

**МІНІСТЕРСТВО ВНУТРІШНІХ СПРАВ УКРАЇНИ
ХАРКІВСЬКИЙ НАЦІОНАЛЬНИЙ УНІВЕРСИТЕТ
ВНУТРІШНІХ СПРАВ
КРЕМЕНЧУЦЬКИЙ ЛЬОТНИЙ КОЛЕДЖ**

Циклова комісія авіаційного і радіоелектронного обладнання

МЕТОДИЧНІ МАТЕРІАЛИ

до лабораторних занять

із навчальної дисципліни

«Мікроконтролерні пристрої електромеханічних систем»

обов'язкових компонент

освітньо-професійної програми першого(бакалаврського) рівня вищої освіти

***141 Електроенергетика, електротехніка та електромеханіка
(Електромеханіка)***

ЗАТВЕРДЖЕНО

Науково-методичною радою
Харківського національного
університету внутрішніх справ
Протокол від 30.08.2023 № 7

СХВАЛЕНО

Методичною радою
Кременчуцького льотного коледжу
Харківського національного
університету внутрішніх справ
Протокол від 28.08.2023 № 1

ПОГОДЖЕНО

Секцією Науково-методичної ради
ХНУВС з технічних дисциплін
Протокол від 29.08.2023 № 7

Розглянуто на засіданні циклової комісії авіаційного і радіоелектронного обладнання, протокол від 28.08.2023р № 1.

Розробник: викладач циклової комісії авіаційного і радіоелектронного обладнання, к.т.н., спеціаліст вищої категорії, викладач-методист Волканін Є.Є.

Рецензенти:

1. Доцент кафедри електричних станцій Національного технічного університету «Харківський політехнічний інститут», к.т.н. Шокарьов Д.А.
2. Викладач циклової комісії авіаційного і радіоелектронного обладнання КЛК ХНУВС, к.т.н., професор Гаврилюк Ю.М.

1. Розподіл часу навчальної дисципліни за темами

1.1. Розподіл часу навчальної дисципліни за темами (денна форма навчання)

Не передбачено

1.2. Розподіл часу навчальної дисципліни за темами (заочна форма навчання)

Номер та назва навчальної теми	Кількість годин, відведених на вивчення навчальної дисципліни						Вид контролю
	Всього	з них:					
		Лекції	Семінарські заняття	Практичні заняття	Лабораторні заняття	Самостійна робота	
Семестр № 5							
Тема № 1. Архітектура мікропроцесорів.	28	2		2		24	контр. робота
Тема № 2. Запам'ятовуючі пристрої мікропроцесорних систем.	26	2				24	контр. робота
Тема № 3. Інтерфейс мікропроцесорних систем.	30	2		2		26	контр. робота
Тема № 4. Застосування мікроконтролерів.	32	2			2	28	контр. робота
Тема № 5. Реалізація мікроконтролерних систем керування електроприводами.	34	2		2	2	28	контр. робота
Всього за семестр № 5:	150	10		6	4	130	Іспит

2. Методичні вказівки до лабораторних занять

Тема № 3. Інтерфейс мікропроцесорних систем.

Лабораторне заняття: Моделювання цифрових пристроїв в середовищі PROTEUS (ISIS).

Навчальна мета заняття: моделювання роботи комбінаційних і послідовних цифрових автоматів.

Кількість годин - 2 (денна форма); 0 (заочна форма).

Місце проведення: аудиторія коледжу.

Навчальні питання:

1. Визначення і функції інтерфейсу.
2. Пристрої вводу-виводу інформації.
3. Інтерфейс каналу мікроЕОМ.
4. Спеціалізовані периферійні пристрої.

Література: 1, 2.

План проведення заняття:

I. Вступ до заняття. Проведення попереднього контролю теоретичних знань, практичних умінь і навичок здобувачів вищої освіти.

II. Основна частина заняття.

1. Розгляд наведених теоретичних питань.
2. Моделювання роботи цифрових автоматів є заключною операцією їх синтезу, що дозволяє переконатися в правильності всіх попередніх операцій з отримання рівнянь, мінімізації та остаточного синтезу електричної принципової схеми. Нижче наводиться методика моделювання комбінаційних і послідовних цифрових автоматів.

2.1. Створення проекту в середовищі ISIS. Спочатку створіть на диску папку проекту з ім'ям з не більше 8 латинських символів. Далі відкрийте середу ISIS Proteus (рис. 1), запустивши піктограму ISIS з робочого столу. Інтерфейс середовища містить горизонтальне меню з вертикально розташованими зліва піктограмами швидкого доступу до основних інструментів моделювання, внизу зліва розташовані кнопки управління процесом моделювання. Після появи основного вікна середовища з ім'ям «UNTITLED-ISIS Professional», через пункт основного меню: «Файл - Зберегти проект» в який з'явився стандартному вікні збереження файлу в створеній раніше папці створіть файл проекту з ім'ям що збігається з ім'ям папки.

Закладки бібліотек елементів Пункти меню піктограми поле креслення швидкого доступу кнопки моделювання

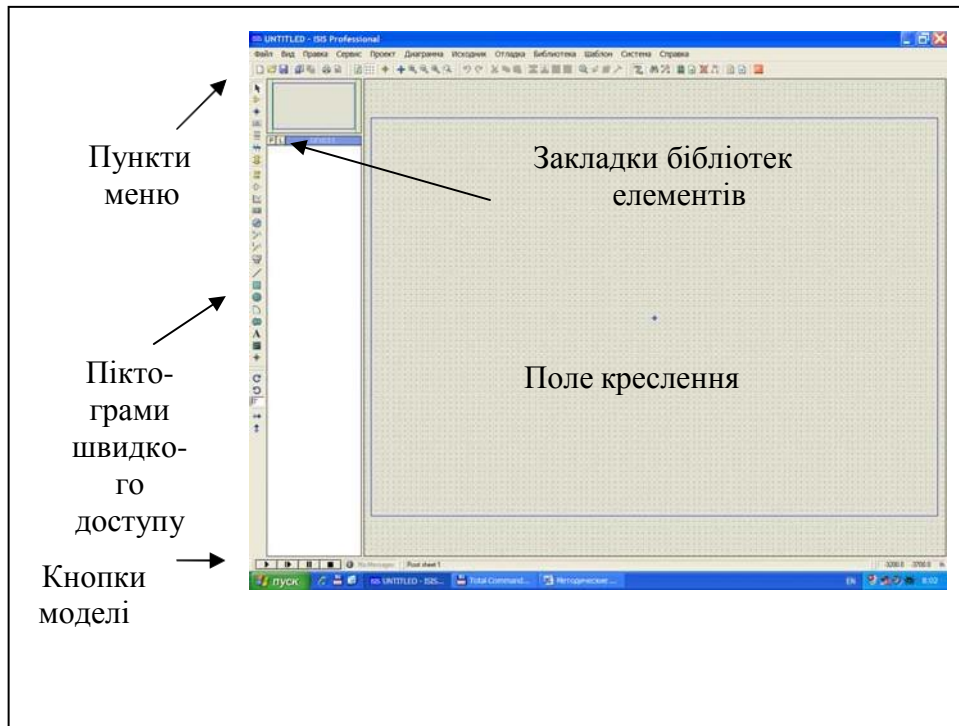


Рисунок - 1 Призначення елементів інтерфейсу в Proteus

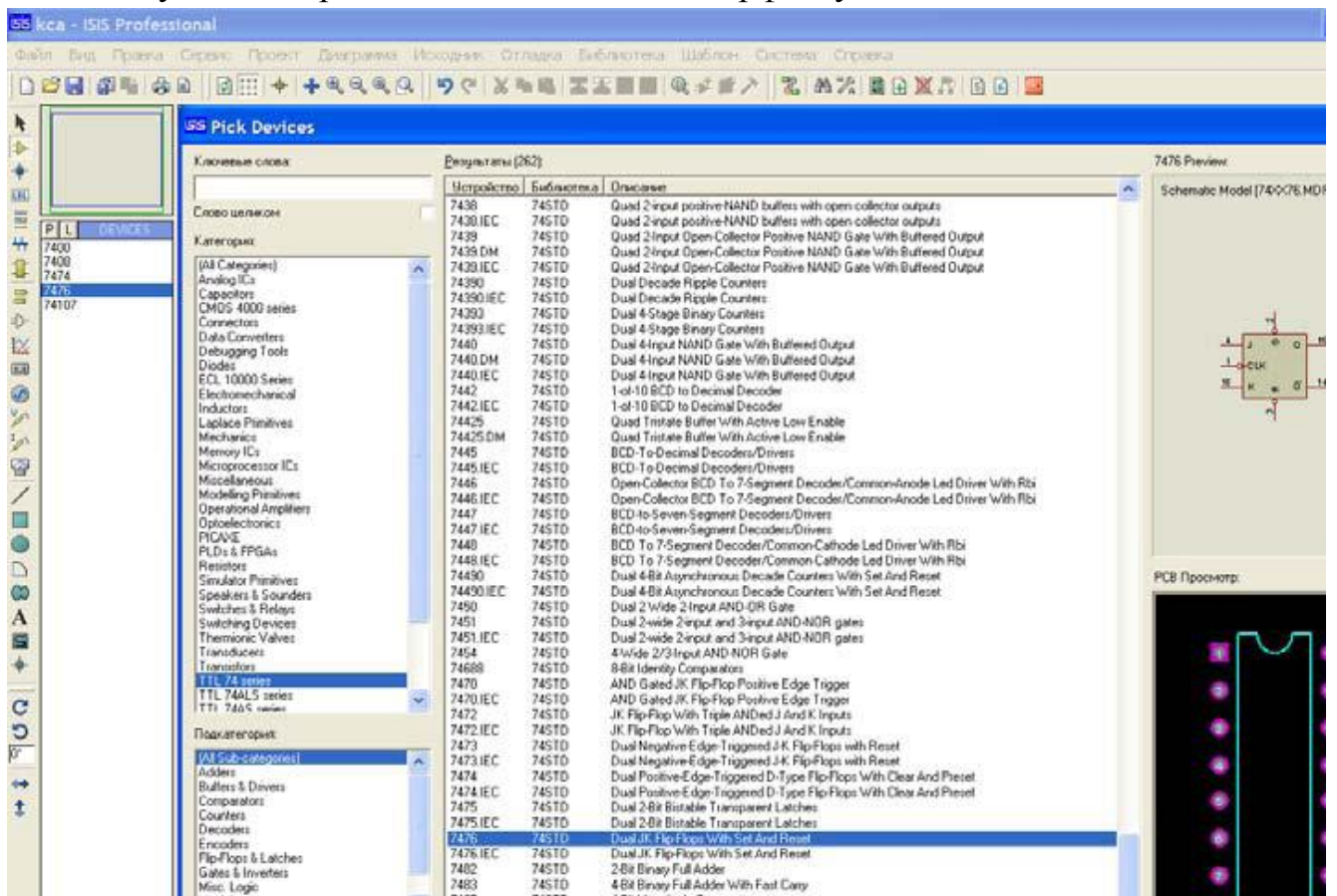




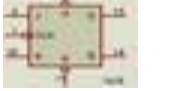



Рисунок - 2 Вибір необхідних елементів схеми

Розширення dsn середовище присвоїть автоматично, а його назва з'явиться у верхньому горизонтальному рядку вікна середовища. У нашому прикладі ім'я файлу - КСА.

