

МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ
Тернопільський національний технічний університет імені Івана Пулюя

Кафедра комп'ютерних систем та мереж

МЕТОДИЧНІ ВКАЗІВКИ ДЛЯ ВИКОНАННЯ ЛАБОРАТОРНИХ РОБІТ
з дисципліни «Комп'ютерна електроніка та схемотехніка» Частина 2
для студентів денної та заочної форми навчання
спеціальності 123 «Комп'ютерна інженерія»

Тернопіль 2020

Методичні вказівки для виконання лабораторних робіт з дисципліни «Комп'ютерна електроніка та схемотехніка» Частина 2 для студентів денної та заочної форми навчання розроблені у відповідності з навчальним планом спеціальності 123 «Комп'ютерна інженерія» / Уклад. Паламар А.М., Паламар М.І., Пастернак Ю.В., Стрембіцький М.О. – Тернопіль: ТНТУ, 2020. – 48 с.

Укладачі: Паламар А.М., проф., д.т.н. Паламар М.І., Пастернак Ю.В.,
к.т.н. Стрембіцький М.О.

Рецензент:

Відповідальний за випуск: зав. каф. комп'ютерних систем та мереж, к.т.н.,
доц. Осухівська Г.М.

Затверджено на засіданні кафедри комп'ютерних систем та мереж, протокол
№ ____ від « ____ » _____ 2020 р.

Методичні вказівки складені з врахуванням методичних розробок інших вищих закладів освіти, а також матеріалів літературних джерел, перелічених в списку.

ЗМІСТ

Вступ.....	4
Лабораторна робота №1. Дослідження портів вводу/виводу інформації в мікроконтролері. Робота з світлодіодною лінійкою.....	5
Лабораторна робота №2. Дослідження процесу статичного відображення інформації за допомогою семи-сегментного індикатора.....	17
Лабораторна робота №3. Дослідження методів динамічного відображення інформації в мікроконтролері за допомогою матриці світлодіодів.....	21
Лабораторна робота №4. Дослідження методів зчитування інформації в мікроконтролері за допомогою клавіатури.....	25
Лабораторна робота №5. Робота з рідкокристалічним дисплеєм.....	31
Лабораторна робота №6. Дослідження процесу аналого-цифрового перетворення в мікроконтролері на прикладі джойстика на змінних резисторах.....	40
Список використаних джерел.....	48

ВСТУП

Дисципліна «Комп'ютерна електроніка та схемотехніка» є складовою частиною підготовки студентів за спеціальністю 123 «Комп'ютерна інженерія».

Метою дисципліни «Комп'ютерна електроніка та схемотехніка» є формування знань у студентів з теоретичних методів аналізу і синтезу схем комп'ютерів (електронних обчислювальних машин) і засобів їхньої технічної реалізації, вивчення методів удосконалення архітектури комп'ютерів, якісного підвищення їхньої продуктивності та надійності.

Завданням вивчення дисципліни є ознайомлення студентів з основами цифрової та аналогової схемотехніки, вивчення теоретичних основ проектування пристроїв комп'ютерної схемотехніки, набуття практичних навичок застосування методів проектування пристроїв комп'ютерної схемотехніки.

Для закріплення знань та навичків студентам необхідно виконати низку лабораторних робіт. Лабораторна робота – невеликий науковий звіт, що узагальнює проведену студентом роботу, яку представляють для захисту викладачу. До лабораторних робіт пред'являється низка вимог, основною з яких є повний, вичерпний опис всієї проведеної роботи, що дозволяє оцінити отримані результати, міру виконання завдання та професійної підготовки студентів.

Звіт по лабораторній роботі друкується або пишеться студентом на одній стороні аркуша паперу формату 210x297 мм (A4). При цьому необхідно залишати поля: зліва – 25 мм, справа – 10 мм, поверх – 20 мм, знизу – 15 мм.

Звіт повинний включати наступні пункти:

1. Мета лабораторної роботи.
2. Теоретичні відомості.
3. Індивідуальне завдання (згідно варіанту) лабораторної роботи.
4. Основні етапи та результати виконання роботи.
5. Висновки.

ЛАБОРАТОРНА РОБОТА №1

ТЕМА: Дослідження портів вводу/виводу інформації в мікроконтролері. Робота з світлодіодною лінійкою.

МЕТА: Ознайомитись з процесом написання та відлагодження програми середовищі Keil. Вивчити можливості апаратної відладки програми. Навчитись завантажувати програму в мікроконтролер. Написати програму для роботи зі світлодіодною лінійкою.

1. Теоретичні відомості

1.1 Загальна технічна характеристика навчального стенда ST841/CPLD

З моменту появи перших мікроконтролерів їх складність постійно зростає за рахунок появи нових апаратних рішень та додавання нових команд для програмної підтримки нових функціональних вузлів, призначених для вирішення різноманітних задач.

Проектування вбудованих систем на сучасній елементній базі значно підвищує ефективність розробки за рахунок скорочення часу, мініатюризації, зниження споживаної потужності і збільшення швидкодії і надійності. На сьогоднішній день актуальною є задача придбання навиків розробки інформаційно-керуючих пристроїв на базі мікроконтролерів і програмованих логічних пристроїв, що дозволяють реалізувати алгоритми високої складності.

Розроблений стенд дозволяє ознайомитися із сучасною елементною базою, практично вивчати типові вузли приладів інформаційних вимірювальних систем, алгоритмів роботи, проводити відладку нових програм, організувати взаємодію датчиків з мікроконтролером, а також керувати виконуючими пристроями, такими як двигуни різних типів, електромагнітні виконавчі органи, та інші.

Стенд призначений для освоєння архітектури та методів проектування інформаційно-керуючих систем, систем збору і оброблення інформації на базі найпоширеніших мікроконтролерів сімейства MCS-51/52, а також ПЛІС фірм Altera або Xilinx.

Стенд може бути використаний для:

- вивчення вузлів мікроконтролерних інформаційно-вимірювальних систем, промислових інтерфейсів, пристроїв вводу-виводу;
- виконання курсових, дипломних та науково-дослідних робіт.
- розробки і відлагодження апаратно-орієнтованого програмного забезпечення;
- взаємодії із аналоговими та цифровими первинними перетворювачами та виконавчими органами;

Функціональні можливості:

- мікроконтролер ADuC841 (однотактове ядро, 62 кбайт флеш-пам'яті програм, 4 кбайт флеш-пам'яті даних, 2304 байт оперативної пам'яті, апаратна підтримка інтерфейсів UART, I2C, SPI, восьмиканальний 12-розрядний АЦП, 320кГц, два 12-розрядні цифро-аналогові перетворювачі, джерело опорної напруги, датчик температури);
- програмована логічна матриця CPLD: EPM3032, EPM3064 (Altera), або XCR3032XL, XCR3064XL (Xilinx);
- інвертуючий, неінвертуючий, диференційний та інструментальний підсилювачі;
- пристрої вводу;

- матрична клавіатура,
- дискретні кнопки,
- джойстик-потенціометр,
- механічний енкодер;
- пристрої відображення:
 - лінійка світлодіодів,
 - 4 семисигментні індикатори,
 - 1 матричний індикатор,
 - індикатор на рідких кристалах;
- звукова підсистема:
 - звуковий рст кодек (TLV320AIC1109),
 - підсилювач звукової частоти TDA7052,
 - динамік, мікрофонний вхід, вихід на телефони;
- шини даних:
 - паралельна 8-бітна шина даних;
 - SPI
 - I2C
 - RS232
 - RS485
- цифрова периферія:
 - годинник реального часу з I2C інтерфейсом DS1307-33;
 - 8 Мбіт флеш-пам'ять із SPI інтерфейсом,
 - генератор прямокутних імпульсів (NE555),
 - генератор тактової частоти;
- виконавчі пристрої:
 - драйвери крокових двигунів (8 ключів);
 - ШІМ-регулятор;
 - Н-міст для двигуна постійного струму.

1.2 Будова контролера

Як процесор використовується однокристальний мікроконтролер ADuC841 з такими характеристиками:

- Однотактне 20MIPS ядро архітектури 8052;
- Швидкісний 12-розрядний АЦП;
- Два 12-розрядні ЦАП;
- 62 кбайт вбудованої пам'яті програм;
- 2 кбайти вбудованої пам'яті даних;

- UART, I2C, SPI інтерфейси;
- Програмування та можливість відладки через інтерфейс UART.

Для роботи із стендом необхідне знання архітектури контролера, його регістрів та периферії. Проведемо короткий огляд найбільш важливих вузлів.

ADuC841 є функціонально завершеним контролером інтелектуальних датчиків і включає в себе високоякісний багатоканальний АЦП із самокалібруванням, два ЦАП, і швидкий (20МГц) з однократним виконанням команд 8-ми розрядний програмований мікроконтролер з системою команд МК 8051 на одному кристалі. Ядром МК є контролер 8052, що забезпечує пікову продуктивність до 20 MIPS. На кристалі розміщено 62Кбайт Flash пам'яті програм, 4Кбайт Flash пам'яті даних, 256 байт пам'яті з довільним доступом (RAM) і 2Кбайт розширеної пам'яті з довільним доступом (XRAM).

В склад ADuC841 входять також додаткові аналогові пристрої: два 12-розрядні ЦАП, монітор напруги живлення і джерелом опорної напруги. Додатковими цифровими пристроями є: два 16-розрядні Σ - Δ ЦАП, два 16-розрядні широтно-імпульсні модулятори, сторожовий таймер, лічильник часових інтервалів, три таймери-лічильники, і три порти послідовного вводу-виводу (SPI, I2C, і UART). Заводська прошивка контролера підтримує завантаження програмного забезпечення через послідовний порт UART, а також емуляцію через один контакт пристрою – EA. Далі приведена функціональна схема ADuC841.

1.3 Периферійні пристрої стенду

Програмована логічна інтегральна схема.

На платі стенду розміщена панелька для встановлення ПЛІС типу EPM3032ALC44, EPM3064ALC44 виробництва Altera а також сумісних з ними XCR3032XL, XCR3064XL виробництва XILINX. Для завантаження програмного забезпечення в ПЛІС та його відладки призначений роз'єм інтерфейсу JTAG. ПЛІС працює незалежно від МК і має свої периферійні вузли.

Блок операційних підсилювачів для аналогових сигналів.

Блок призначений для підсилення вхідних сигналів, що подаються на входи АЦП МК і сигналів з виходів ЦАП.

Звуковий кодек.

