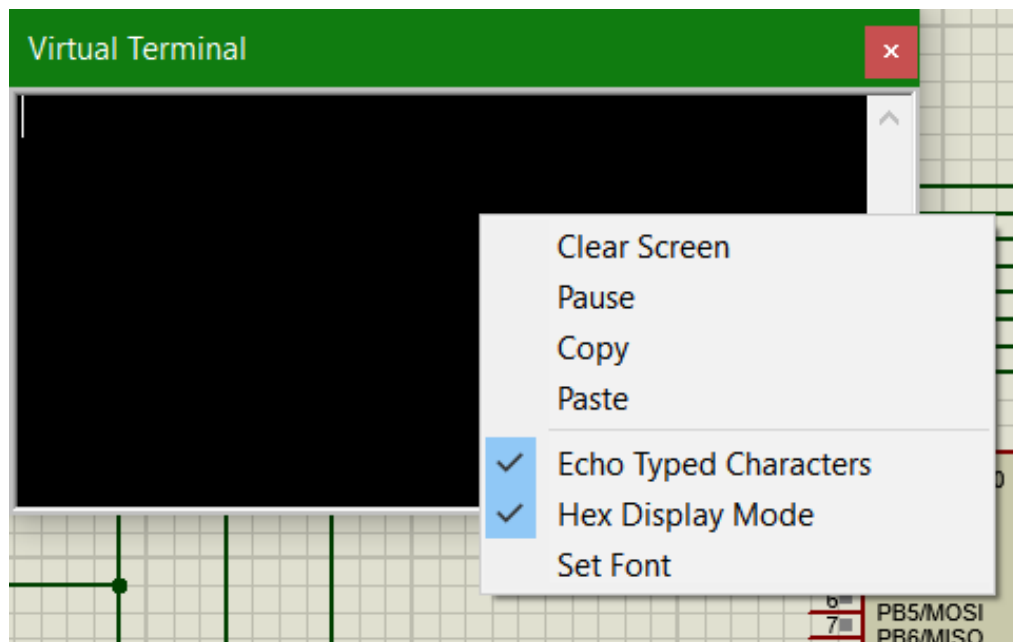


Рисунок 9.2 - Налаштування віртуального терміналу

```

0000      rcall init_USART ; (блок 1) відносний виклик підпрограми
-----
; set port C as OUT
0002      LDI R16, 0xFF      ; (блок 2) R16 <- 0xFF
0004      OUT DDRC, R16      ; (блок 2) DDRC <- R16
-----
0006      rcall USART_receive ; receive from UART [to R16]
-----
; OUT the received data from port C (LEDs connected)
0008      OUT PORTC, R16    ; (блок 4) PORTC <- R16
-----
; set port A as IN
000A      LDI R16, 0x00      ; (блок 5) R16 <- 0x00
000C      OUT DDRA, R16     ; (блок 5) DDRA <- R16
-----
; read buttons into A
000E      in R16, PINA      ; (блок 6) R16 <- PINA
-----
; out read to C (LEDs)
0010      OUT PORTC, R16    ; (блок 7) PORTC <- R16
-----
0012      rcall USART_send ; send read data (R16) by UART
-----
init_USART:
0014      ldi R16, high(UBRR_value) ; (блок 1) R16 <- старший байт UBRR_value
0016      out UBRRH, R16      ; (блок 1) UBRRH <- R16
0018      ldi R16, low(UBRR_value) ; (блок 1) R16 <- молодший байт UBRR_value
001A      out UBRRL, R16     ; (блок 1) UBRRL <- R16
-----
001C      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
001E      out UCSRB, R16     ; write settings above to their corresponding UCSRB register
-----
; 7 бітів даних, перевірка на непарність, 2 стоп-біти
0020      ldi R16, (1<<URSEL)|(1<<UPM1)|(0<<UPM0)|(1<<USCZ1)|(0<<USCZ0)|(1<<USBS)
-----
; URSEL 1 means we write to UCSRC (not UBRRH)
; UPM1,0 set to 1,0 - even parity (outputs 0 if ODD | 1 if EVEN)
; UCSZ2,1,0 set to 0,1,0 - 7 data bits
; USBS set to 1 - 2 stop bits
0022      out UCSRC, R16     ; (блок 1) UCSRC <- R16
-----
0024      ret                ; return [to main]
-----
; передача через модуль USART
USART_send:
0026      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)
0028      sbis UCSRA, TXC    ; (блок 8) якщо біт TXC = 1, то наступна команда пропускається, інакше -
002A      rjmp sending      ; loop
002C      ret                ; return
-----
USART_receive:
002E      sbis UCSRA, RXC    ; loop while receiving
0030      rjmp USART_receive ; loop
-----
0032      in R16, UDR        ; populate R16 with read data
0034      ret                ; return

```

Рисунок 9.3 - Встановлення двох точок зупинки

Зм	Лист	№ докум.	Підп.	Дата

AAAA.123456.007 ПЗ

Лист

22.