2.2. Виберіть потрібні Вам для синтезу схеми логічні елементи (мікросхеми «2І», «2 І-НІ», тригера і т.д.) з бібліотек. Для цього клацніть лівою кнопкою миші по закладці «Р» і у вікні «Pick Devices» (рис. 2) у вікні «Категорія» виберіть рядок «TTL 74 Series». Після цього подвійним клацанням лівої кнопки миші «наберіть» потрібні мікросхеми зі списку, щоб вони з'явилися в лівій колонці вікна проекту.

Таблиця №1 Відповідність мікросхем 74-ої серії і логічних елементів

Логічний елемент	Мікросхема 74-ої серії	Умовне позначення
«2 І-НІ»	7400	
«2І»	7408	
«2АБО-НІ»	7402	
«2АБО»	7432	
JK- тригер	7476	
D-тригер	7474	

При необхідності виберіть інші потрібні елементи. Вибравши елементи, закрийте вікно «Pick Devices». Видалення непотрібних елементів - виділення правою кнопкою миші з наступним видаленням через меню, або клавишею «DEL».

2.3. Створіть електричну принципову схему модельованого пристрою. Для цього, вибираючи, клацанням лівої кнопки миші, елементи зі списку, закріплюючи їх місце розташування - подвійним клацанням лівої кнопки миші (рис. 3). З'єднайте контакти елементів у відповідності із синтезованою схемою. Для цього підведіть курсор під необхідний контакт (при цьому з'явиться червоний квадратик близько контакту на рис. 4), клацніть лівою кнопкою миші для фіксації початку провідника, і ведіть провідник в потрібну точку схеми, завершивши операцію подвійним клацанням (при необхідності змінити напрямок провідника, клацніть в потрібній точці схеми один раз лівою кнопкою).

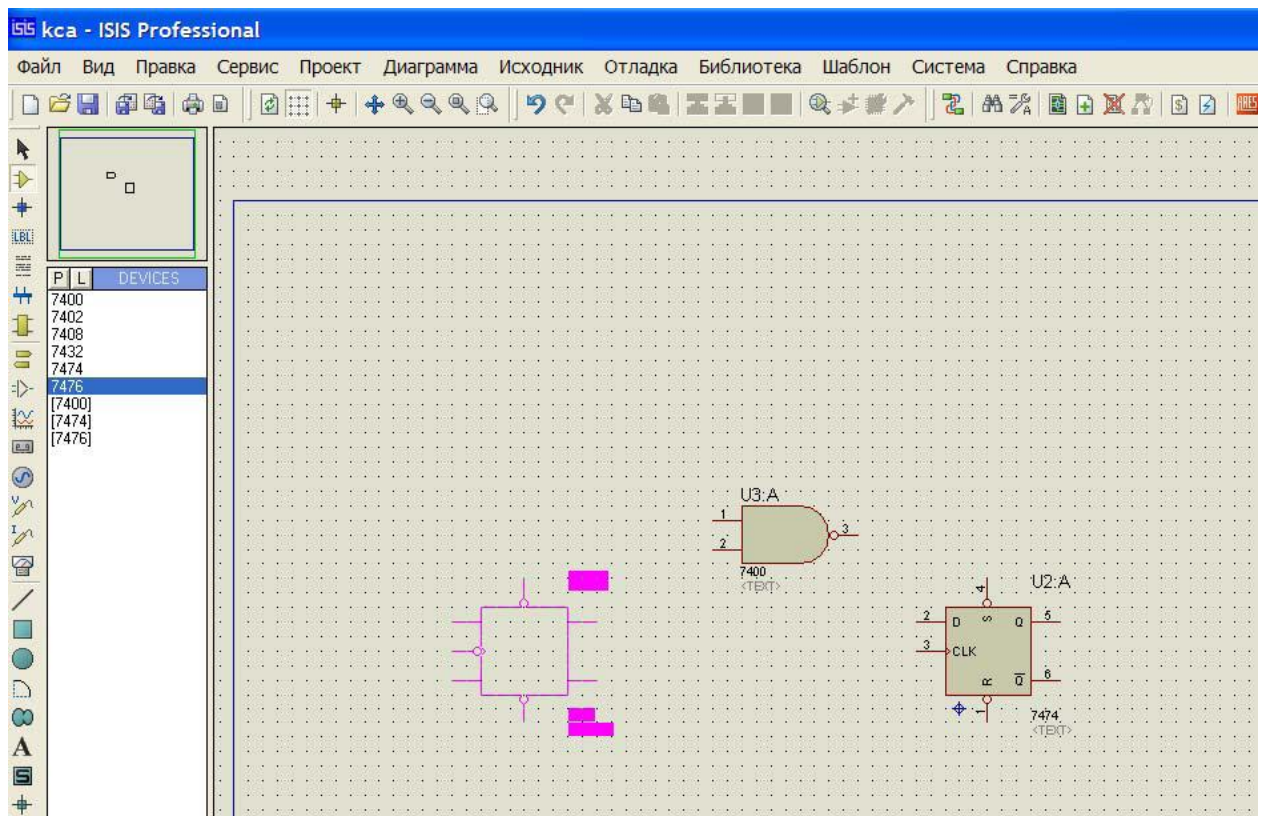


Рисунок - 3 Розташування елементів схеми «полі креслення»

2.4. Подайте потрібні статичні (постійні) сигнали на необхідні контакти мікросхем. На рис. 4 це контакти J K тригера U1: A, і входи S установки початкового стану обох тригерів. Це робиться за допомогою з'єднання цих контактів зі спеціальним елементом, званим терміналом (рис. 5). елементи типу термінал знаходяться в бібліотеці, яка вибирається піктограмою «Термінали», в лівому вертикальному графічному меню (рис. 5). Рівень логічної «1» забезпечує термінал «POWER», а рівень «0» - термінал «GROUND». Розташування їх на кресленні і з'єднання з провідниками і контактами - аналогічно.

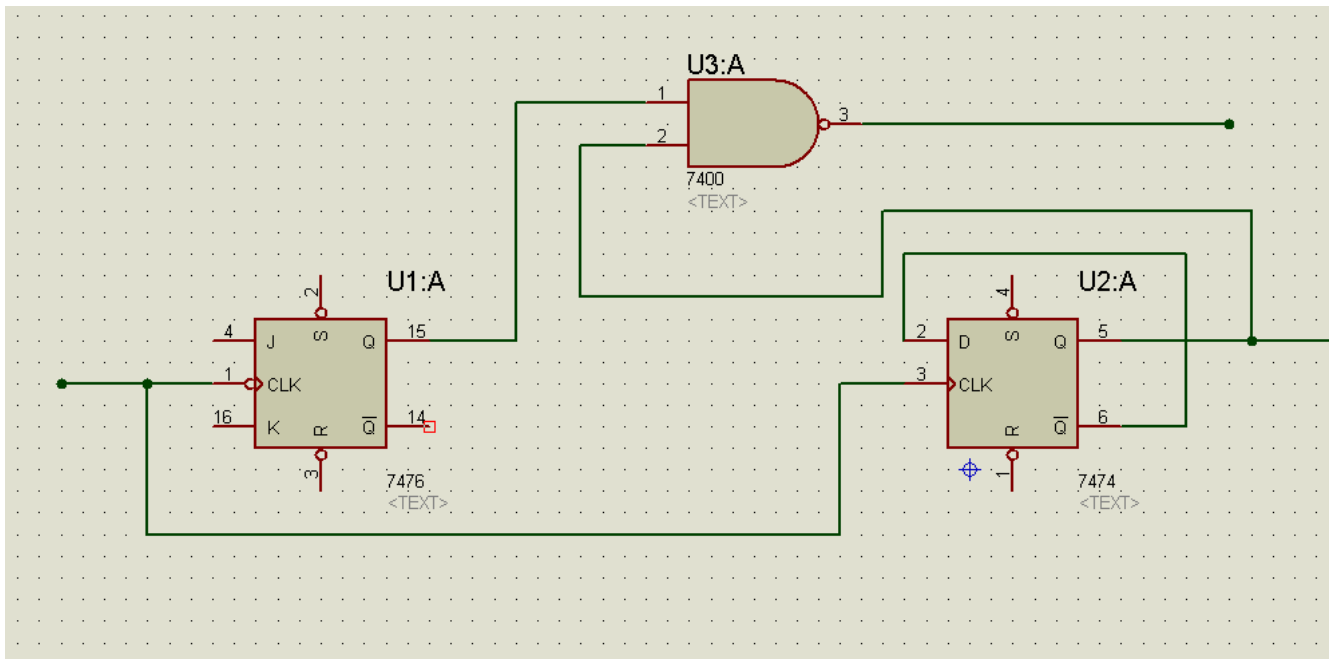


Рисунок – 4 З'єднання елементів схеми на «полі креслення»

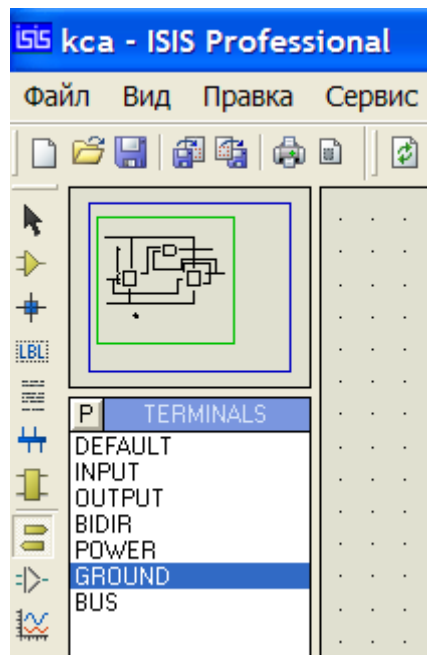


Рисунок – 5 Вибір терміналів «0» і «1»

2.5. Подайте потрібні динамічні (змінні) сигнали на необхідні контакти мікросхем, як правило, це входні сигнали пристрою. На рис. 4 це сигнали на входах CLK тригерів, і сигнали установок на їх вході R. Сигнал на вході CLK - це імпульсний тактовий сигнал, сигнал на входах R - дискретний, що змінюється лише кілька разів в процесі роботи пристрою. Це можна робити по-різному, але для уніфікації ми скористаємося одним генератором, з відповідною зміною параметрів його вихідного сигналу. Вибір генератора як елемента схеми проводиться з бібліотеки генераторів в лівій вертикальній частині основного вікна програми (рис. 6).