Звуковий кодек TLV320AIC1109 призначений для перетворення аналогового сигналу звукової частоти в стиснений цифровий формат для подальшої передачі, а також для здійснення зворотного перетворення цифрових даних в звуковий сигнал. Для підключення мікрофона і телефона передбачено роз'єми. Сигнал з виходу кодека може подаватись на розміщені на платі підсилювач і динамік.

Джойстик та механічний енкодер.

На платі установлений потенціометричний джойстик з двома осями і кнопкою. Сигнал з виходів джойстика подається на входи 6 та 7 АЦП МК.

Механічний енкодер призначений для перетворення кутового переміщення в електричний сигнал. Дані пристрої дозволяють проектувати системи з необхідністю ручного вводу інформації (наприклад, маніпулятори, прилади, спецтехніка, блоки керування і т.п.).

Цифрова клавіатура.

Матрична клавіатура застосовується при побудові вузлів вводу даних в інформаційну систему.

Кнопки переривань та скидання процесора.

Кнопки переривань призначені для генерації сигналів переривання процесора. Кнопка скидання призначена для скидання процесора під час програмування або для перезапуску існуючої програми.

Рідкокристалічний дисплей.

На платі установлений знакосинтезуючий рідкокристалічний 4x10 дисплей, підключений до шини даних по чотирибітній схемі. Дисплей має вбудовану підсвітку та регулятор контрасту.

Семисегментний чотиризначний та матричний індикатори, світлодіодна лінійка.

На основі світлодіодів, семисегментних та матричних індикаторів будуються підсистеми відображення інформації. Схема включення семисегментного індикатора дозволяє здійснювати статичну індикацію, матричного – динамічну, знакосинтезуючу.

Інтерфейси стандартів RS485, Ethernet, I2C, SPI, I-wire та системний інтерфейс.

На платі розміщено драйвери інтерфейсів UART, який побудований на мікросхемі CP2102, що дозволяє під'єднувати стенд до USB входу компютера, RS485, а також Ethernet-драйвер, які узгоджують логічні рівні даних стандартів та TTL рівень а також роз'єми для підключення кабелів чи шлейфів периферійних пристроїв. Системний інтерфейс включає шину даних, адресні лінії, лінії UART, I2C, живлення та загальний провід.

Годинник реального часу DS1307.

Використовується для генерації точних секундних імпульсів, а також як годинник і календар.

Восьмиканальний драйвер крокових двигунів.

Драйвер дозволяє керувати двома уніполярними двигунами з чотирма обмотками невеликої потужності (із споживаним струмом до 2А).

Контрольні точки для підключення осцилографа.

Контрольні точки розміщені у важливих функціональних вузлах і дозволяють проводити візуальне спостереження форми сигналу за допомогою осцилографа.

Генератор імпульсів.

Генератор прямокутних імпульсів зі зноу частоти генерації побудований на основі таймера 555 і може використовуватись для генерації контрольних імпульсів.

Стабілізатори живлення.

На платі встановлено стабілізатори напруг 5В, 3.3В.

1.4 Створення проекту в середовищі Keil

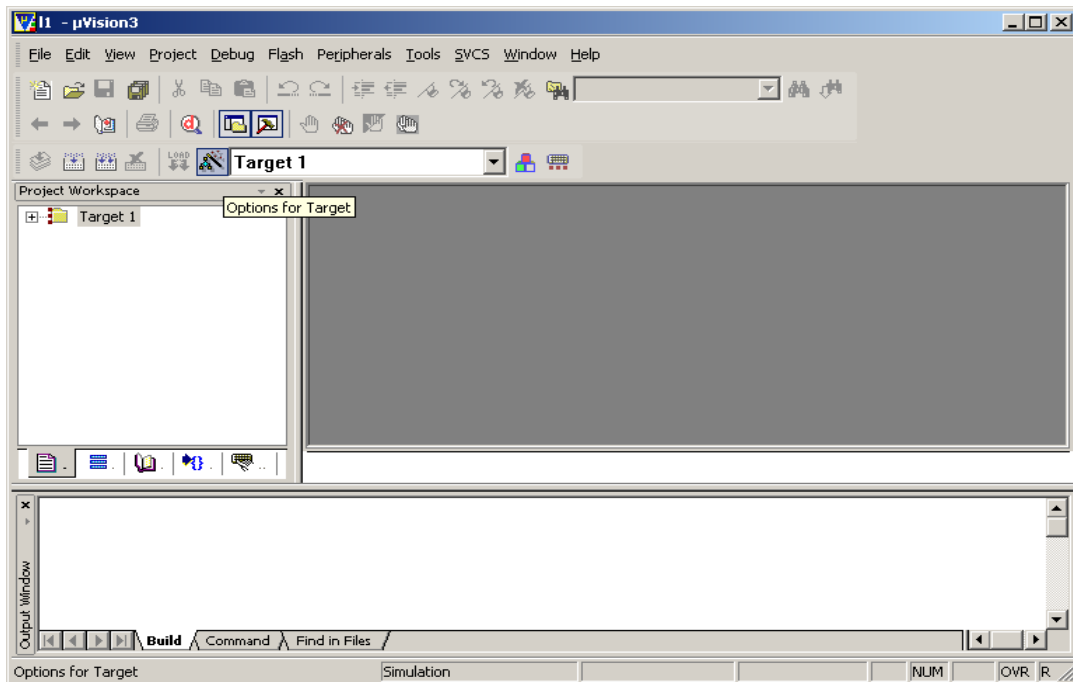


Рисунок 1 – Вибір створення прив'язки

У вікні, що появилось, на закладці Target необхідно вказати частоту кварцового резонатора, що використовується – 11.0592 МГц. На закладці Output слід відмітити пункт Create HEX File.

На закладці Utilities відмітити пункт Use Target Driver for Flash Programming, із списку слід вибрати пункт ADI Monitor Driver, (рисунок 1.2), натиснути кнопку Settings, встановити відповідний порт для зв'язку (COM1), швидкість лінії 9600, пункти RTS і DTR встановити у значення Inactive.

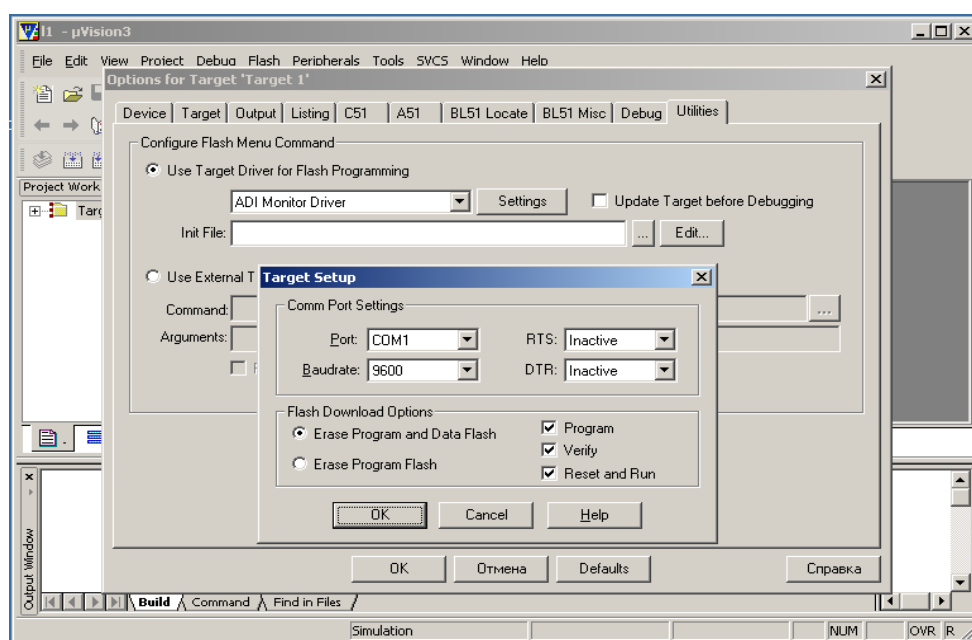


Рисунок 2 – Настроювання прив'язки

Після налаштування прив'язки потрібно створити новий файл – меню File -> New, і зберегти його із розширенням .asm або .a51. У вікні Project Workspace розкрити групу Target 1, перейти на групу Source Group 1, викликати контекстне меню, клацнувши правою кнопкою миші по групі Source Group 1 і вибрати пункт Add Files to Source Group 1. У діалогові вікні, що з'явилося, вказати створений файл, встановивши тип файлів у значення Asm Source Files.

Набрати текст програми, натиснути кнопку Rebuild all target files.

Для того, щоб завантажити відкомпільований файл у стенд необхідно:

Для стендів 1го типу

1. Замкнути перемичку JP13.
2. Натиснути кнопку RESET (SW2).
3. Вибрати команду Flash -> Download.

Для стендів 2го типу

4. Замкнути перемичку J6.
5. Перемикач SW8 перевести в положення замкнуто (у верхньому положенні).
6. Натиснути кнопку RESET (SW9).
7. Вибрати команду Flash -> Download.

Внаслідок таких дій програма завантажиться в пам'ять програм мікроконтролера і запуститься на виконання.