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 якого 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 терміналу через виходи порту C (рисунок 9.4).

Світлодіоди та осцилограф відображають бінарний код символу, який було прийнято від віртуального терміналу.

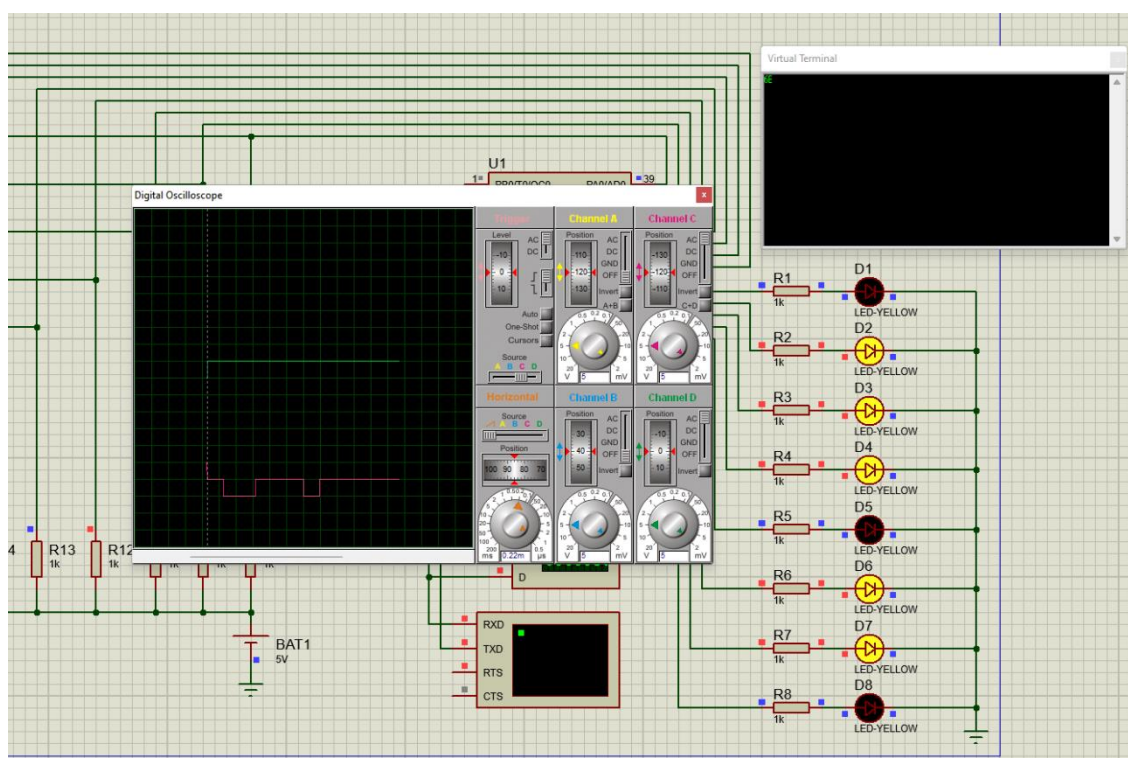


Рисунок 9.4 - Встановлення переданого від терміналу символу на світлодіодах та осцилографі

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 заковано число: **0b01101010 = 0x6A** .

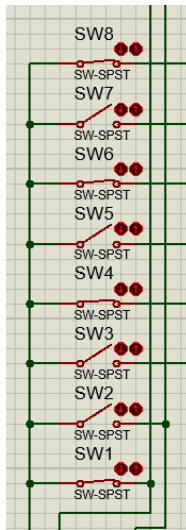


Рисунок 9.5 - Кодування кнопками символу для передачі

Щоб побачити переданий сигнал на осцилографі, потрібно натиснути кнопку на опції One-Shot (один кадр)(Рисунок 9.6)