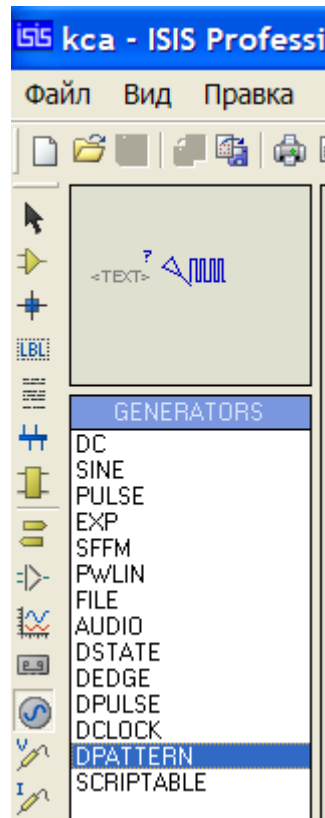
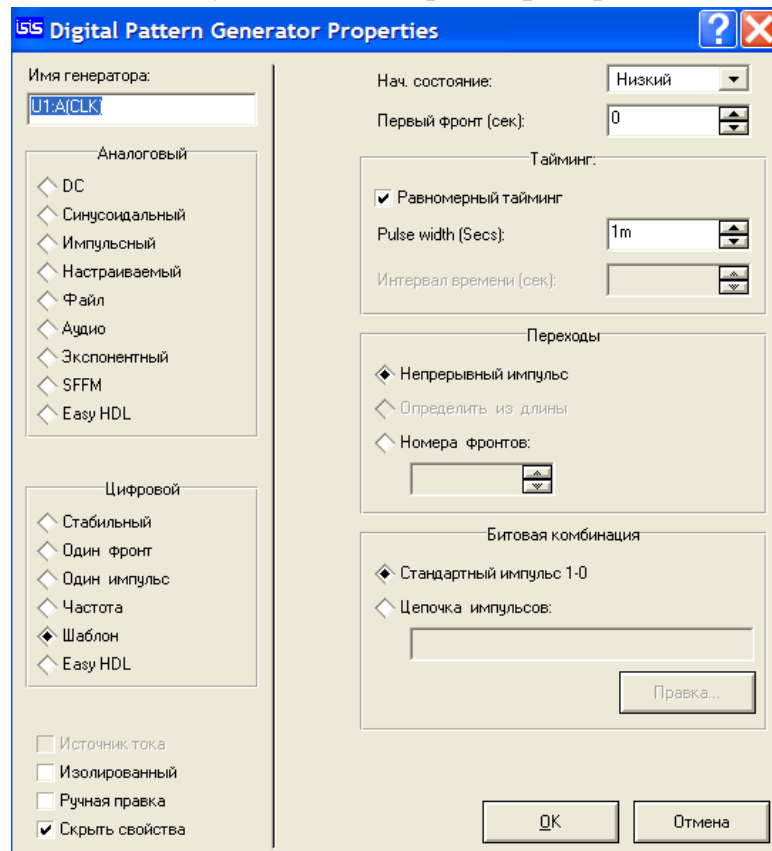
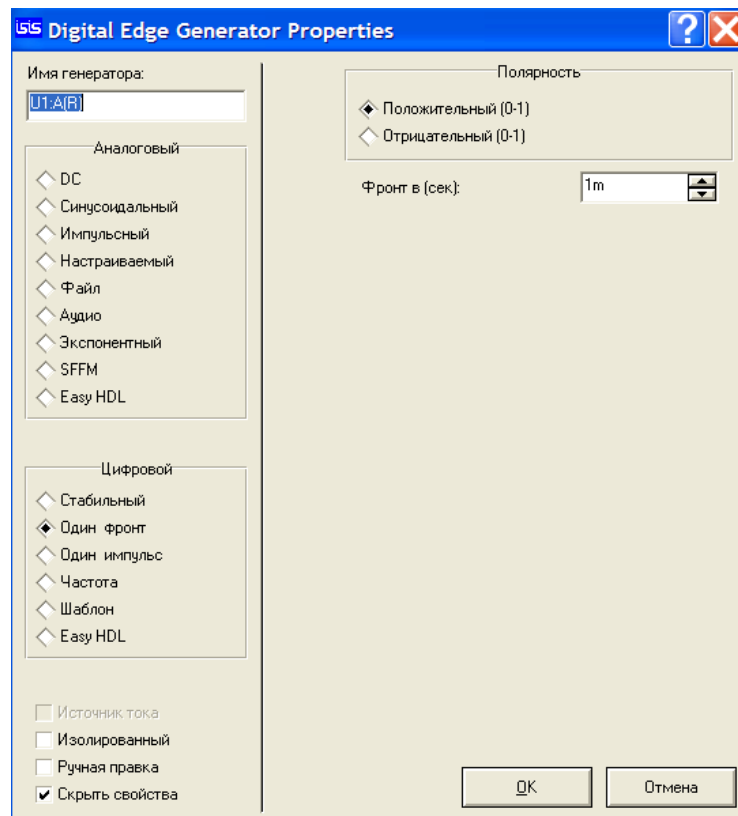


Рисунок – 6 Вибір генераторів



імпульси CLK



Перепад напруги

Рисунок – 7 Конфігурування генераторів вхідних сигналів

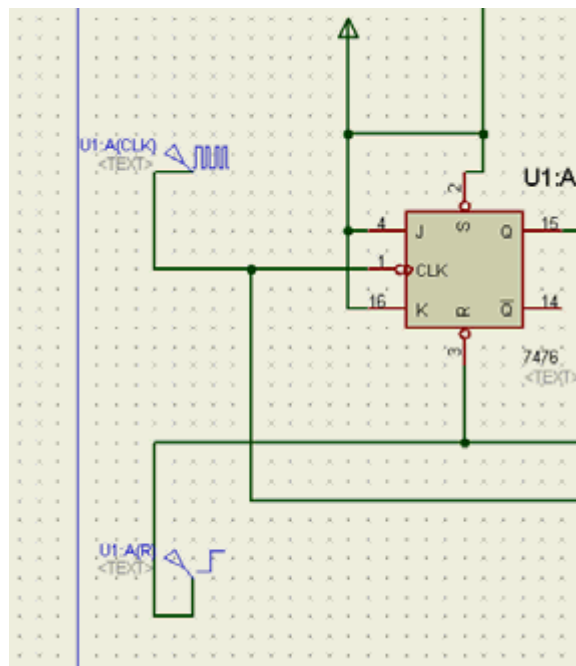


Рисунок – 8 Підключення генераторів

Тип генератора «DPATTERN». Розташування на кресленні і з'єднання з провідниками і контактами - за звичайними правилами. Підключаємо два (або кілька) таких генераторів, але властивості їх змінимо наступним чином. Клацнувши два рази по зображенню генератора, отримуємо доступ до вікна його властивостей (Рис. 7). Для генератора CLK вибираємо вид сигналу «Цифровий -

Шаблон» і встановлюємо галочку у вікні «Рівномірний таймінг» вікна «Таймінг» і тривалість імпульсу (Pulse width) 50m (50 мілісекунд). Для генератора R - параметри відповідно також вказані на рис. 7 (тривалість низького рівня встановлена 1m). Інші параметри імпульсів наведені на рис. 7. Для зручності можна привласнити назви генераторів за змістом сигналів у вікні «Ім'я генератора». Для генератора перепаду відповідно вибираємо вид сигналу - Цифровий / Один фронт; Полярність - позитивна з тривалістю фронту 1 m (як показано на рис. 7). Фрагмент схеми з підключеними генераторами представлений на рис. 8.

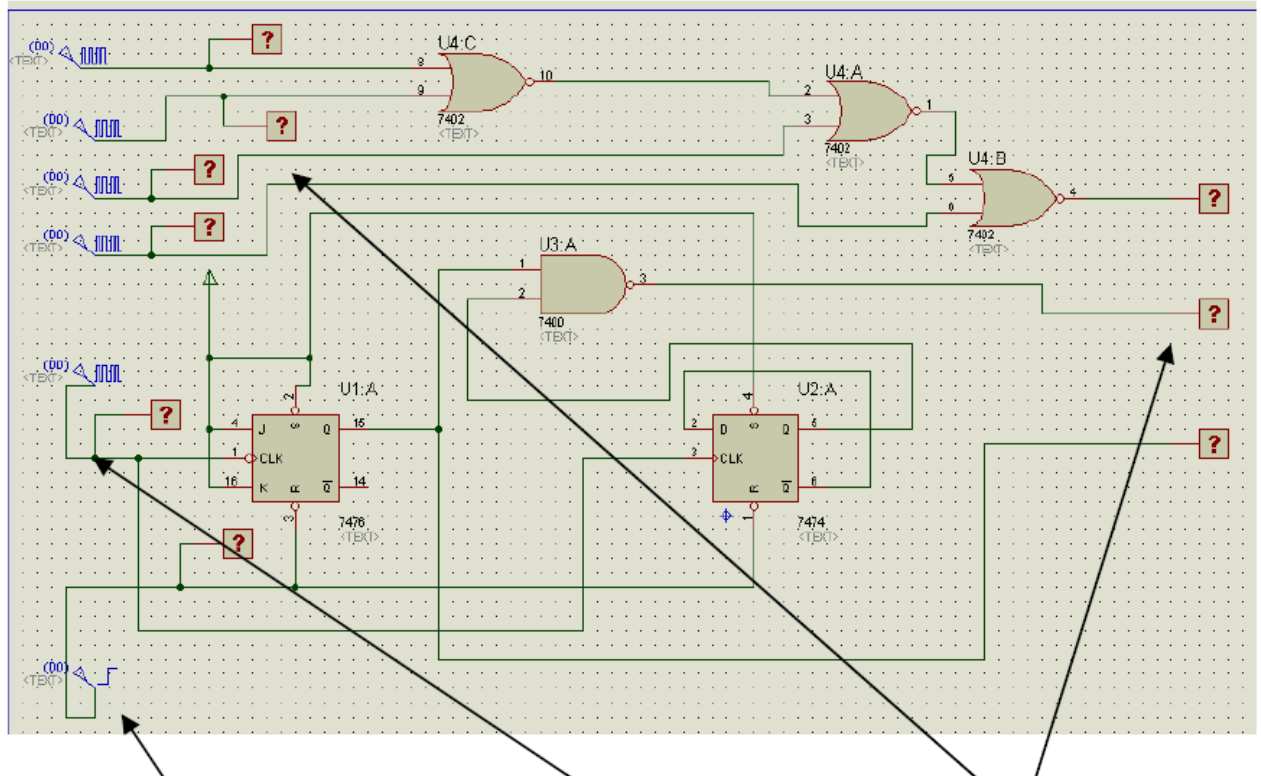
Для генераторів вхідних сигналів КЦА, що мають 5 входів, частоти 5-ти генераторів відрізняються кожен в 2 рази, у відповідність з їх розташуванням в таблиці функціонування (тривалість імпульсу відповідно рівні 50m - 100m - 200m - 400m - 800m і т.д.).

2.6. Підключаємо логічні пробники до цікавлять ланцюгах, як правило, це вхідні і вихідні ланцюги пристрою, хоча при пошуку «несправностей» може знадобитися їх підключення і до «передбачувано несправним» ділянкам ланцюгів схеми. Для їх вибору з бібліотек клацніть лівою кнопкою миші по закладці «Р» і у вікні «Pick Devices» (рис. 2) у вікні «Ключові слова» наберіть LOGICPROBE (BIG). Після цього подвійним клацанням лівої кнопки миші «виберіть» зазначений компонент зі списку, щоб він з'явився в лівій колонці вікна проекту. Додайте їх на поле креслення і підключіть до необхідних ланцюгах стандартним чином. Результуюча схема з підключеними приладами показана на рис. 9 (в ній доданий ряд логічних елементів для демонстрації подачі вхідних сигналів, періоди яких розрізняються у відповідність з кодом «8-4-2-1»). При необхідності натисканням клавіші F8 розташуйте всю схему на екрані.

2.7. Збережіть проект через основне меню програми.