1.5 Схема для роботи з світлодіодною лінійкою

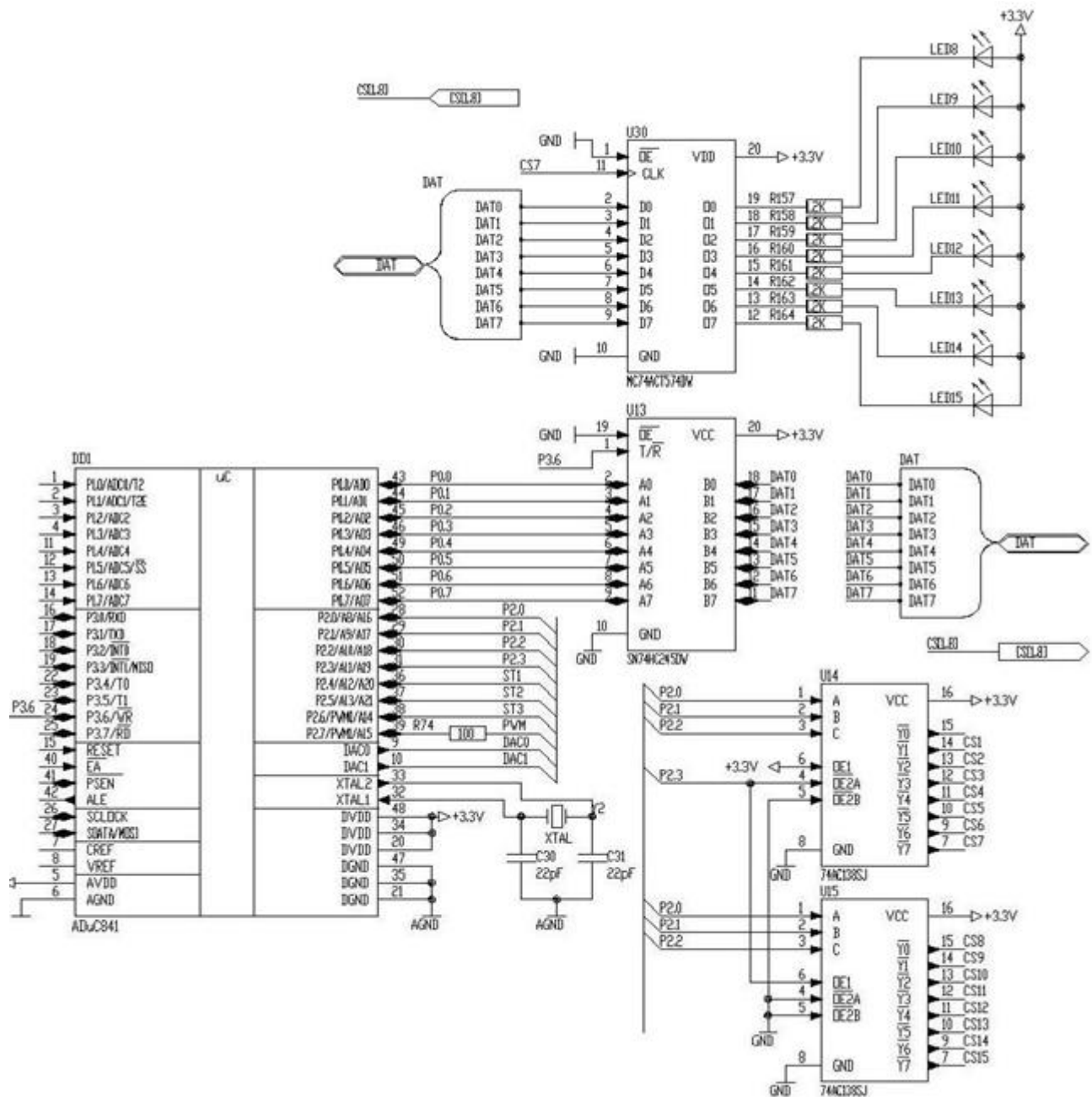


Рисунок 3 – Схема для роботи з світлодіодною лінійкою

Світлодіодна лінійка може бути використана як бінарний 8-розрядний індикатор. Адреса регістра лінійки - 07h. Світлодіоди увімкнені анодами до +5В, катодами – до виходів регістра. Для погашення світлодіодів в регістр потрібно записати число FFh. Щоб засвітити світлодіод, потрібно відповідний біт регістра встановити рівний логічному 0.

Для адресації використовується молодша тетрада порту P2 мікроконтролера (Рисунок 2.1). На даному рисунку живлення та інші порти контролера умовно показані непідключеними. Біти 0, 1, 2 подаються на входи дешифраторів А, В, С, а біт 3 – на інвертований вхід ОЕА (Output Enable A) першого і одночасно на неінвертований вхід OE1 другого. Таким чином, стан порту P2.3 визначає, який дешифратор буде працювати, тобто при значенні P2.3=0 працює верхній (по схемі) дешифратор, а при P2.3=1 – нижній. Вихід дешифратора Y0 не задіяний,

тому що при значенні молодшої тетради порту P2, рівній 0000b, на цьому виході буде логічний нуль (виходи дешифраторів інвертовані). Напрямок роботи буфера визначається значенням на його вході DIR. Це значення задається портом мікроконтролера P3.6. Для роботи буфера на передачу від контролера до шини значення сигналу DIR повинно бути рівним логічній 1, а для роботи на прийом від шини до контролера – логічному 0.

Покажемо часові діаграми роботи контролера із периферійними пристроями.

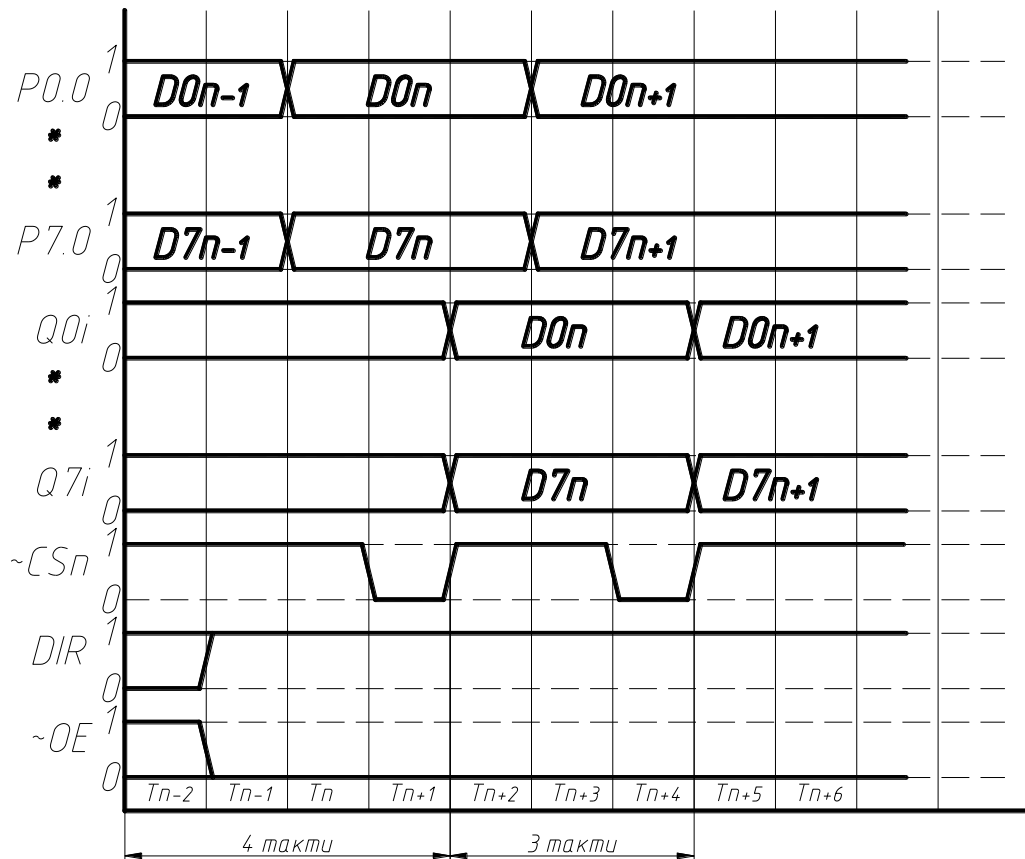


Рисунок 4 – Часова діаграма запису двох чисел в регістр периферійного пристрою

Число фіксується регістром (периферійним пристроєм) по зростаючому фронту сигналу $\sim CS$. Послідовність роботи контролера така:

1. Встановлюється напрям роботи буфера ($DIR=1$) і увімкнення його і дешифраторів ($\sim OE=0$);
2. В порт P0 записується число, яке потрібно записати;
3. В порт P2.0-P2.3 записується адрес периферійного пристрою (регістра);
4. В порт P2.0-P2.3 записується число 0000b.

Таким чином запис числа проходить за 4 такти.

Якщо напрям роботи буфера був встановлений і дешифратори увімкнені, то процес проходить за 3 такти.

При зчитуванні контролером із шини даних потрібно встановити напрям роботи буфера на читання ($DIR=1$), дати строб $\sim CS$ для відповідного адресу, і зчитати стан порту P0. Слід пам'ятати, що з регістрів, які розміщені на платі стенду, зчитування неможливе, але зчитувати можна із системного роз'єму.

1.6 Приклад програми для роботи з світлодіодною лінійкою

На світлодіодній лінійці засвітити мигаючий світлодіод.

```
// address of LEDs 0x07
// define DATA transfer register
DAT EQU R0
// define ADDRESS transfer register
ADDR EQU R1
// define temporary0 register
Temp0 EQU R2
// Address for PUSH and POP commands
_Temp0 EQU 0x02
//Interrupt vector
ORG 0x0000
JMP Start

ORG 0x0033
START:
    MOV DAT,#11111110b
    MOV ADDR, #0x07
    CALL write
    CALL LDelay
    CALL LDelay
    CALL LDelay
    CALL LDelay
    CALL LDelay
    CALL LDelay
    CALL LDelay
    MOV DAT,#11111111b
    MOV ADDR, #0x07
    CALL write
    CALL LDelay
    CALL LDelay
    CALL LDelay
    CALL LDelay
    CALL LDelay
    CALL LDelay
    CALL LDelay
    JMP START
// Write to periferal device subroutine
```

```

write:  setb P3.6           //Set Data bus buffer to TX
        mov P0, DAT         //moving data to bus buffer
        mov P2, ADDR //set peripherals address
        nop                 //wait
        nop
        mov P2, #0x00 //clock pulse for device latch
ret                //exit from subroutine

LDelay:           // Підпрограма затримки тривалістю 17.9 мс.
        MOV Temp0, #0xFF
LD1:
        DEC Temp0
        PUSH _Temp0
            MOV Temp0, #0xFF
            DJNZ Temp0, $
        POP _Temp0
        CJNE Temp0, #0x00, LD1
RET
END

```

2. Хід роботи

1. Вивчити структурну схему стенду, спосіб адресації до периферійних вузлів, призначення окремих елементів схеми, ознайомитись із структурою та системою команд OEOM ADuC841.
2. Ознайомитись із середовищем Keil, навчитись створювати новий проект, ознайомитись із методами програмної відладки.
3. Розробити алгоритм виконання індивідуального завдання.
4. Розробити програму відповідно до варіанту і скомпілювати її.
5. Завантажити програму в стенд, переконатись в правильності її роботи, при негативному результаті виявити допущені помилки і виправити їх. Повторити завантаження програми в стенд.