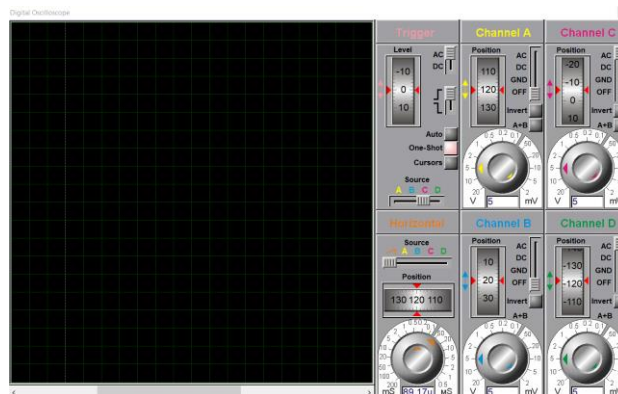


Рисунок 9.6 - Налаштування осцилографа

Для продовження виконання програми знову натискається Run Simulation. Програма переходить до другої точки зупину. Отримуємо дані на осцилографі. (рисунок 9.7)

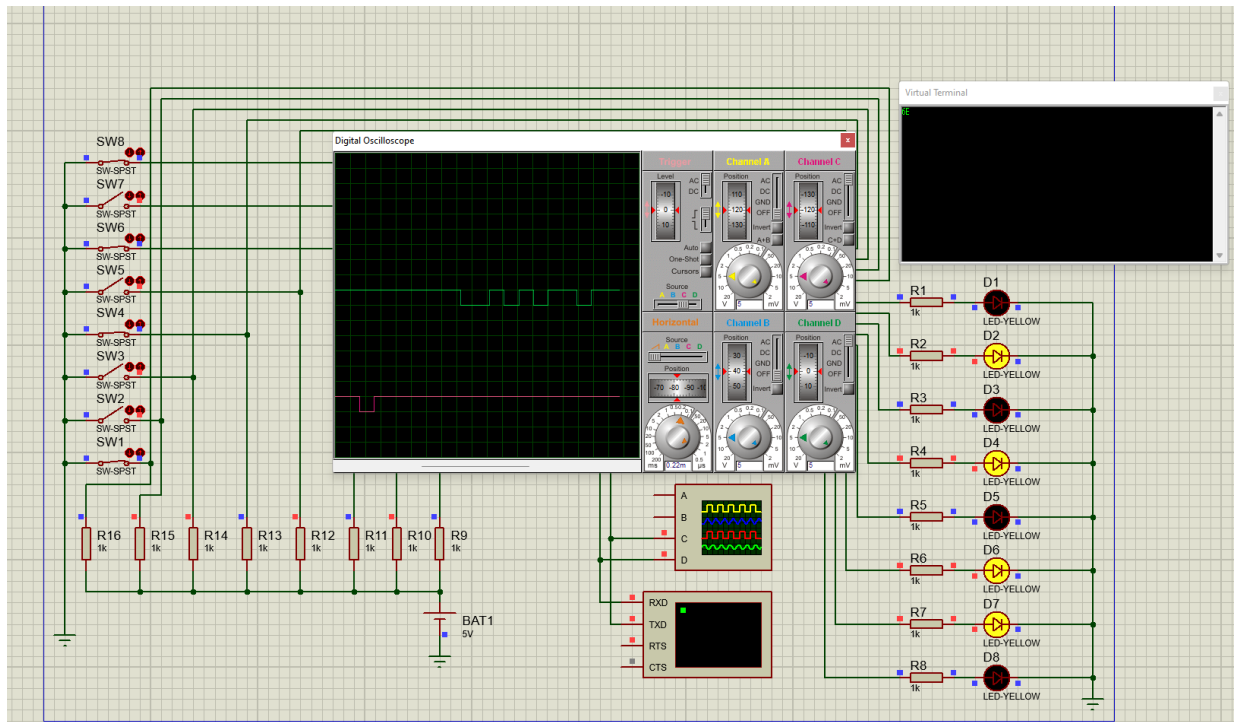


Рисунок 9.7 - Відображення сигналів на осцилографі: від терміналу – червоний; від МК (кнопки) – зелений

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 P_{odd} , який дорівнює 1 та два стоп-біти.

$$P_{odd} = d_{n-1} \oplus \dots \oplus d_3 \oplus d_2 \oplus d_1 \oplus d_0 \oplus 0$$

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 P_{odd} , який дорівнює 0 та два стоп-біти.