2.8. Перед початком моделювання встановіть тривалість кроку моделювання: в головному меню System / Set Animation Options у вікні параметрів Single Step Time = 51m. Моделювання роботи схеми проводиться в покроковому режимі шляхом одноразового натискання на піктограми в лівому нижньому кутку середовища (рис. 10). При цьому пробники будуть відображати рівні логічних сигналів на місці значків «?». При цьому одне натискання кнопки «Крок» відповідає 1 тактовому імпульсу CLK або сигналу з мінімальним періодом. Слід зазначити, що в процесі моделювання аналогічні «маленькі пробники» з'являться близько контактів мікросхем автоматично, проте їх розміри незручні для аналізу рівнів сигналів. При збігу осцилограмм з законами зміни вихідних сигналів з таблиці функціонування моделювання вважає

При збігу осцилограмм з законами зміни вихідних сигналів з таблиці функціонування моделювання вважається закінченим. В іншому випадку, зупинивши моделювання, перейдіть до редагування схеми.



Генератор перепаду R Генератор імпульсів Пробники

Рисунок – 9 Схема пристрою з підключеними генераторами і пробниками, підготовлена до моделювання



Рисунок – 10 Піктограми керування моделюванням

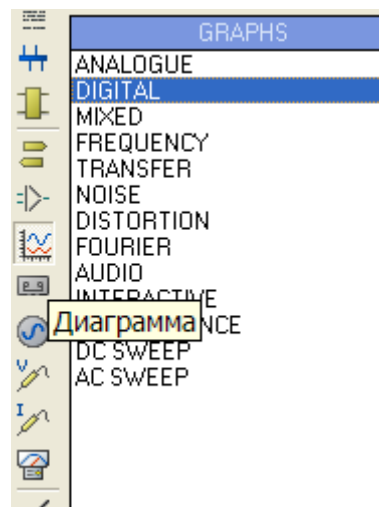


Рисунок – 11 Вибір графічного аналізатора

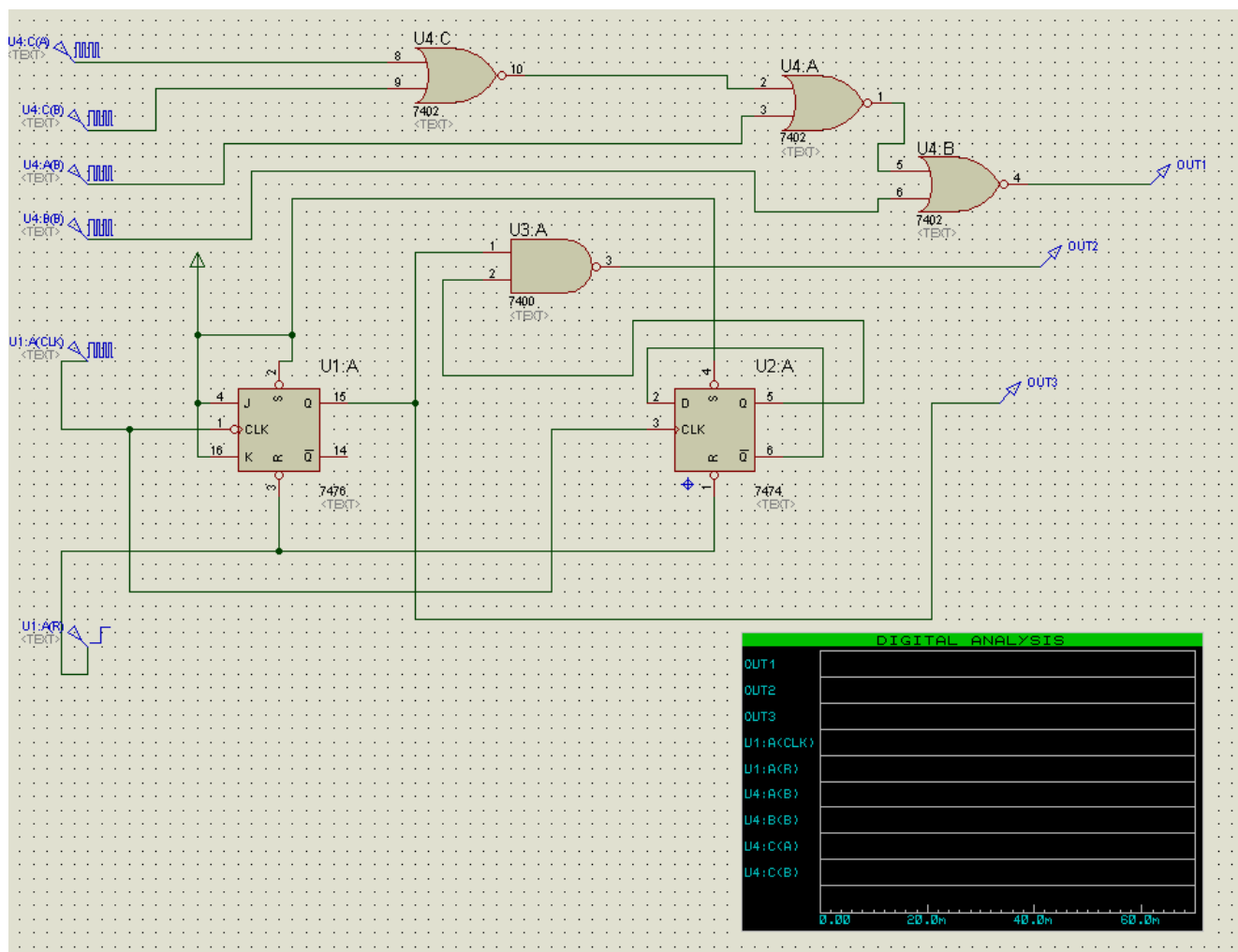


Рисунок – 12 Схема пристрою з підключеними генераторами і аналізатором, підготовлена до моделювання

2.9. Редагування схеми проводиться в тому випадку, якщо в результаті моделювання були виявлені помилки функціонування, тобто робота схеми відрізнялася від її роботи по таблиці функціонування. Після внесення змін до схеми, повторіть моделювання.

2.10. У разі необхідності спостереження за роботою цифрової схеми в безперервному режимі роботи, можна скористатися графічним аналізатором. Підключаємо графічний аналізатор до цікавлять нас точкам схеми. Як правило, це все вхідні і вихідні сигнали схеми. Це робиться в три етапи.

2.10.1. Вибір аналізатора здійснюється з бічних піктограм: «Діаграма-GRAPHS-DIGITAL» (рис. 11). Після вибору аналізатора, на полі креслення мишею з натиснутою лівою кнопкою «малюється» в потрібному місці «системі приладу», завершуючи повторним клацанням лівої кнопки миші.

2.10.2. Підключаємо щупи напруги до необхідних вихідних (або інших) проводам схеми. Вибір щупів здійснюється з бічної піктограми одноразовим клацанням лівою кнопкою миші: «Щуп напруги» з фіксацією його на необхідному дроті таким же клацанням.

2.10.3. Налаштовуємо параметри аналізатора: - клацнувши лівою кнопкою по зображенню аналізатора, в який з'явився ок не "Діаграма перехідного процесу» встановлюємо параметр «Кон. час »рівним 2s (або інше, так щоб на діаграмі вміщувалося необхідне число тактових імпульсів для аналізу роботи схеми). - клацнувши правою кнопкою по зображенню аналізатора, вибираємо пункт «Додати трасу» і у вікні «Add Transient Trace» в рядку «Щуп P1» зі списку вибираємо необхідний провід (точніше ім'я генератора або щупа).

Аналогічно підключаємо всі необхідні дроти схеми. Для зручності аналізу розташуйте сигнали в наступному порядку:

- сигнал CLK;
- вхідні сигнали в порядку ЗБІЛЬШЕННЯ їх періоду;
- вихідні сигнали.

Результуюча схема з підключеними приладами показана на рис. 12.

При роботі з аналізатором необхідно враховувати: - назви трас (проводів) можуть бути тільки англійські; - видалити трасу можна.

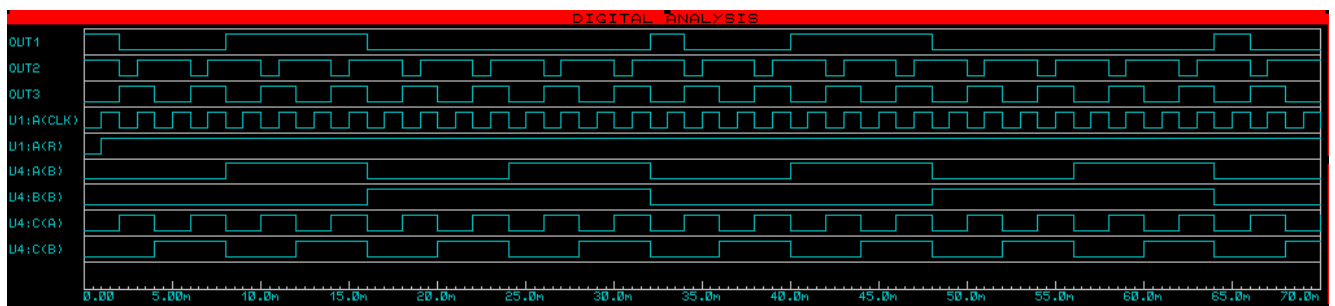


Рисунок – 13 Результати моделювання на графічному аналізаторі

Моделювання роботи схеми проводиться натисканням клавіші «Пропуск». При цьому на діаграмі відображаються тимчасові діаграми в обраних точках схеми (рис. 13). Якщо розгорнути вікно діаграми, подвійним клацанням лівої кнопки миші по зеленій смужці діаграми, то утримуючи ліву кнопку миші, з'являється вертикальний переміщуваний маркер, а в лівій частині діаграми відображаються рівні сигналів («1» = Н, «0» = L) на тимчасових діаграмах.

III. Заключна частина заняття. Перевірка і оцінювання виконаних завдань. Підведення підсумків лабораторного заняття, акцентування уваги на основних помилках при його виконанні.

Тема № 4. Застосування мікроконтролерів.

Лабораторне заняття: Застосування мікроконтролерів.

Навчальна мета заняття: моделювання роботи програмованих цифрових пристроїв на мікроконтролерах PIC 16C52 з використанням середовища MPLAB.

Кількість годин - 4 (денна форма); 2 (заочна форма).

Місце проведення: аудиторія коледжу.

Навчальні питання:

1. Застосування мікроконтролерів в електромеханічних системах.
2. Застосування мікроконтролерів в електроапаратобудуванні.
3. Застосування МК51 при випробуваннях електричних апаратів захисту.
4. Структурна схема АСУ ТПВ з паралельними АЦП.

Література: 1, 2.

План проведення заняття:

I. Вступ до заняття. Проведення попереднього контролю теоретичних знань, практичних умінь і навичок здобувачів вищої освіти.

II. Основна частина заняття.

1. Розгляд наведених теоретичних питань.