Варіанти індивідуальних завдань

№	Текст індивідуального завдання
1	На світлодіодній лінійці засвітити 4 світлодіоди через один починаючи з нижнього на 1с., після цього погасити усі світлодіоди на 1 с., повторити процедуру спочатку.
2	На світлодіодній лінійці засвітити усі світлодіоди на 1 с., після цього погасити усі світлодіоди на 1 с., повторити процедуру спочатку.
3	На світлодіодній лінійці реалізувати біжучу точку у вигляді засвіченого світлодіода в напрямку від нижнього світлодіода до верхнього, швидкість переміщення 1 світлодіода в 2 секунди.
4	На світлодіодній лінійці реалізувати біжучу точку у вигляді погашеного світлодіода в

	напрямоку від нижнього світлодіода до верхнього, швидкість переміщення 1 світлодіод в секунду.
5	На світлодіодній лінійці засвітити 4 світлодіоди через один починаючи з нижнього на 1с., після цього засвітити 4 світлодіоди через один але вже починаючи з верхнього також на 1 с., повторити процедуру спочатку.
6	На світлодіодній лінійці реалізувати режим наростання кількості засвічених світло діодів від нижнього до верхнього з інтервалом 1 с.: в початковому стані всі світлодіоди погашені, спочатку засвічується нижній світлодіод, через 1 с. засвічується два нижніх, потім три і т.д., після того як засвітяться усі світлодіоди процедуру повторити спочатку.
7	На світлодіодній лінійці реалізувати режим наростання кількості погашених світло діодів від нижнього до верхнього з інтервалом 1 с.: в початковому стані всі світлодіоди засвічені, спочатку гаситься нижній світлодіод, через 1 с. гасяться два нижніх, потім три і т.д., після того як погасяться усі світлодіоди процедуру повторити спочатку.
8	На світлодіодній лінійці реалізувати режим наростання кількості засвічених світло діодів від верхнього до нижнього з інтервалом 1 с.: в початковому стані всі світлодіоди погашені, спочатку засвічується верхній світлодіод, через 1 с. засвічується два верхніх, потім три і т.д., після того як засвітяться усі світлодіоди процедуру повторити спочатку.
9	На світлодіодній лінійці реалізувати режим наростання кількості погашених світло діодів від верхнього до нижнього з інтервалом 1 с.: в початковому стані всі світлодіоди засвічені, спочатку гаситься верхній світлодіод, через 1 с. гасяться два верхніх, потім три і т.д., після того як погасяться усі світлодіоди процедуру повторити спочатку.
10	На світлодіодній лінійці засвітити 4 світлодіоди через один починаючи з верхнього на 1с., після цього погасити усі світлодіоди на 1 с., повторити процедуру спочатку.
11	На світлодіодній лінійці погасити усі світлодіоди, після цього засвітити усі світлодіоди. Цю процедуру повторювати з частотою 0,5 Гц.
12	На світлодіодній лінійці реалізувати біжучу точку у вигляді засвіченого світлодіода в напрямоку від верхнього світлодіода до нижнього, швидкість переміщення 1 світлодіод в секунду.
13	На світлодіодній лінійці реалізувати біжучу точку у вигляді погашеного світлодіода в напрямоку від верхнього світлодіода до нижнього, швидкість переміщення 1 світлодіод в 2 секунди.
14	На світлодіодній лінійці засвітити 3 нижніх світлодіоди на 2с., після цього засвітити 3 верхніх світлодіоди також на 2 с., повторити процедуру спочатку.
15	На світлодіодній лінійці реалізувати режим наростання кількості засвічених світло діодів від нижнього до верхнього з інтервалом 2 с.: в початковому стані всі світлодіоди погашені, спочатку засвічується два нижніх світлодіоди, через 2 с. засвічується чотири нижніх, потім шість і т.д., після того як засвітяться усі світлодіоди процедуру повторити спочатку.
16	На світлодіодній лінійці реалізувати режим наростання кількості погашених світло діодів від нижнього до верхнього з інтервалом 2 с.: в початковому стані всі світлодіоди засвічені, спочатку гаситься два нижніх світлодіоди, через 2 с. гасяться чотири нижніх, потім шість і т.д., після того як погасяться усі світлодіоди процедуру повторити спочатку.
17	На світлодіодній лінійці реалізувати режим наростання кількості засвічених світло діодів від верхнього до нижнього з інтервалом 2 с.: в початковому стані всі світлодіоди погашені, спочатку два верхніх світлодіоди, через 2 с. засвічується чотири верхніх, потім шість і т.д., після того як засвітяться усі світлодіоди процедуру повторити спочатку.
18	На світлодіодній лінійці реалізувати режим наростання кількості погашених світло діодів від верхнього до нижнього з інтервалом 2 с.: в початковому стані всі світлодіоди

	<p>засвічені, спочатку гаситься два верхніх світлодіоди, через 2 с. гасяться чотири верхніх, потім шість і т.д., після того як погасяться усі світлодіоди процедуру повторити спочатку.</p>
--	---

ЛАБОРАТОРНА РОБОТА №2

ТЕМА: Дослідження процесу статичного відображення інформації за допомогою семи сегментного індикатора.

МЕТА: Вивчити метод статичного відображення інформації на прикладі семисегментного індикатора.

1. Теоретичні відомості

1.1 Семисегментний індикатор

До вузла статичної індикації входять: 4 регістри-фіксатори, струмообмежуючі резистори і чотирьохцифровий світлодіодний семисегментний дисплей. Схема їх увімкнення показана на рис. 2.1. Зміст статичної індикації полягає у тому, щоб кожен індикатор постійно засвічувався від свого джерела інформації.

Число, яке виводиться на індикатор, фіксується відповідним регістром. Після запису числа в регістр, шина даних може приймати довільне значення, воно не буде впливати на значення, яке виводиться на індикаторі. Хоча застосування статичної індикації і потребує відносно великих апаратних затрат, зате відпадає потреба в постійному переключенні індикаторів, як у випадку динамічної індикації, спрощується програмне забезпечення, контролер звільняється від необхідності постійної почергової передачі чисел на кожен індикатор, як у випадку динамічної індикації.

Коли потрібно засвітити сегмент індикатора, слід встановити відповідний біт регістра, рівний логічному нулю. Для того, щоб погасити відповідний індикатор, у регістр потрібно записати число 0FFh. Оскільки індикатори підключені без дешифраторів, то процес дешифрування слід проводити програмно. Програмне дешифрування дозволяє виводити на індикатор не тільки числа, а й деякі букви та інші символи.

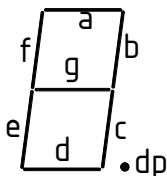


Рисунок 1 – Схема позначення сегментів індикатора

Таблиця 1.1 – Дешифрування двійкового коду для семисегментного індикатора

Сегмент/цифра	1	2	3	4	5	6	7	8	9	0
Dat0 (a)	1	0	0	1	0	0	0	0	0	0
Dat1 (b)	0	0	0	0	1	1	0	0	0	0
Dat2 (c)	0	1	0	0	0	0	0	0	0	0
Dat3 (d)	1	0	0	1	0	0	1	0	0	0
Dat4 (e)	1	0	1	1	1	0	1	0	1	0
Dat5 (f)	1	1	1	0	0	0	1	0	0	0
Dat6 (g)	1	0	0	0	0	0	1	0	0	1
Dat7 (dp)	1	1	1	1	1	1	1	1	1	1

Для засвічування десяткової коми (dp, decimal point) потрібно встановити біт 7 дешифратора, рівний 0.

Адреси цифр дисплея розміщені наступним чином:

- для запису в перший (справа наліво) розряд адрес рівний 01h;
- для запису в другий розряд P2 адрес рівний 02h;
- для запису в третій розряд P2 адрес рівний 03h;
- для запису в четвертий розряд P2 адрес рівний 04h.

1.2 Схема для роботи з семисегментним індикатором

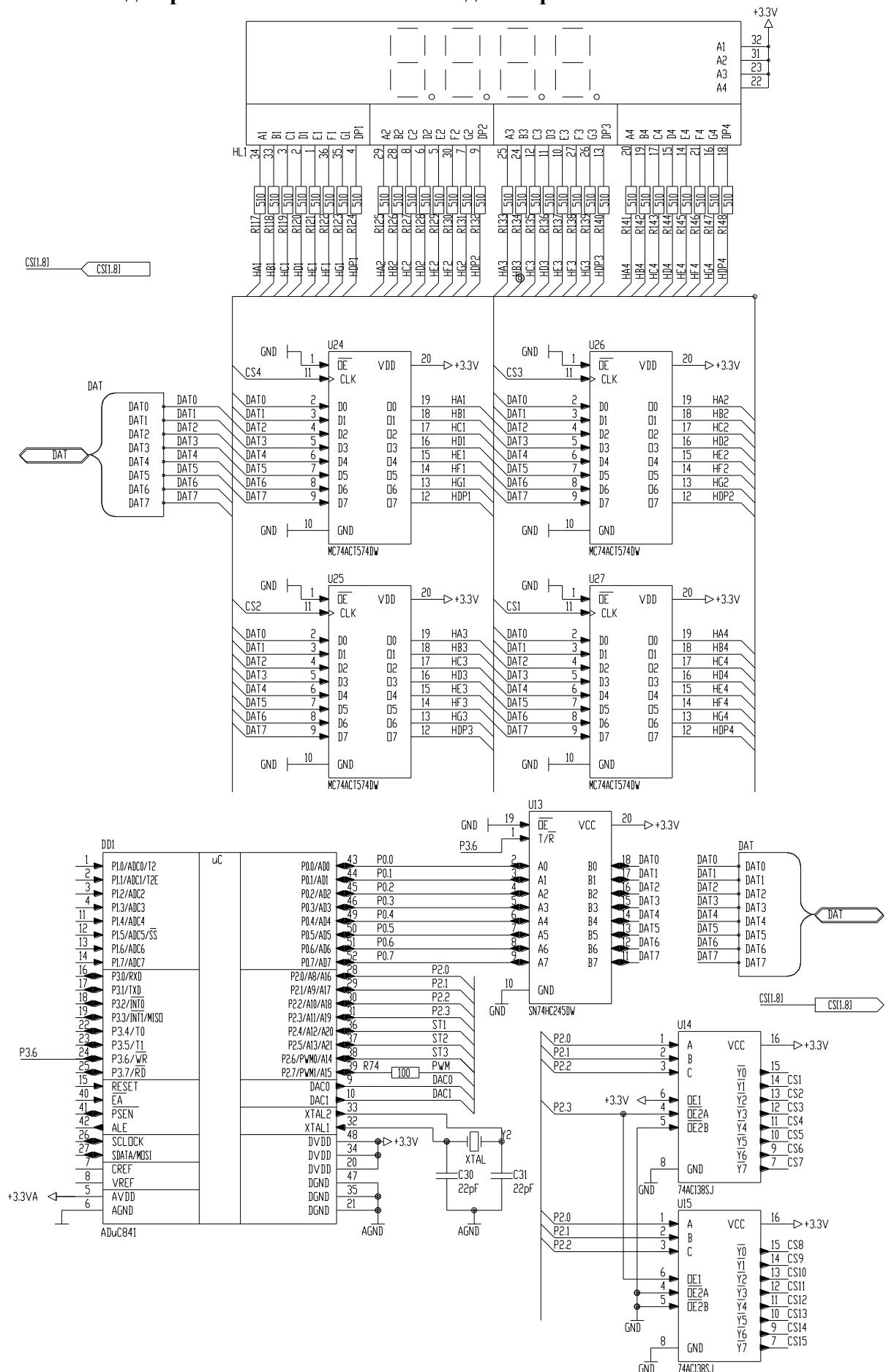


Рисунок 2 – Схема для роботи семисегментним індикатором

1.3 Приклад програми для роботи з семисегментним індикатором

; На семисегментному індикаторі по чергово появляються цифри '1', '9', '8', '8'.

```
dat    EQU    R0
adr    EQU    R1
Temp1  EQU    R2
Temp2  EQU    R3
Temp3  EQU    R4
```

```
Segm   EQU 0x02
```