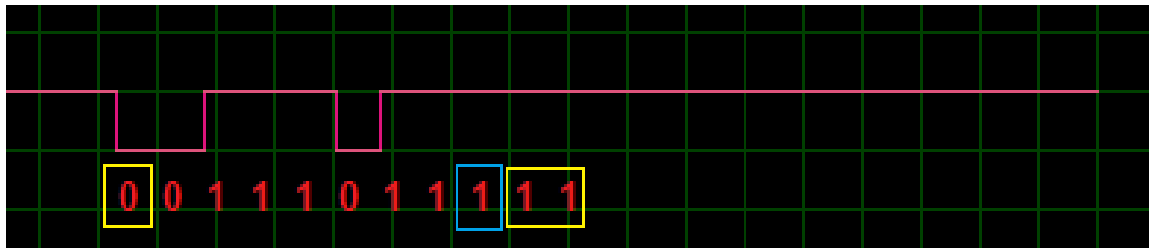


Рисунок 9.8 - Пояснення до сигналу

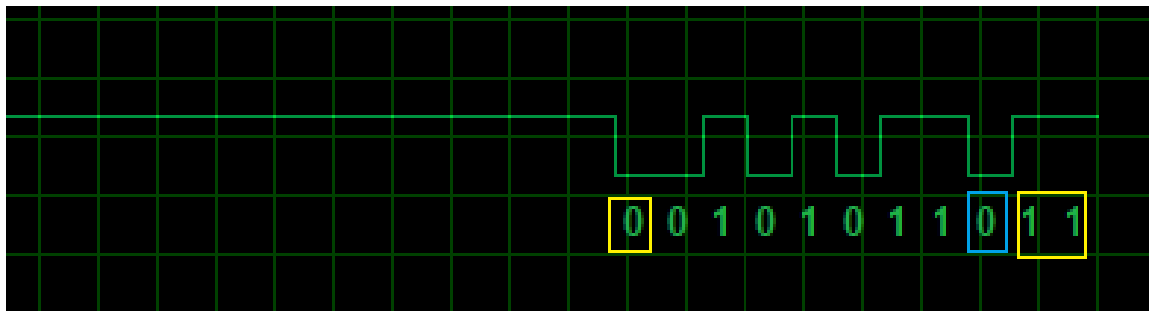


Рисунок 9.9 - Пояснення до сигналу

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).

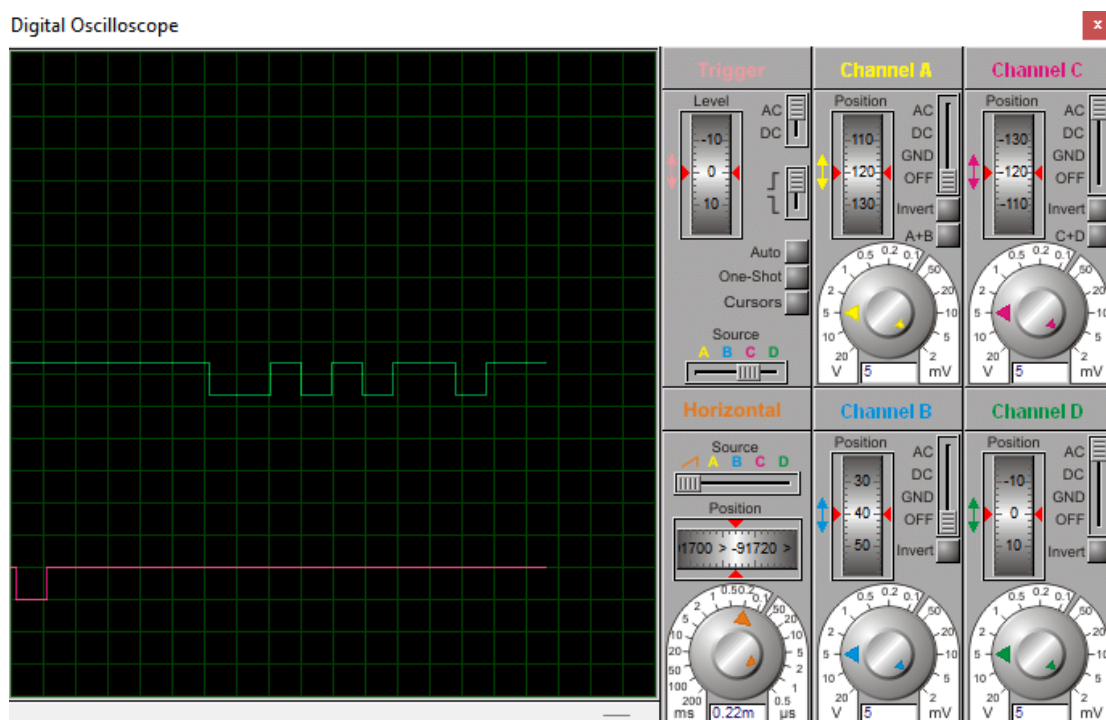


Рисунок 9.10 - Визначення отриманої швидкості передачі на осцилографі

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:

$$V = \frac{1}{0.22} * 1000 \approx 4800 \frac{\text{біт}}{\text{сек}}.$$

Зм	Лист	№ докум.	Підп.	Дата

AAAA.123456.007 ПЗ

Лист

27.

10 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.

11 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&dp_r=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		29.