2. При розробці ПО для мікроконтролерів сімейства PIC найдоцільніше користуватися спеціалізованим програмним продуктом, рекомендованим виробником контролерів - середовищем MPLAB, яка дозволяє створювати і налагоджувати вихідний код. Однак на практиці часто користуються більш зручними в плані емуляції вхідних сигналів і візуалізації результатів продуктами, до яких можна віднести і Proteus. В цьому випадку MPLAB використовується для розробки, редагування, налагодження вихідного коду в asm-файлі і спостереження за станом внутрішніх регістрів контролера. Proteus - для імітації вхідних сигналів і спостереження за роботою вихідних портів контролера до яких підключені зовнішні виконавчі пристрої. При використанні середовища Proteus для моделювання роботи PIC контролерів вихідний код і раніше створюється в середовищі MPLAB. Крім того, через відсутність моделі PIC16C52 в середовищі Proteus його доводиться замінювати іншим контролером (PIC16F877). При цьому потрібно дотримуватися деяких правил, тому що контролер PIC16F877 значно функціональніша. Розглянемо методику моделювання PIC 16C52 з використанням середовищ Proteus і MPLAB.

3. Створення проекту і розробка вихідного коду на мові асемблер в середовищі MPLAB включає в себе наступні кроки:

3.1. Для написання вихідного коду (asm-файлу) рекомендується використовувати Microchip MPLAB IDE версії старше 7.50 (на момент написання посібника остання версія 8.53).

3.2. Після запуску MPLAB в меню виберіть «Project» → »Project Wizard». На- тисніть «Далі». Виконайте запропоновані кроки:

- крок 1: створіть папку в якій будуть зберігатися файли проекту. Для нашого прикладу, розглянутого далі, це буде папка example на диску C.

- крок 2: зі списку пропонованих контролерів «Devices» виберіть PIC16F877. Натисніть «Далі».

- крок 3: виберіть мову програмування - асемблер, для цього переконайтеся, що у вікні «Active Toolsuite» обраний «Microchip MPASM Toolsuite» і натисніть «Далі».

- крок 4: введіть назву проекту і виберіть місце його розташування. Натисніть «Browse ...», виберіть папку для збереження проекту, створену раніше в кроці 1. Введіть його ім'я і натисніть «Зберегти». У нашому прикладі це буде "c: \ example \ ex1" - назва проекту буде "ex1", папка "c: \ example". Натисніть «Далі».

- крок 5: необов'язковий крок додавання раніше створених файлів в проект. Нічого не змінюючи, натисніть «Далі» потім, «Готово». У головному вікні середовища і вікні зі списком файлів проекту заголовок зміниться на назву проекту.mcw

3.3. Створення asm-файлу. Для цього виберіть пункт «Project» → «Add New File to Project ...». У діалоговому вікні введіть ім'я файлу з розширенням ".asm" і збережіть. Після цього доданий файл відобразиться у вікні файлів проекту в папці "Source Files" і з'явиться нове вікно з моргає курсором для введення вихідного коду ПЗ з ім'ям asm-файлу в заголовку.

3.4. Розробка вихідного коду ПЗ проводиться у відповідність з алгоритмом роботи контролера, синтаксисом мови, системою команд і директив.

Подальший виклад наведено для ПО, що реалізує наступний алгоритм: Приклад: розробити ПО для контролера 16C52, що здійснює прийом байта даних з порту В, операцію арифметичного додавання цього байта з константою 0x4D і записуючий результат в РОН з адресою 0x2C. При роботі контролера на порте А повинні бути постійно виставлені сигнали 1011. При розробці ПО необхідно обов'язково враховувати відмінності контролера 16C52 від 16F877: - адресація РОН-ів починається з адреси 0x20, тобто перший вільний регістр має адресу 0x20, а не 0x07; - вхід лічильника T0СКІ є контактом порту А RA4, тобто фактично 5-им контактом порту, тому обов'язкова конфігурація порту А повинна бути: `movlw 0x3 * tris PORTA`, де * - необхідна Вам комбінація вхід / вихід 4-х контактів порту А;

- регістр FSR 8-ми розрядний, а не 5-ти;

- ПО повинно починатися з блоку команд, наведеного на рис. 14. Для написання коду активізуйте вікно asm-файлу, вставте в нього обов'язкову частину, після мітки begin розташуйте Ваш код (рис. 15), закінчивши його командою end.

3.5. Компіляція вихідного коду здійснюється шляхом натискання клавіші F10. При першій компіляції середу видасть повідомлення (рис. 16), виберіть пункт «Relocatable». У разі відсутності синтаксичних помилок середу видасть у вікні

компіляції «Output» повідомлення «BUILD SUCCEEDED», при їх наявності - повідомлення «BUILD FAILED». У цьому ж вікні з'являться повідомлення про характер помилок і номерах рядків ПЗ, в яких вони виявлені. Проаналізуйте помилки, після чого закрийте вікно «Output». Усуньте синтаксичні помилки і знову відкомпілюйте код. Завершивши роботу, збережіть результат (пункт меню «Project» → «Save Project») і вийдіть з-поміж MPLAB.

```

PROCESSOR PIC16F877
errorlevel -302
__config __WDT_OFF
#include <p16f877.inc>
org 0x000
movlw 0x20
movwf STATUS
clrw
movwf PCLATH
movlw 0x07
movwf ADCON1
banksel ADCON0
clrf PORTA
clrf PORTB
goto begin
begin

```

Рисунок – 14 Обов'язкова початкова частина ПЗ для 16C52

```

PORTA      equ 0x05
PORTB      equ 0x06
movlw 0xFF
tris PORTB
movlw 0x30
tris PORTA
movlw 0x4D
addwf PORTB,0
movwf 0x2C
bsf PORTA,0x03
bcf PORTA,0x02
bsf PORTA,0x01
bsf PORTA,0x00
goto begin
end

```

Рисунок – 15 Код ПЗ для 16C52

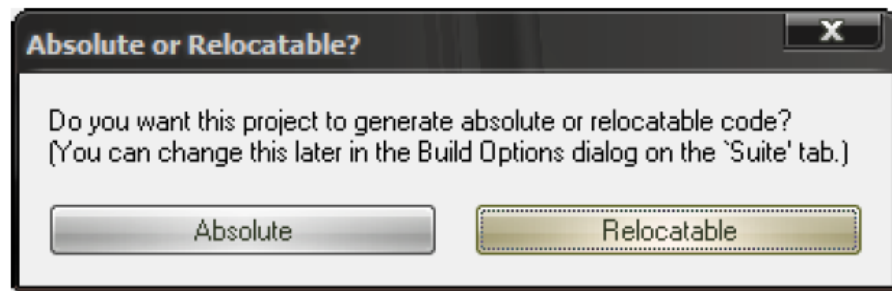


Рисунок – 16 Повідомлення середовища перед першою компіляцією

4. Створення схеми з'єднання контролера з зовнішніми пристроями в середовищі Proteus. Це робиться для організації взаємодії між середовищем MPLAB і Proteus, тому що Proteus дозволяє наочно формувати вхідні і візуалізувати вихідні сигнали контролера в процесі його роботи разом з підключеними до нього пристроями.

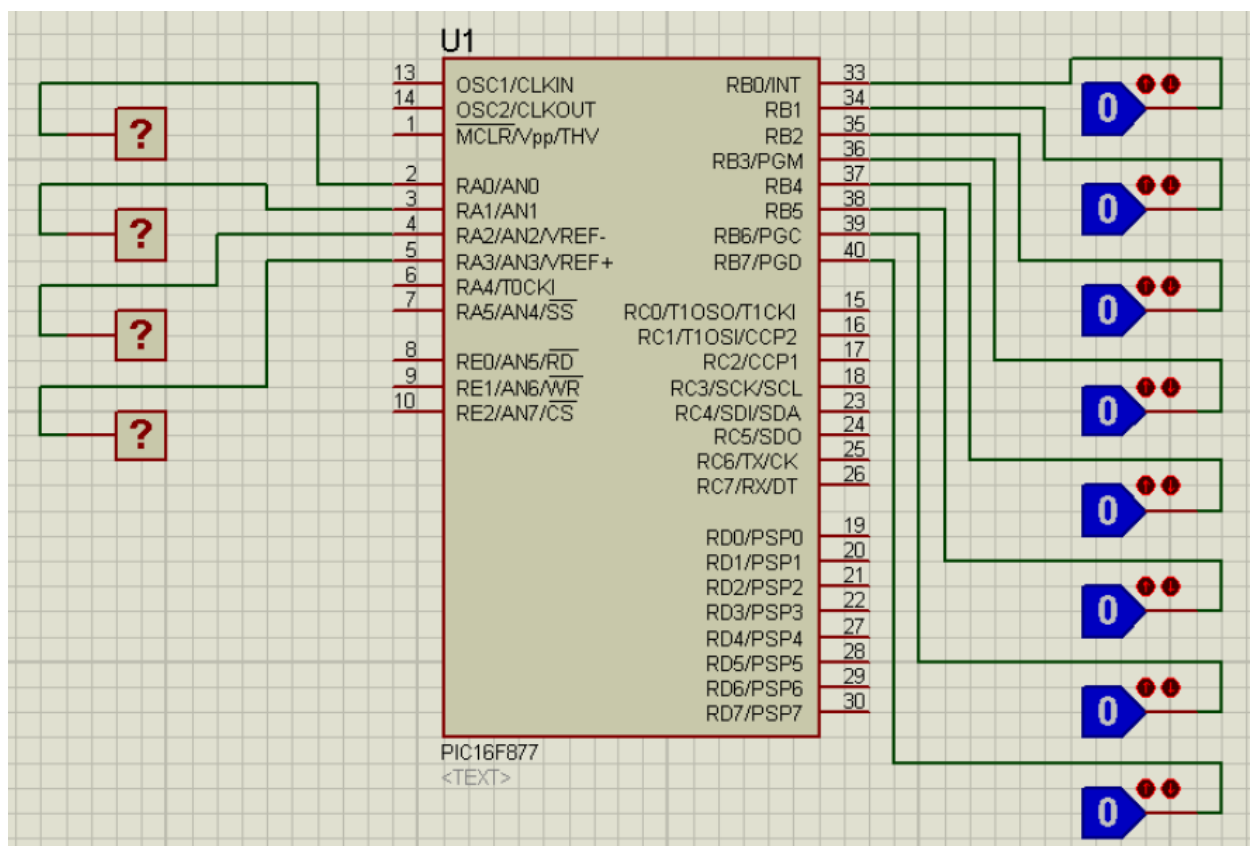


Рисунок – 17 Схема пристрою в середовищі Proteus

4.1. Використовуючи п.п. 1-7 методики з розділу 1.1 цього посібника створіть схему, яка містить контролер з підключеними до нього джерелами вхідних сигналів і вихідними виконавчими пристроями:

- файл проекту Proteus збережіть в папці проекту MPLAB;
- при виборі елементів схеми використовуйте контролер PIC 16F877 (ввівши

його назву в рядку «Keywords» вікна «Pick Devices»);

- до вихідних контактах контролера (в нашому прикладі це 4 молодших розряди порту A, RA3 - RA0, RA3 - старший розряд) підключіть індикатори: LOGICPROBE (BIG) (знайшовши їх аналогічно);

- на вхідні контакти портів контролера (в нашому прикладі це все 8 контактів порту B, RB7 - RB0, RB7 - старший розряд) подайте вхідні сигнали. Для цього використовуйте або фіксується в натиснутому положенні кнопку LOGICSTATE, або аналогічну кнопку, але без фіксації - LOGICTOGGLE. В результаті у Вас повинна вийти схема (рис. 17).