CSEG

ORG 0x0000

JMP run

ORG 0x0033

run:

```
    call off
    call ldelay
    call ldelay
    mov dat, #11001111b
    mov adr, #00000100b
    call strob
    call ldelay
    mov dat, #10010000b
    mov adr, #00000011b
    call strob
    call ldelay
    mov dat, #10000000b
    mov adr, #00000010b
    call strob
    call ldelay
    mov dat, #10000000b
    mov adr, #00000001b
    call strob
    call ldelay

    jmp run
```

strob:

```
    setb P3.6
    mov P0, dat
    mov P2, adr
    mov P2, #0x00
```

ret

off:

```
    mov dat, #11111111b
    mov adr, #00000001b
    call strob
    mov dat, #11111111b
    mov adr, #00000010b
    call strob
    mov dat, #11111111b
    mov adr, #00000011b
    call strob
    mov dat, #11111111b
    mov adr, #00000100b
    call strob
```

ret

Delay:

```
    mov Temp1, #0xFF
```

loop:

```

    dec Temp1
    mov Temp2, #0xFF
    djnz Temp2, $
    cjne Temp1, #0x00, loop
ret
Ldelay:
    call delay
    call delay
    call delay
    call delay
    call delay
    call delay
    call delay
    call delay
    call delay
    call delay
    call delay
ret
END

```

2. Хід роботи

6. Вивчити принцип статичного методу відображення інформації на прикладі семисегментного індикатора.
7. Розробити алгоритм індивідуального завдання до початку заняття.
8. Розробити програму і скомпілювати її.
9. Завантажити програму в стенд, переконатись в правильності її роботи, при негативному результаті виявити допущені помилки і виправити їх. Повторити завантаження програми в стенд.

Варіанти індивідуальних завдань

№	Текст індивідуального завдання
1-18	На семисегментному індикаторі на 1 секунду вивести номер свого варіанту, потім на 2 секунди вивести дату і місяць свого народження, розділені крапкою, після цього на 3 секунди вивести рік свого народження. Після цього процедуру повторити спочатку.

ЛАБОРАТОРНА РОБОТА №3

ТЕМА: Дослідження методів динамічного відображення інформації в мікроконтролері за допомогою матриці світлодіодів.

МЕТА: Вивчити метод динамічного відображення інформації на прикладі матриці світлодіодів.

1. Теоретичні відомості

1.1 Схема для роботи з матрицею світлодіодів

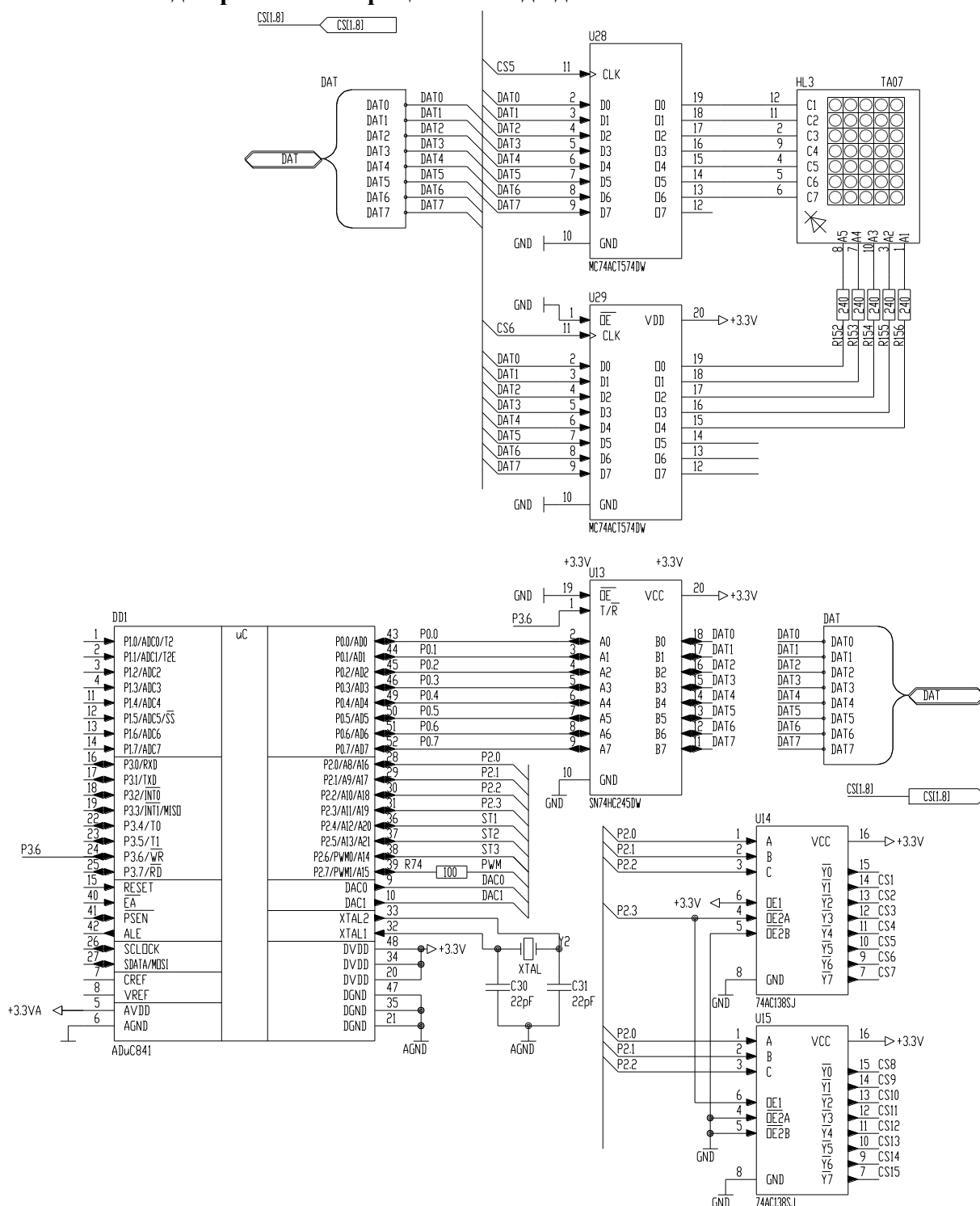


Рисунок 1 – Схема для роботи з матрицею світлодіодів

1.2 Динамічна індикація

Динамічна знакосинтезуюча індикація реалізована на світлодіодній матриці 5x7 і дозволяє вивести довільний символ, організувати біжучу стрічку та ін. Зміст динамічної індикації полягає у тому, щоби по черзі засвічувати елементи індикатора з такою частотою, при якій око за рахунок деякої інерційності сприйняття не спроможне помітити мерехтіння. Максимум частоти переключення обмежений необхідністю забезпечити такий час світіння елемента, при якому значно не зменшується його яскравість. Елементи індикатора засвічуються записом логічних нулів у відповідні розряди регістру рядків, який підключений до катодів індикатора (на схемі – U28, адрес регістра 05h) і логічних одиниць у розряди регістру стовбців, підключеного до анодів (на схемі – U29, адрес регістра 06h). Є зміст виводити знак на індикацію не по елементах (точках), а по рядках. В такому випадку, для того, щоб засвітити у верхньому рядку першу і четверту точку, потрібно в регістр U28 записати число 11111110b, а в регістр U29 – число 00001001b.

1.3 Приклад програми для роботи з матрицею світлодіодів

; Засвітити на матриці світлодіодів літеру “S”.

```
dat    Equ R0
adr    Equ R1
Temp1  Equ R2
Temp2  Equ R3
stec   Equ 0x02
```

```
Org    0x000
```

```
run:
```

```
    mov dat,#10111110b
    mov adr,#00000101b
    call writ
;    call delay
    mov dat,#00001110b
    mov adr,#00000110b
    call writ
;    call delay
    call off

    mov dat,#11011101b
    mov adr,#00000101b
    call writ
;    call delay
    mov dat,#00010001b
    mov adr,#00000110b
    call writ
;    call delay
    call off

    mov dat,#11111011b
    mov adr,#00000101b
    call writ
;    call delay
    mov dat,#00010000b
    mov adr,#00000110b
    call writ
;    call delay
    call off

    mov dat,#11101111b
    mov adr,#00000101b
    call writ
```

```

;      call delay
      mov dat,#00000001b
      mov adr,#00000110b
      call writ
;      call delay
      call off

      mov dat,#11110111b
      mov adr,#00000101b
      call writ
;      call delay
      mov dat,#00001110b
      mov adr,#00000110b
      call writ
;      call delay
      call off

```

jmp run

Writ:

```

      setb P3.6
      mov P0, dat
      mov P2, adr
      nop
      nop
      mov P2, #0x00

```

ret

Delay:

```

      mov Temp1, #0xFF
loop:  dec Temp1
      mov Temp2, #0xFF
      djnz Temp2, $
      cjne Temp1, #0x00, loop

```

ret

off:

```

      mov dat,#11111111b
      mov adr,#00000101b
      call writ
;      call Ldelay
      mov dat,#00000000b
      mov adr,#00000110b
      call writ
;      call Ldelay
ret

```

Ldelay:

```

      call delay
      call delay
      call delay
      call delay
      call delay
      call delay
      call delay
      call delay
      call delay
      call delay

```

RET

end1:

END

2. Хід роботи

1. Вивчити принцип динамічного методу відображення інформації на прикладі матриці світлодіодів.
2. Розробити алгоритм індивідуального завдання до початку заняття.
3. Розробити програму і скомпілювати її.
4. Завантажити програму в стенд, переконатись в правильності її роботи, при негативному результаті виявити допущені помилки і виправити їх. Повторити завантаження програми в стенд.

Варіанти індивідуальних завдань

№	Текст індивідуального завдання
1-18	На матрицю світлодіодів по черзі вивести 4 символи тривалістю 1с кожен. 1: Номер варіанту 2: Перша літера прізвища 3: Перша літера імені 4: Перша літера по-батькові Після цього процедуру повторювати спочатку

ЛАБОРАТОРНА РОБОТА №4

ТЕМА: Дослідження методів зчитування інформації в мікроконтролері за допомогою клавіатури.

МЕТА: Вивчити методи взаємодії контролера із засобами вводу даних та виводу даних на засоби відображення інформації.

1. Теоретичні відомості

1.1 Зчитування клавіатури

Зчитування із клавіатури відбувається методом сканування по рядках. Для здійснення зчитування із першого стовпця потрібно записати в порт P2 адресу 0EFh, і зчитати дані із старшого півбайта порту P0. Для цього необхідно задати напрям передачі даних буфера U13 на прийом інформації (встановити вивід DIR в логічний нуль), це значення задається портом мікроконтролера P3.6. Для роботи на прийом від шини до контролера встановити P3.6 в логічний 0. При цьому якщо натиснута кнопка із першого рядка, то значення порту P0.0 буде рівне логічному 0, якщо натиснута кнопка із другого рядка, то P0.1=0 і т.д. Для зчитування із другого стовпця в порт P2 записується адрес 0DFh, третього – 0BFh.

Зчитування із рядків клавіатури слід проводити не частіше ніж через 50мс, через те, що для будь-якого механічного контакту характерне явище вібрації контактної пари, протягом якої контакт декількаразово розмикається-замикається, протягом приблизно 8-12мс.

1.2 Схема для роботи з клавіатурою

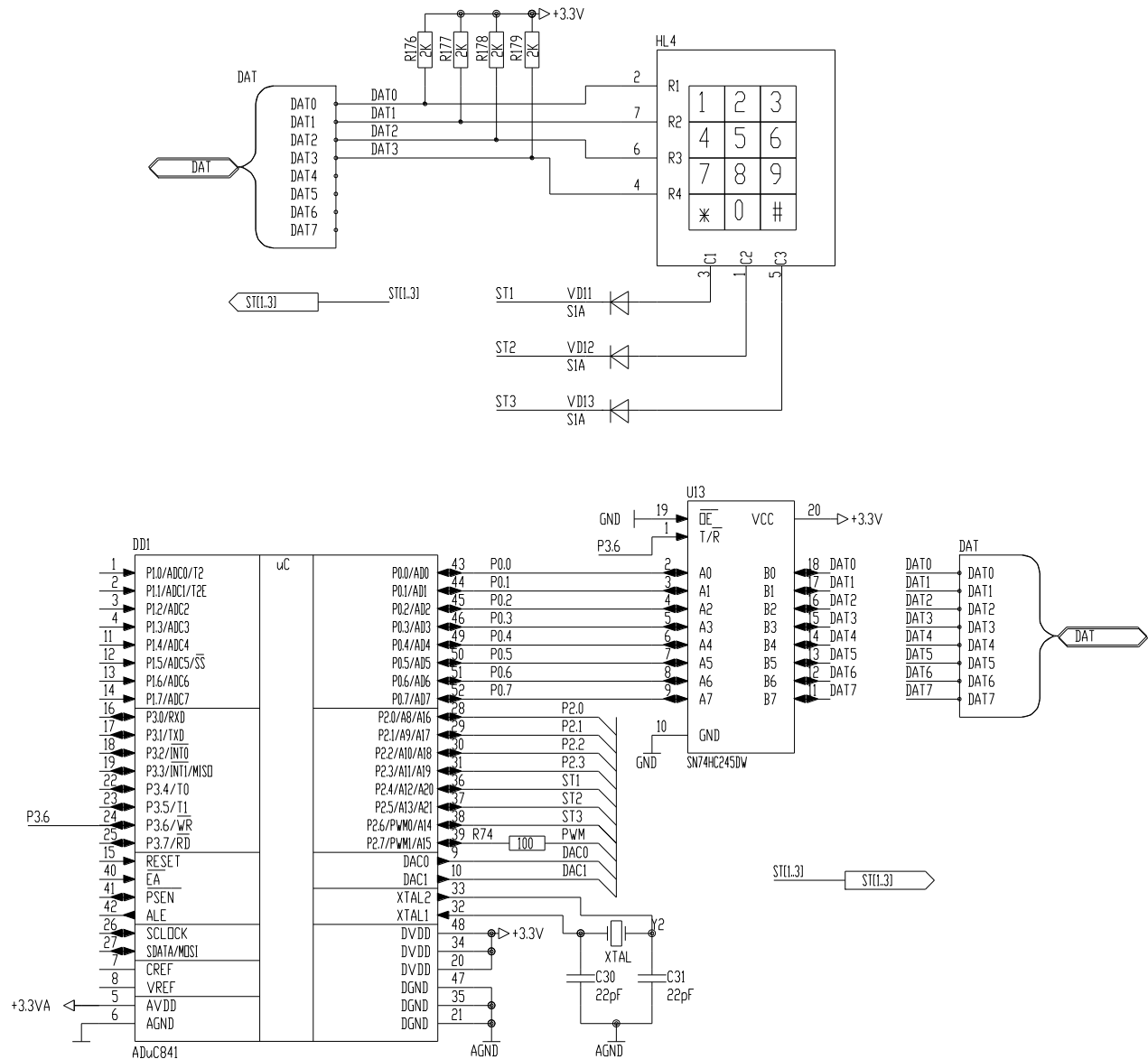


Рисунок 1 – Схема для роботи з клавіатурою

1.3 Приклад програми для роботи з клавіатурою

; Програма зчитує дані з клавіатури і засвічує відповідний світлодіод.

```
dat    Equ R0
adr    Equ R1
Temp1  Equ R2
Temp2  Equ R3
;key1  Equ R4
;key2  Equ R5
stec   Equ 0x02
```

```
Org 0x000
jmp run
```

```
cseg
org 0x033
```

```
run:
```

```
stovb1:
```

```
    mov adr, #01100000b
    mov P2, adr
    clr p3.6
```

```

        mov a, p0
next0:  jb p0.0, next1
        call svit8
        jmp run
next1:  jb p0.1, next2
        call svit5
        jmp run
next2:  jb p0.2, next3
        call svit2
        jmp run
next3:  jb p0.3, stovb2
        call gas1
        jmp run

stovb2: mov adr, #01010000b
        mov P2, adr
        clr p3.6
        mov a, p0
next02: jb p0.0, next12
        call svit7
        jmp run
next12: jb p0.1, next22
        call svit4
        jmp run
next22: jb p0.2, next32
        call svit1
        jmp run
next32: jb p0.3, stovb3
        call svit7
        call svit6
        call svit5
        call svit4
        call svit3
        call svit2
        call svit1
        jmp run

stovb3: mov adr, #00110000b
        mov P2, adr
        clr p3.6
        mov a, p0
next03: jb p0.0, next13
        call svit6
        jmp run
next13: jb p0.1, next23
        call svit3
        jmp run
next23: jb p0.2, next33
        call svit2
        call svit3
        call svit4
        call svit5
        call svit6
        call svit7
        call svit8
        jmp run
next33: jb p0.3, stovb1
        call gas1
        jmp run

svit1:  mov dat, #01111111b
        mov adr, #00000111b
        call writ
        call ldelay

```

```

ret
svit2:
    mov dat,#10111111b
    mov adr,#00000111b
    call writ
    call ldelay
ret
svit3:
    mov dat,#11011111b
    mov adr,#00000111b
    call writ
    call ldelay
ret
svit4:
    mov dat,#11101111b
    mov adr,#00000111b
    call writ
    call ldelay
ret
svit5:
    mov dat,#11110111b
    mov adr,#00000111b
    call writ
    call ldelay
ret
svit6:
    mov dat,#11111011b
    mov adr,#00000111b
    call writ
    call ldelay
ret
svit7:
    mov dat,#11111101b
    mov adr,#00000111b
    call writ
    call ldelay
ret
svit8:
    mov dat,#11111110b
    mov adr,#00000111b
    call writ
    call ldelay
ret
gas1:
    mov dat,#11111111b
    mov adr,#00000111b
    call writ
    call ldelay
ret
Writ:
    setb P3.6
    mov P0, dat
    mov P2, adr
    nop
    mov P2, #0x00
ret
Delay:
    mov Temp1, #0xFF
loop:
    dec Temp1
    mov Temp2, #0xFF
    djnz Temp2, $
    cjne Temp1, #0x00, loop
ret
Ldelay:
    call delay
    call delay
    call delay

```

RET
END

call delay
call delay
call delay
call delay
call delay
call delay
call delay

2. Хід роботи

1. Вивчити алгоритми зчитування даних із клавіатури.
2. Розробити алгоритм індивідуального завдання до початку заняття.
3. Розробити програму і скомпілювати її.
4. Завантажити програму в стенд, переконатись в правильності її роботи, при негативному результаті виявити допущені помилки і виправити їх. Повторити завантаження програми в стенд.