4.2. У властивостях контролера (подвійне клацання по ньому лівою клавішею миші) необхідно виставити його частоту рівної 4MHz. Змініть поле 'Processor Clock Frequency' і натисніть «OK».

4.3. Збережіть проект і закрийте Proteus.

5. Підключення схемної моделі контролера з середовища Proteus до вихідного коду в середовищі MPLAB для цього:

5.1. Відкрийте Ваш проект (створений в п. I) в середовищі MPLAB, через пункт меню «Project» → «Open ...». Закрийте вікна «Output», «ex1.mcw».

5.2. Виберіть засіб налагодження ПО в MPLAB через пункт меню 'Debugger' → 'Select Tool' → 'Proteus VSM'. Повинно з'явитися вікно «Proteus MPLAB VSM Viewer». Якщо воно не з'явилося, активізуйте його через пункт меню «View» → «Proteus VSM Viewer».

5.3. У вікні «Proteus MPLAB VSM Viewer» через стандартну піктограму «Open Design» відкрийте схему в середовищі Proteus, і масштабуйте її для зручності спостереження.

5.4. Відкрийте додаткові вікна для перегляду вмісту регістрів контролера. Для цього виберіть пункт «View» → «File Registers» для регістрів загального призначення і пункт «View» → «Special Functions Registers» для спеціальних регістрів. Упорядкуйте їх на екрані, вибравши пункт «Window» → «Tile Vertically» (Рис. 18), закривши зайві вікна.

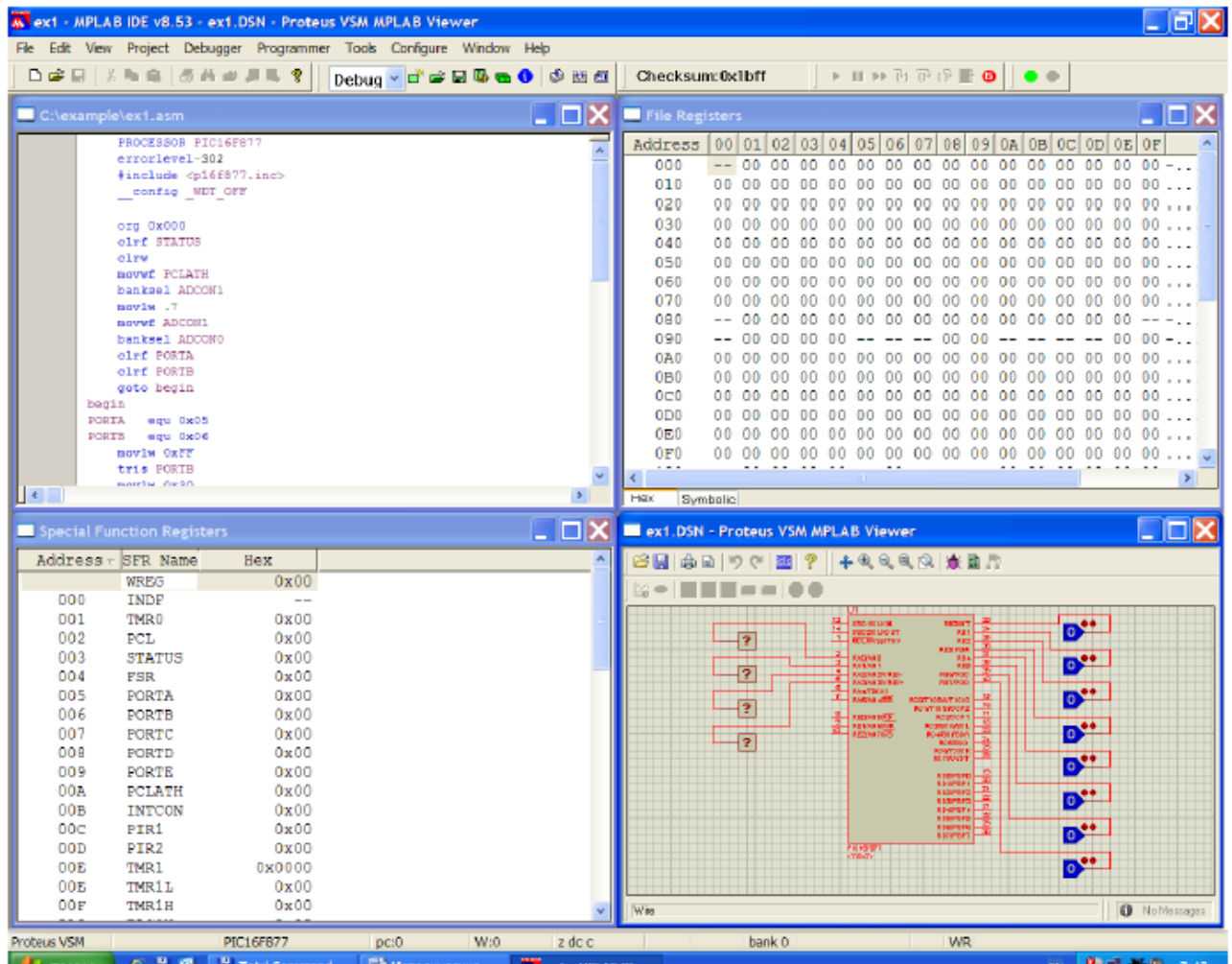


Рисунок – 18 Вид середовища MPLAB, підготовленого для налагодження ПЗ контролера 16C52, з підключеним вікном середовища Proteus

6. Проведення налагодження програми проводиться в такій послідовності:

6.1. Відкомпілюйте ПЗ, натиснувши клавішу F10. Якщо п.3.5 був виконаний правильно, то помилок бути не повинно, в іншому випадку усуньте їх в тексті asm-файлу. При відсутності помилок закрийте вікно «Output».

6.2. Почніть покрокову налагодження ПЗ:

- встановіть у вікні Proteus необхідні за умовами завдання початкових значень вхідних сигналів;

- почніть налагодження, натиснувши на панелі інструментів кнопку з зеленим колом (або клавішу F12). При цьому навпроти першої виконуваної команди asm- файлу з'явиться зелена стрілка;

- проведіть покрокову налагодження, натискаючи клавішу F7 і подаючи в вікні Proteus необхідні вхідні сигнали. При зміні вхідних сигналів слідкуйте, щоб ці зміни «побачив» кон'роллер, тобто щоб вони з'явилися в регістрі портів у вікні спеціальних реєстрів MPLAB (а не тільки у вікні PROTEUS). Поки цього не сталося, контролер буде обробляти «минулі» значення вхідних сигналів. Кожне натискання клавіші F7 призводить до виконання однієї команди контролером.

Необхідно щоб перед виконанням команди обробки вхідних сигналів (принаймні за 1 крок перед нею) їх нове значення вже було в регістрі порту (в нашому прикладі це порт В). Іноді може знадобитися повторне натискання кнопки, щоб зміна сигналу з'явилася в регістрі порту, або повторна емуляція (з виходом через натискання кнопки з червоним кружком і повторним її запуском через натискання кнопки з зеленим кружком). У нашому прикладі такою командою є команда `addwf PORTB`, яка складає прочитаний з порту В сигнал з байтом константи, що знаходяться в регістрі W. Тому, наприклад, якщо у вікні Proteus змінити сигнал на контактах порту В безпосередньо перед виконанням цієї команд (по нажатію клавіші F7), то MPLAB цього зміни «не побачить» і команда виконається зі «старим» вмістом порту В.

- в процесі роботи спостерігайте за зміною контактів портів контролера і його внутрішніх регістрів. Налагодження ПЗ полягає в тому, щоб перевірити правильність функціонування (відповідність алгоритму із завдання) контролера при ВСІХ можливих комбінаціях вхідних сигналів у всіх заданих режимах роботи;

- на практиці для спостереження за станом спеціальних регістрів контролера іноді буває зручно скористатися спеціальним вікном, в якому буде відображатися стан тільки обраних Вами регістрів. Це робиться через пункт меню «View» → «Watch»;

- для перевірки працездатності контролера при великому числі можливих комбінацій вхідних сигналів, або великому числі команд в коді, можна скористатися режимом анімації (пункт Debugger → Animate). В цьому режимі ПЗ почне автоматично виконуватися з деякою фіксованою швидкістю і залишиться лише змінювати вхідні сигнали; - для завершення налагодження натисніть на кнопку з червоним колом (або комбінацію клавіш CTRL-F12).

6.3. При збігу результатів роботи контролера з заданим алгоритмом, пред'явіть роботу викладачеві, при наявності розбіжностей внесіть зміни до початкового коду і повторіть налагодження. Після внесення будь-яких змін в `asm`-файл повторно відкомпілюйте код.

III. Заключна частина заняття. Перевірка і оцінювання виконаних завдань. Підведення підсумків лабораторного заняття, акцентування уваги на основних помилках при його виконанні.

Тема № 5. Реалізація мікроконтролерних систем керування електроприводами.

Лабораторне заняття: Реалізація мікроконтролерних систем керування електроприводами.

Навчальна мета заняття: розглянути приклади реалізації мікроконтролерних систем керування електроприводами.

Кількість годин - 4 (денна форма); 2(заочна форма).

Місце проведення: аудиторія коледжу.

Навчальні питання:

1. Керування двофазним безколекторним електродвигуном постійного струму без датчиків.
2. Керування асинхронним електродвигуном змінного струму за принципом постійності V/f і звичайного ШІМ-керування.

Література: 1, 2.

План проведення заняття:

I. Вступ до заняття. Проведення попереднього контролю теоретичних знань, практичних умінь і навичок здобувачів вищої освіти.

II. Основна частина заняття.

1. Розгляд наведених теоретичних питань.
2. Віртуальний стенд являє собою модель реального лабораторного стенду на промисловому контролері TTF 5.0, що містить електричну схему з'єднання моделі мікроконтролера PIC 16F877 з моделями зовнішніх пристроїв. Віртуальний стенд призначений для запису в модель контролера, його «прошивки», отриманої в середовищі MPLAB, і спостереження за роботою контролера з сигналами датчиків і виконавчих пристроїв стенду в середовищі Proteus, що не підключаючись до реального обладнання.

Схема стенду № 1 включає в себе:

- мотор з датчиком числа його оборотів. Управляється двигун цифровим сигналом.
- датчик температури DS1820.

Модель дозволяє імітувати зміна поточної температури датчика, шляхом натискання на символи «+», «-». При цьому на його виході за попереднім запитом контролера формується послідовний код, а значення температури відображається у внутрішньому вікні моделі.

- 1.5 розрядний семисегментний індикатор.
- незалежна пам'ять 25C040 об'ємом 512 байт з інтерфейсом SPI. Додатково модель дозволяє спостерігати за станом всіх своїх внутрішніх регістрів. Для цього, натисніть кнопку «Пауза», а потім, клацнувши правою кнопкою миші по зображенню моделі мікросхеми пам'яті і вибравши пункт «Internal Memory», у вікні «SPI Memory Internal Memory» Ви отримаєте доступ до поточного стану

внутрішніх регістрів РПЗУ. Зверніть увагу, що стан регістрів буде відображено на момент часу безпосередньо ПЕРЕД зупинкою моделювання. У режимі моделювання це вікно не відображається, тобто стану регістрів можна спостерігати тільки в момент зупинки моделювання, а не в процесі їх зміни.