Варіанти індивідуальних завдань

№	Текст індивідуального завдання
1	Початковий стан: всі елементи матриці світлодіодів погашено. При натисненні на матричній клавіатурі кнопки «1» засвічується нижній рядок світлодіодів. При натисненні кнопки «2» засвічуються два нижніх рядки світлодіодів. Аналогічно для кнопок «3»..«7». При натисненні кнопки «*» всі світлодіоди матриці повинні погаситись.
2	Початковий стан: всі світлодіоди на лінійці світлодіодів погашені. При натисканні на матричній клавіатурі кнопки «1» засвітити верхній світлодіод на лінійці світлодіодів; при натисненні кнопки «2» засвітити другий зверху світлодіод Аналогічно для кнопок «3»..«8». При натисненні кнопки «0» – всі світлодіоди погасити.
3	Початковий стан: всі елементи матриці світлодіодів погашено. При натисненні на матричній клавіатурі кнопки «1» засвічується крайній лівий стовбець світлодіодів. При натисненні кнопки «2» засвічуються зліва два перших стовпці світлодіодів. Аналогічно для кнопок «3»..«5». При натисненні кнопки «#» всі світлодіоди матриці повинні погаситись.
4	Початковий стан: всі світлодіоди на лінійці світлодіодів погашені. При натисканні на матричній клавіатурі кнопки «1» засвітити верхній світлодіод на лінійці світлодіодів; при натисненні кнопки «2» засвітити два верхніх світлодіоди. Аналогічно для кнопок «3»..«8». При натисненні кнопки «#» – всі світлодіоди погасити.
5	Початковий стан: засвічено 1 елемент по середині матриці світлодіодів. За допомогою кнопок матричної клавіатури переміщувати засвічену точку по матричному індикаторі. При натисненні кнопки: «4» – переміщувати засвічений світлодіод на один сегмент вліво, «6» – переміщувати на один сегмент вправо, «2» – переміщувати на один сегмент вгору, «8» – переміщувати на один сегмент вниз. При досягненні країв екрану засвічений індикатор за межі не повинен виходити.
6	Початковий стан: всі розряди семисегментного індикатора погашено. При натисканні на матричній клавіатурі відображати на крайньому лівому символі семисегментного індикатора номер натиснутої кнопки. При натисканні «*» загасити індикатор.
7	Початковий стан: всі елементи матриці світлодіодів погашено. При натисненні на матричній клавіатурі кнопки «1» засвічується лівий стовбець світлодіодів. При натисненні кнопки «2» засвічується другий зліва стовбець світлодіодів. Аналогічно для кнопок «3»..«5». При натисненні кнопки «0» всі світлодіоди матриці повинні погаситись.
8	Початковий стан: всі елементи матриці світлодіодів погашено. При натисненні на матричній клавіатурі кнопки відображати відповідну цифру на матриці світлодіодів. При натисканні «*» або «#» загасити всі світлодіоди матриці.
9	Початковий стан: всі елементи матриці світлодіодів погашено. При натисненні на

	матричній клавіатурі кнопки «1» засвічується перший нижній світлодіод першого лівого стовпця. При натисненні кнопки «2» засвічується перший нижній світлодіод першого лівого стовпця і два нижніх другого стовпця. При натисненні кнопки «3» крім попередніх світлодіодів засвічуються три нижніх світлодіоди третього стовпця. При натисненні кнопки «4» крім попередніх світлодіодів засвічуються ще 4 нижніх світлодіодів четвертого стовпця. При натисненні кнопки «5» крім попередніх світлодіодів засвічуються ще 5 нижніх світлодіодів п'ятого стовпця. При натисненні кнопки «0» всі світлодіоди погашені.
10	Початковий стан: всі світлодіоди на лінійці світлодіодів погашені. При натисканні на матричній клавіатурі кнопки «0» на лінійці світлодіодів починає переміщуватися засвічений світлодіод від нижнього до верхнього по колу з інтервалом 0,5 с. При натисненні кнопок від 1 до 9 інтервал переміщення змінюється від 1 с. до 9 с. При натисканні кнопок «*» або «#» всі світлодіоди погасяться.
11	Початковий стан: всі розряди семисегментного індикатора погашено. При натисканні кнопки на матричній клавіатурі від «0» до «4» номер натиснутої кнопки відображати на другому розряді семисегментного індикатора, при натисканні кнопки від «5» до «9» відображати номер натиснутої кнопки на третьому розряді семисегментного індикатора, натискання «*» гасить всі розряди індикатора.
12	Початковий стан: всі розряди семисегментного індикатора погашено. При натисканні кнопки на матричній клавіатурі від «0» до «2» номер натиснутої кнопки відображати на першому розряді семисегментного індикатора. При натисканні кнопки від «3» до «4» відображати номер натиснутої кнопки на другому розряді семисегментного індикатора. При натисканні кнопки від «5» до «7» відображати номер натиснутої кнопки на третьому розряді семисегментного індикатора. При натисканні кнопки від «8» до «9» відображати номер натиснутої кнопки на четвертому розряді семисегментного індикатора. При натисканні кнопки «*» або «#» розряди індикатора повинні погаситись.
13	Початковий стан: всі розряди семисегментного індикатора та всі елементи матриці світлодіодів погашені. При натисканні кнопки на матричній клавіатурі від «0» до «4» номер натиснутої кнопки відображати на останньому розряді семисегментного індикатора, при натисканні кнопки від «5» до «9» відображати номер натиснутої кнопки на матриці світлодіодів, натискання кнопки «#» гасить всі розряди індикатора.
14	Початковий стан: всі елементи матриці світлодіодів погашено. При натисненні на матричній клавіатурі кнопки «1» засвічується нижній ряд світлодіодів. При натисненні кнопки «2» засвічується другий знизу ряд світлодіодів. Аналогічно для кнопок «3»..«7». При натисненні кнопки «0» всі світлодіоди матриці повинні погаситись.
15	Початковий стан: всі елементи матриці світлодіодів погашено. При натисненні на матричній клавіатурі кнопки «1» засвічується верхній рядок світлодіодів. При натисненні кнопки «2» засвічується другий зверху рядок світлодіодів. Аналогічно для кнопок «3»..«7». При натисненні кнопки «0» всі світлодіоди матриці повинні погаситись.
16	Початковий стан: всі розряди семисегментного індикатора погашено. При натисканні на матричній клавіатурі відображати на крайньому правому символі семисегментного індикатора номер натиснутої кнопки. При натисканні «#» загасити індикатор.
17	Початковий стан: всі світлодіоди на лінійці світлодіодів погашені. При натисканні на матричній клавіатурі кнопки «1» засвітити нижній світлодіод на лінійці світлодіодів; при натисненні кнопки «2» засвітити два нижніх світлодіоди. Аналогічно для кнопок «3»..«8». При натисненні кнопки «*» – всі світлодіоди погасити.
18	Початковий стан: всі світлодіоди на лінійці світлодіодів погашені. При натисканні на матричній клавіатурі кнопки «1» засвітити нижній світлодіод на лінійці світлодіодів; при натисненні кнопки «2» засвітити другий знизу світлодіод. Аналогічно для кнопок «3»..«8». При натисненні одночасно кнопок «#» і «*» – всі світлодіоди погасити.

ЛАБОРАТОРНА РОБОТА №5

ТЕМА: Робота з рідкокристалічним дисплеєм.

МЕТА: Вивчити команди PKI, вивід інформації на дисплей, взаємодія дисплея і контролера.

1. Теоретичні відомості

1.1 Схема для роботи з рідкокристалічним дисплеєм

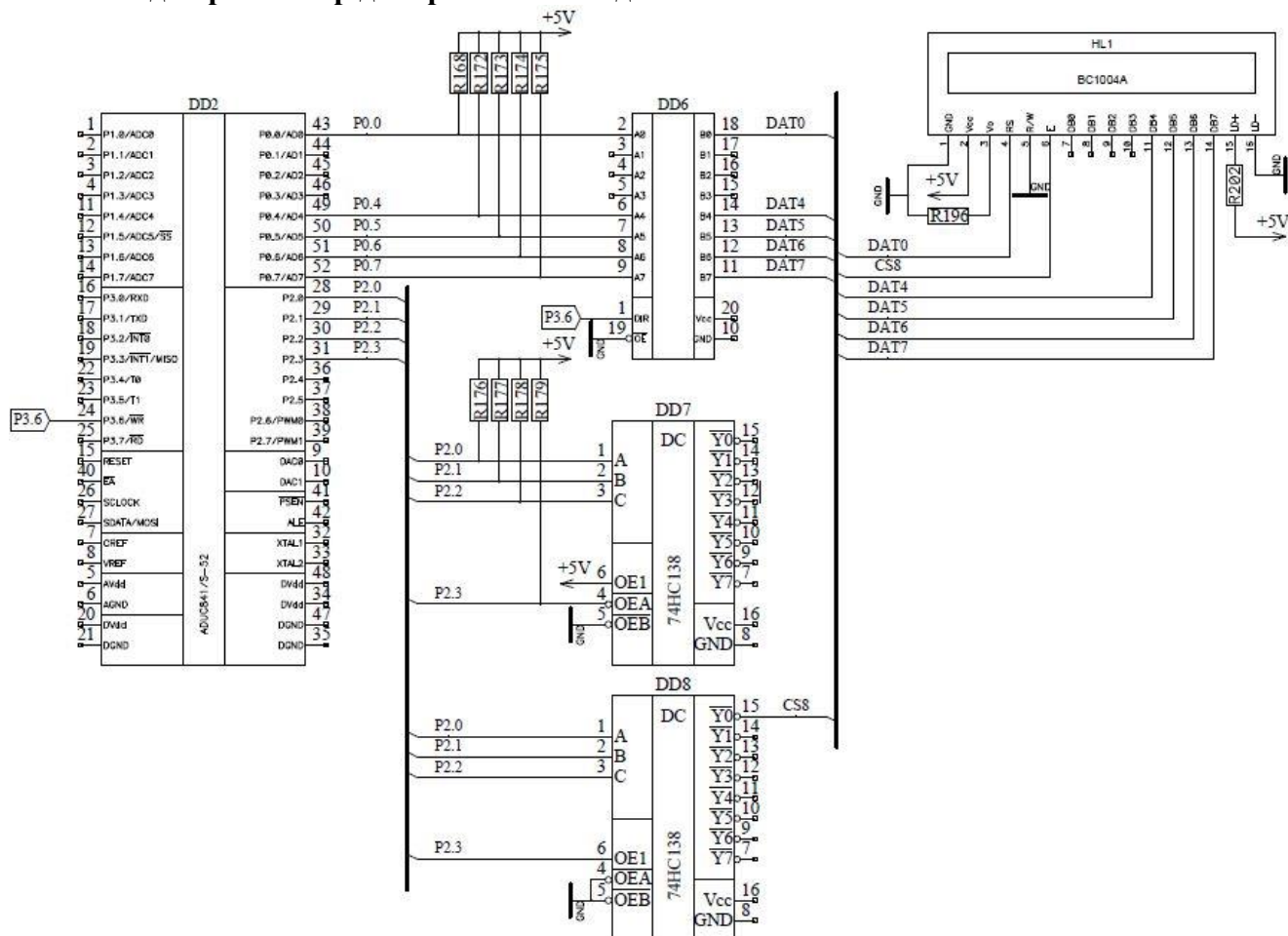


Рисунок 1 – Схема для роботи з рідкокристалічним дисплеєм

1.2 Аналого-цифровий перетворювач

Стенд оснащений символьним рідкокристалічним індикатором (PKI) типу BC1004 побудованим на контролері HD44780 стандарту. Індикатор підключений по схемі із 4-бітною шиною даних. Адрес індикатора 08h.