- клавіатура з 4-ма не зафіксованими клавішами. Емуляція утримуваної в натиснутому стані кнопки проводиться клацанням лівої кнопки миші по червоному кружечку поруч з моделлю кнопки.

- аналоговий осцилограф. Стандартний компонент Proteus, по органам управління відповідає 4-х канальному реальному осцилографу.

На вхід А підключений цифровий сигнал з контакту RD2 мікроконтролера, на вхід В - аналоговий сигнал з виходу інтегратора, підключеного до контакту RC1 модуля ШИМ контролера. На вхід 3 підключений вихід генератора прямокутних імпульсів, що розглядається нижче. Це зроблено для зручності візуальної настройки тривалості імпульсів цього генератора. Активізація компонента відбувається тільки в режимі «Play».

- генератор імпульсів. Стандартний компонент Proteus, має органи плавного і дискретного управління амплітудою (регулятор Level - плавно, регулятор Range - множник) і частотою (регулятор Centre - плавно, регулятор Range - множник), і видом генерується сигналу (в нашому випадку як генератор прямокутних імпульсів). Відповідає реальному генератору Г5-63, сигнал з якого подається на вхід RA5 мікроконтролера PIC 16F877 (і паралельно, для зручності візуальної настройки, на вхід 3 осцилографа). Режим генерації прямокутних імпульсів можливий тільки зі шпаруватістю 2. Активізація компонента тільки в режимі «Play».
- регулятор напруги. Модель реалізована у вигляді змінного опору і за сигналами управління повністю збігається з реальним пристроєм.

- датчик освітленості. Модель реалізована у вигляді змінного опору, вихідна напруга з якого, при зміні положення движка, імітує зміну вихідної напруги з датчика при зміні його освітленості.

- датчик температури t_{сп}-100. Модель реалізована у вигляді змінного резистора, вихідна напруга з якого, при зміні положення движка, імітує зміну вихідної напруги з датчика при зміні температури навколишнього середовища.

- гучномовець. Модель реалізована у вигляді вбудованого компонента Proteus з вбудованим (на схемі не показаний) перетворювачем «частота-напруга». Перетворювач виконує роль інтегратора, що виділяє постійну складову з імпульсної послідовності з ШИМ на контакти RC2 мікроконтролера.

При моделюванні формування звукових сигналів за допомогою моделі гучномовця необхідно враховувати такі особливості: 1. Для формування паузи в звучанні гучномовця (фактично його виключення) контролер ОБОВ'ЯЗКОВО повинен формувати на контакті RC2 імпульсний сигнал з шпаруватістю 2 і довільним періодом а не просто рівень логічного «0». 2.

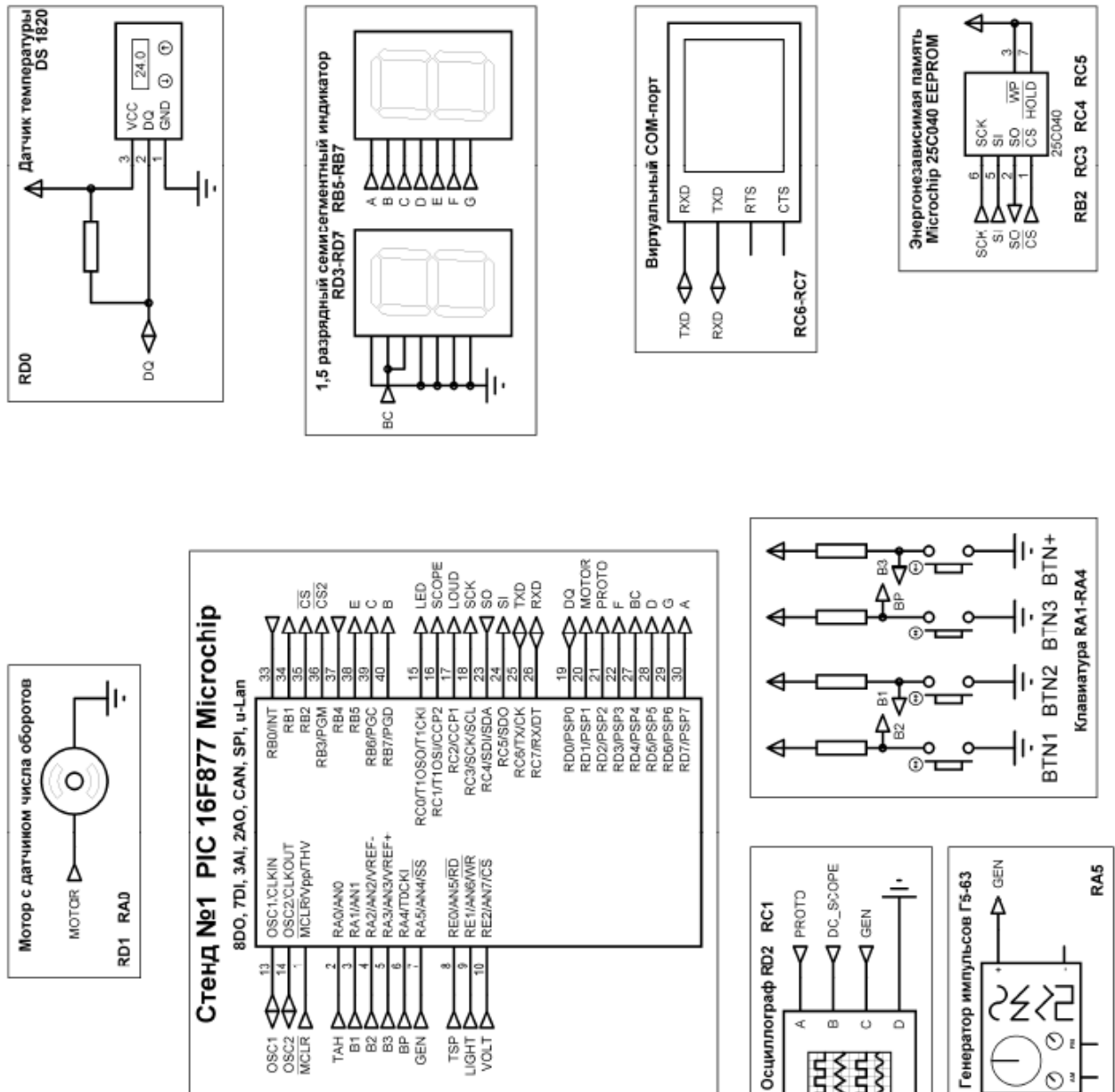
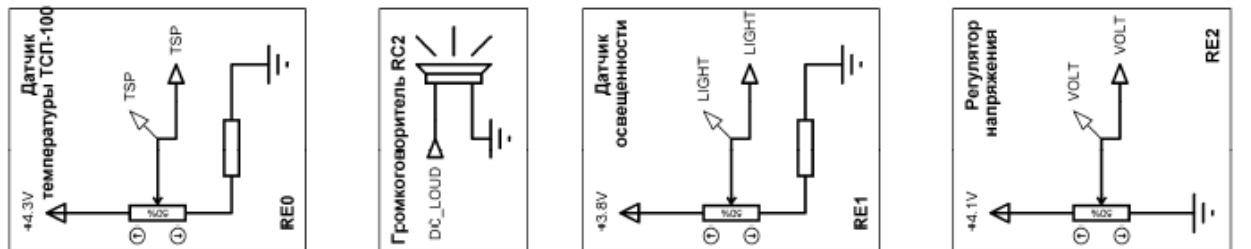


Рисунок – 19 Схема виртуального стенда №1

У моделі передбачена зв'язок із звуковою картою ПК, тому всі звукові сигнали будуть чути в колонках або навушниках.

- інтерфейс зв'язку з ПК. Модель реалізована у вигляді віртуального COM порту ПК, за допомогою якого на контакт RX моделі мікроконтролера можна відправляти окремі байти (або групи байт, але по одному) і приймати його відповіді (з виходу TX моделі мікроконтролера). Таким чином, емулюється робота ПК з модулем UART реального мікроконтролера. Візуально модель з'являється (тільки в режимі «Play») у вигляді вікна (терміналу) введення-виведення в якому відображаються символи або їх коди в каналі зв'язку.

При роботі з терміналом потрібно враховувати особливості:

1. Для передачі байта на контролер, на клавіатурі ПК натискається клавіша (крім деяких спеціальних клавіш, таких як Esc, Enter, F1 - F12 і деяких інших), при цьому відбувається автоматичне відправлення цього байта на мікроконтролер, але у вікні терміналу переданий символ (або його код) НЕ відображається. Щоб включити відображення символів, що вводять натисніть праву кнопку миші у вікні "Virtual Terminal" і виберіть пункт меню "Echo Typed Characters". Якщо Ви хочете бачити не символи, а їх коди, аналогічно виберіть "Hex Display Mode". При цьому відображення вводяться (і відправляються) символів відбувається без пробілів, а їх кодів - з пробілами. Відправляти можна тільки букви і цифри символи з клавіатури ПК. Відправка довільних байт, наприклад, 0xDF - неможлива.

2. Прийом здійснюється автоматично при включенні режиму «Play» (або покрокового виконання ПО). Відображення прийнятих символів відбувається БЕЗ поділу їх з відправляються.

Схема стенду № 2 включає в себе:

- мотор з датчиком числа його оборотів. Модель побудована з використанням вбудованого двигуна постійного струму, до якого спеціальним чином додані інтегратор і датчик числа обертів. Інтегратор перетворює ШІМ сигнал з контакту RC2 мікроконтролера в постійну напругу, що управляє двигуном. Поруч з двигуном розташований індикатор рівня постійної напруги, що надходить на двигун з модуля ШІМ контроллера (точніше з виходу інтегратора).

- кроковий двигун.

- ІК-приймач СДУ. В якості моделі фотоприймача використана група генераторів прямокутних імпульсів, з регульованою шпаруватістю (розстановкою) імпульсів. Включення генератора проводиться натисканням відповідної кнопки, що емулює кнопку пульта СДУ. У режимі «Play» кнопку треба тримати натиснутою не менше 2-х секунд, в режимі «F11» - її треба зафіксувати в натиснутому положенні. Одночасне натискання 2-х і більше кнопок одночасно - ЗАБОРОНЕНО, тому що призводить до непередбачуваного поведінки моделі. Про цю подію Вам додатково скаже рівні напруги у вигляді жовтих

квадратиків поруч з кнопками під час моделювання.

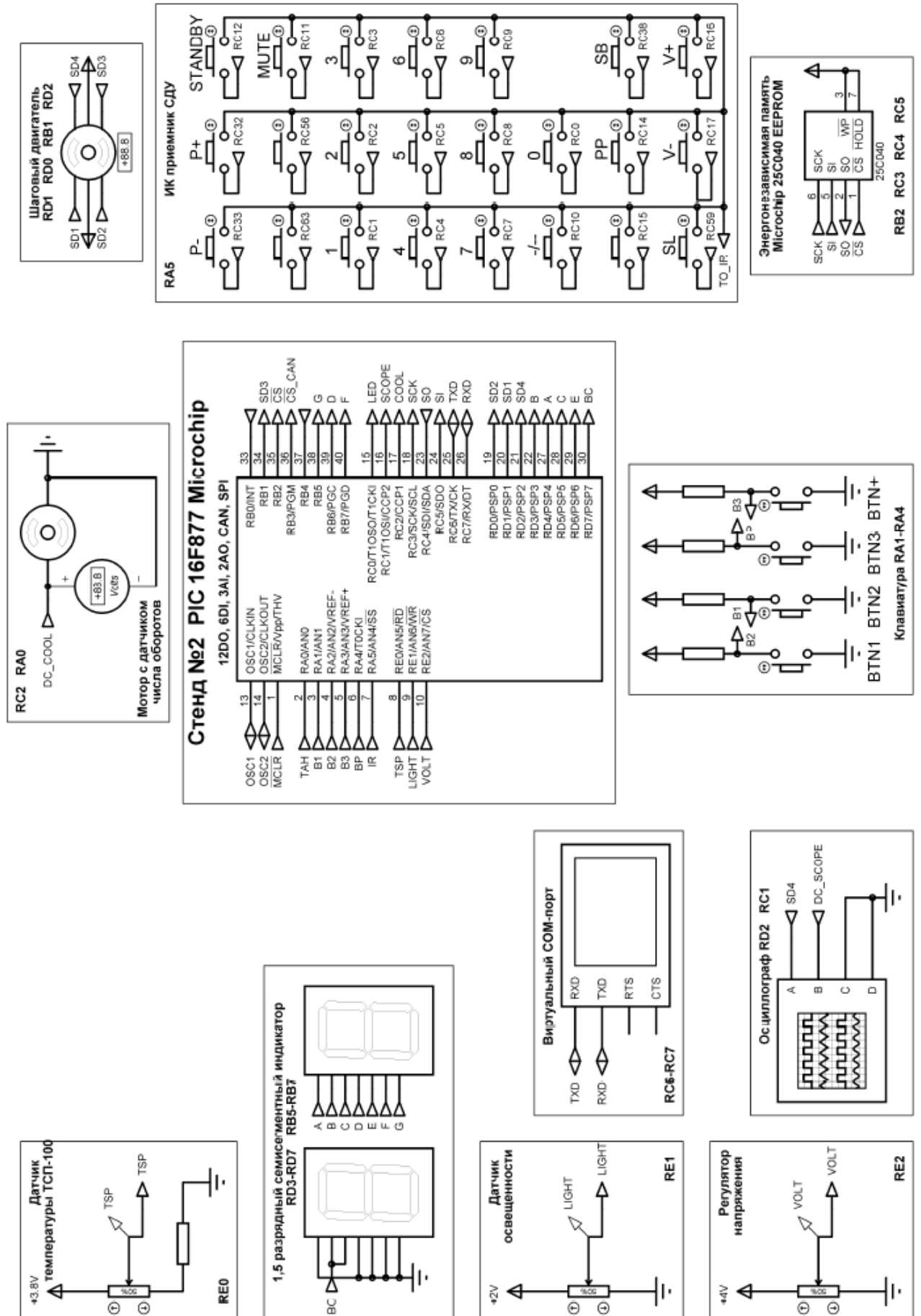


Рисунок – 20 Схеми віртуального стенда №2

При використанні моделі необхідно враховувати, що формування пачки займає час близько 25 000 мкс, що унеможливило використання налагодження по F11 для налагодження прийому.

- 1.5 розрядний семисегментний індикатор.
- інтерфейс зв'язку з ПК.
- незалежна пам'ять 25C040 об'ємом 512 байт з інтерфейсом SPI.
- клавіатура.
- аналоговий осцилограф.
- регулятор напруги.
- датчик освітленості.
- датчик температури t_{сп}-100.

3. Методика моделювання роботи PIC16F877 в середовищах MPLAB і Proteus з використанням віртуальних стендів.

3.1. Розробіть вихідний код ПО для мікроконтролера PIC 16F877 в середовищі MPLAB. Отладьте його на відсутність синтаксичних помилок. Відкомпілюйте проект (клавіша F10) і переконайтеся в наявності файлу з розширенням sof в папці проекту MPLAB. Він дозволить «бачити» вміст asm-файлу в середовищі Proteus. Після цього середу MPLAB не закривайте його.

3.2. Завантажте середу Proteus і через пункт меню «Open Design» завантажте схему потрібного віртуального стенда (один з файлів stand1.dsn - stand4.dsn). Збережіть схему, через пункт меню «File-Save as» (натиснувши «ОК» на попередження середовища), під іншим ім'ям в папку проекту MPLAB. Видаліть які не використовуються у Вашому завданні елементи віртуального стенда, тому що вони будуть сильно «гальмувати моделювання» і викликати зайві попереджувальні повідомлення середовища. Для цього виділіть мишею (з натиснутою лівою кнопкою) РАМКУ елемента і натисніть клавішу DEL. Скасування помилкового видалення - Edit / Undo або Ctrl-Z. Завантажте Вашу прошивку в модель мікроконтролера PIC 16F877. Для цього, двічі клацніть лівою кнопкою миші по зображенню мікроконтролера. У вікні «Edit Component» в поле «Program File» введіть ім'я sof-файлу (з папки з проектом MPLAB), отриманого в пункті 1 методики, і натисніть «ОК». Збережіть проект.

3.4. Проведіть покрокову налагодження Вашого ПО в середовищі Proteus. Для цього натисніть кнопку «Pause», при цьому з'явиться вікно з текстом вихідного asm- файлу і активної першої виконуваної командою. Налагодження проводиться в кілька етапів (особливо якщо це перша налагодження Вашого ПО), причому на практиці в залежності від складності ПО окремі етапи можна не виконувати. Однак, в загальному випадку налагодження може виконуватися в такій послідовності:

а. Для спостереження за станом внутрішніх регістрів мікроконтролера (крім портів, тому що їх стану відображаються середовищем моделювання автоматично) додайте спеціальне вікно «Watch Window». Для цього через пункт меню Debug / Watch викличте це вікно. Клацнувши правою кнопкою миші по

ньому, в меню оберіть пункт «Add Items (By Name) ...» або «Add Items (By Address) ...» для додавання імені або адреси регістра, за станом якого Ви хочете спостерігати в процесі моделювання. Перший пункт дозволяє (подвійним клацанням лівої кнопки миші) вибрати спеціальні регістри контролера, другий (натисканням кнопки «Add») - довільний РОН по його абсолютному, без урахування банку пам'яті, адресою. При цьому також можна вибрати зручний формат уявлення вмісту, наприклад, двійковий. Решта вікна моделювання можна закрити.

b. Встановіть постійні значення всіх вхідних сигналів. Для цього необхідні Вам кнопки зафіксуйте в натиснутому положенні, а движки змінних резисторів встановіть в положення, при якому імітується потрібне Вам значення напруг з датчиків. При зміні сигналів обов'язково враховуйте особливості моделей з п. 1.3.1.

c. Натискаючи клавішу «F11» (НЕ кнопку «Step» в середовищі Proteus) на клавіатурі, проведіть налагодження, спостерігаючи за станом периферійних пристроїв і змінюючи напруги з моделей датчиків.

d. При необхідності встановіть точки примусової зупинки виконання програми, так звані «BreakPoints». Це робиться в тих випадках, коли покрокова налагодження або займає великий час, наприклад, при виконанні процедур затримок, прийому контролером груп бітів, або частина коду не представляє інтересу для «Повільної» покрокової налагодження, наприклад, вже була налагоджена раніше. Для цього у вікні вихідного коду, на необхідній команді ДВА рази клацніть лівою кнопкою миші, при цьому зліва поруч з командою з'явиться червоний кружок. Видалення точки зупинки проводиться шляхом ДВОХ подвійних клацань лівою кнопкою миші по цій команді.

В обох випадках при натисканні кнопки «Play» буде автоматично (а відповідно і швидко) виконана частина програми, розташована ДО точки зупинки (команда на точці зупинки виконана НЕ БУДЕ). Після цього виконання ПЗ буде призупинено, і далі може бути продовжено Вами вже в режимі звичайної покрокової налагодження по F11, коли за кожне натискання виконується одна команда.

На практиці число точок зупинки може бути кілька (або не бути зовсім), а місце їх включення і спосіб чергування з режимом F11 ви можете вибирати самі. Основне, що потрібно пам'ятати - ПЗ буде виконано до точки зупинки і швидко, іншу частину коду можна перевіряти, комбінуючи «BreakPoint-и» і F11.

3.5. У разі виявлення помилок алгоритму або невідповідності функціонування ПО умов завдання, зупиніть моделювання кнопкою «Stop». Не закриваючи середовища Proteus, перейдіть до проекту в середовищі MPLAB, внесіть необхідні зміни в asm-файл і повторно Перекомпілюйте його, проведені Вами зміни будуть автоматично враховані середовищем Proteus. Після цього,

знову проведіть налагодження в середовищі Proteus. Тільки при повній відповідності функціонування Вашого ПЗ і умов завдання переходите до наступного пункту.

3.6. Проведіть остаточну перевірку функціонування Вашого ПЗ в режимі анімації. Для цього натисніть кнопку «Play» і перевірте реакцію контролера на всі можливі сигнали з датчиків в режимі безперервного виконання ПЗ. Для контролю станів внутрішніх регістрів мікроконтролера в цьому режимі також можна користуватися вікном «Watch Window» - воно не зникає при моделюванні. Переконавшись в правильності функціонування ПЗ, пред'явіть результати роботи викладачеві.

3.7. Збережіть проект в середовищах MPLAB, Proteus і переконайтеся в наявності файлу з розширенням hex, що є файлом «прошивки» мікроконтролера PIC 16F877. Цей файл підлягає заливанні в РПЗУ контролера за допомогою утиліти STest.

III. Заключна частина заняття. Перевірка і оцінювання виконаних завдань. Підведення підсумків лабораторного заняття, акцентування уваги на основних помилках при його виконанні.

3. Рекомендована література.

Основна література:

1. Грищук Ю.С. Мікропроцесорні пристрої: Навчальний посібник. – Харків: НТУ «ХПІ», 2007.– 280с.
2. Основи електротехніки, електроніки та мікропроцесорної техніки : навч. посіб. / В. Ф. Болюх, В. Г. Данько, Є. В. Гончаров; за ред. В. Г. Данька ; НТУ «ХПІ». – Харків : Планета-Прінт, 2019. – 248 с.

Допоміжна література:

1. Мікроконтролерні пристрої : навч. посіб. для студ. спец. «Мікро- та наноелектроніка» / О. С. Тонкошкур, І. В. Гомілко, О. В. Коваленко ; Дніпропетровський нац. ун-т ім. О. Гончара. – Д. : Вид-во ДНУ, 2011. – 264 с.
2. Мікропроцесорні пристрої: Методичні вказівки до виконання курсової роботи./ Ю.М.Трофімов К.: [ДЕТУТ], 2007.-191с.
3. Електронні системи: навчальний посібник / Й. Й. Білинський, К. В. Огороднік, М. Й. Юкиш. — Вінниця: ВНТУ, 2011. — 208 с.

Інформаційні ресурси в Інтернеті:

1. <https://uk.wikipedia.org/wiki/Мікроконтролер>
2. http://elprivod.nmu.org.ua/ua/interesting/what_is_mp_mc_plc.php
3. <http://vozm.ho.ua/MP/page61.html>
4. <https://life-prog.ru/ukr/arhitektura.php>