Індикатор обмінюється інформацією з керуючим мікроконтролером (в даному випадку ADuC841). Контролер посилає в PKI команди (таблиця 2), які керують режимами його роботи і ASCII-коди символів, що виводяться на індикатор. В свою чергу, PKI може посылати контролеру по його запиту інформацію про свій внутрішній стан і дані із своїх внутрішніх блоків пам'яті, але в даній схемі включення така можливість не передбачається (вивід R/W замкнутий на землю).

Три виводи роз'єму призначені для подачі живлення і напруги зміщення, що керує контрастом дисплею. 8 виводів (DB0-DB7) призначені для організації мультимплексованої шини команди/дані. На 3 виводи (RS, R/W, E) контролер подає керуючі сигнали.

Таблиця 1 – Опис виводів стандартного роз'єму PKI на базі HD44780

№	Назва виводу	Опис
1	VSS	(-) Живлення. 0 V.
2	VDD	(+) Живлення.+5V.
3	V0	Напруга зміщення, керуюча контрастом
4	RS	Вхід. Високий рівень – дані; Низький - команди
5	R/W	Вхід. Високий - Читання, Низький – Запис
6	E	Вхід. Строб, супроводжуючий сигнали на шині "команди/дані"
7	DB0	Шина "команди/дані"
8	DB1	Те ж
9	DB2	-//-
10	DB3	-//-
11	DB4	-//-
12	DB5	-//-
13	DB6	-//-
14	DB7	-//-

З допомогою сигналу RS – register select, контролер вказує що передається по шині: команда чи дані. Сигнал на лінії E (enable) є стробом, що супроводжує сигнали на шині команд/даних. Запис в PKI проходить по спаду цього сигналу. Потенціал на керуючій виводі R/W задає напрям передачі даних: запис в RAM індикатора (R/W=0) або читання звідти (R/W=1).

Для випадку, коли контролер має обмежену кількість ліній вводу / виводу, передбачено другий варіант підключення PKI з використанням 4-розрядної шини команди / дані. При цьому кожний байт даних передається по лініях DB4 - DB7 послідовно двома тетрадами, починаючи із старшої.

Контролер PKI після прийому байту команди або байту даних потребує деякого часу (таблиця 2) для обробки отриманої інформації, протягом якого мікроконтролер ADuC841 не повинен виконувати нових передач.

Якщо під час циклу запису мікроконтролер передає в контролер індикатора код команди, то цей код записується в регістр команд контролера PKI, і команда відразу ж починає виконуватися. Якщо мікроконтролер передає в контролер PKI дані, які є ASCII-кодом символів, що відображаються, то вони записуються в буфер даних (DDRAM), який звичайно містить 80 комірок. Під час запису або зчитуванні буферу даних звернення здійснюється до комірки, на яку в даний момент вказує курсор.

Для роботи з індикатором спочатку потрібно провести його ініціалізацію – вказати розрядність шини даних/команд, кількість рядків, наявність/відсутність курсору і т. ін.

Оскільки PKI підключений по схемі із 4-розрядною шиною даних/команд, слід пам'ятати, що кожна команда або набір даних передаються по пів байта, а в таблиці 3 вказані команди для 8-розрядного підключення.

Приведемо часові діаграми під час передачі контролером команди для PKI, а також під час передачі даних.

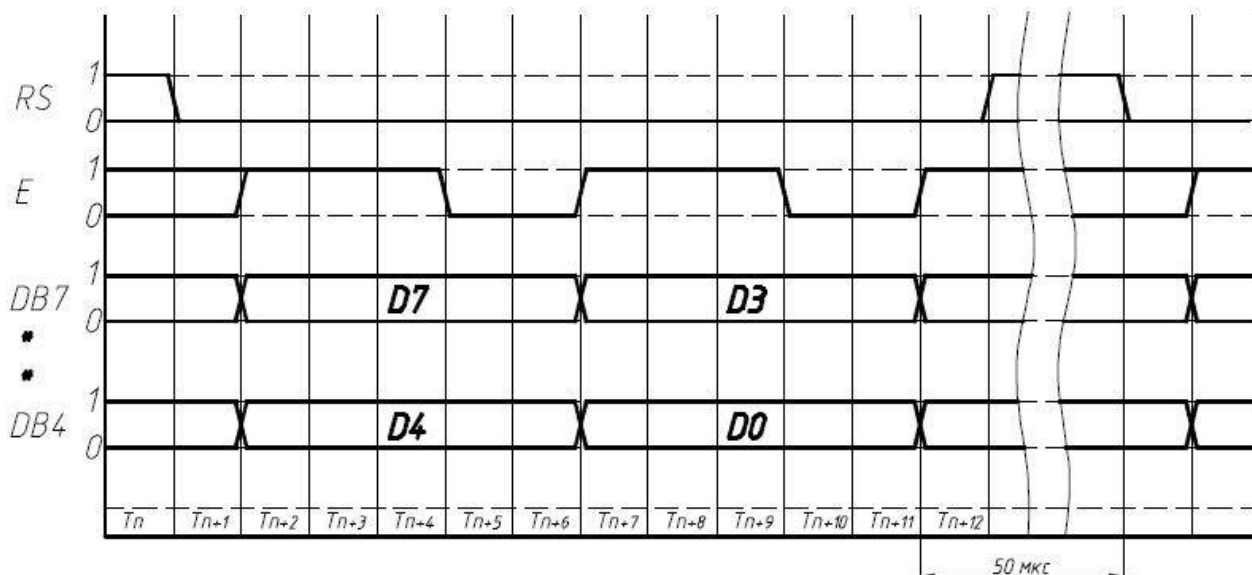


Рисунок 2 – Часова діаграма запису команди в РКІ

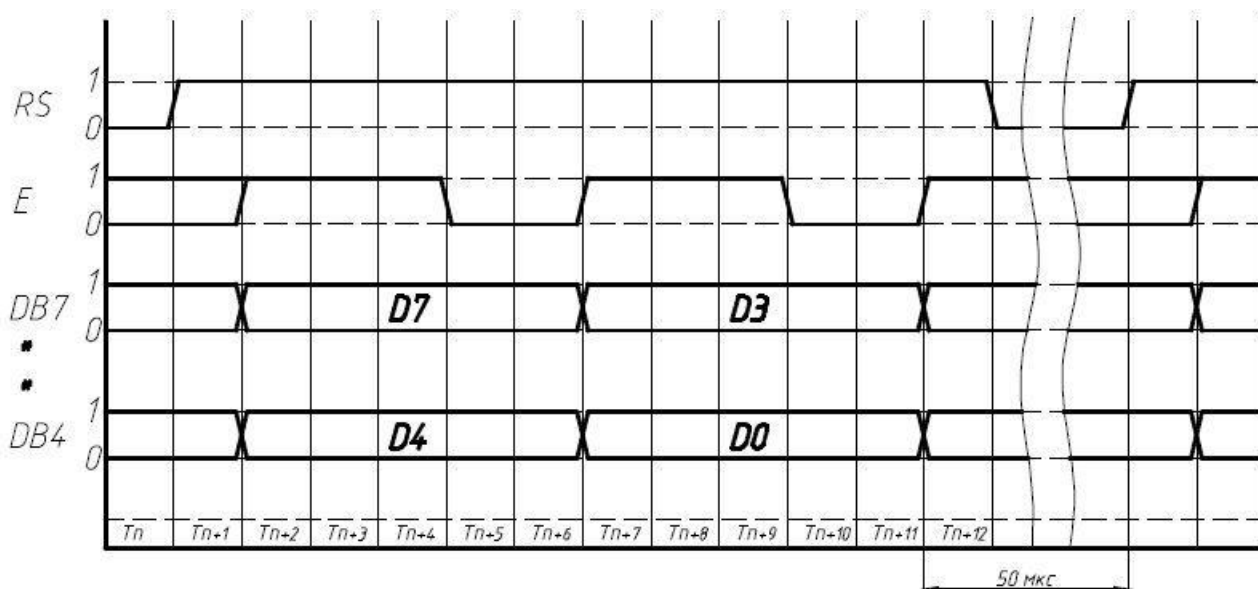


Рисунок 3 – Часова діаграма запису даних в РКІ

Як видно із діаграм, між посилками байтів команд/даних потрібно забезпечити паузу не менше 50мкс, оскільки цей час необхідний для опрацювання РКІ переданих даних.

Примітка:

DDRAM- Display data RAM - ОЗП ASCII-кодів, відображуваних символів

CGRAM - Character generator RAM - ОЗП знакогенератора

Адресація знакомісць РКІ проводиться таким чином (адреси в 16ковім вигляді):

0x00, 0x01, 0x02, 0x03, 0x04, 0x05, 0x06, 0x07, 0x08, 0x09,

0x40, 0x41, 0x42, 0x43, 0x44, 0x45, 0x46, 0x47, 0x48, 0x49,

0x0A, 0x0B, 0x0C, 0x0D, 0x0E, 0x0F, 0x10, 0x011, 0x12, 0x13,

0x4A, 0x4B, 0x4C, 0x4D, 0x4E, 0x4F, 0x50, 0x51, 0x52, 0x53.

Таблиця 2 – Система команд контролера HD4478

Код										Опис команди	Час виконання команди
R S	R / W	DB7	DB6	DB5	DB4	DB3	DB2	DB1	DB0		
0	0	0	0	0	0	0	0	0	1	Очистити дисплей і встановити курсор в нульову позицію (адрес 0)	82мкс до 1.64мс
0	0	0	0	0	0	0	0	1	*	Встановити курсор в нульову позицію (адрес 0). Встановити дисплей відносно буфера DDRAM в початкову позицію. Вміст DDRAM при цьому не міняється.	40мкс до 1.6мс
0	0	0	0	0	0	0	1	I/D	S	Встановити напрям зсуву курсору вправо (I/D=1) або вліво (I/D=0) при записі/читанні чергового коду в DDRAM. Дозволити (S=1) зсув дисплея разом із зсувом курсору.	40мкс
0	0	0	0	0	0	1	D	C	B	Включити(D=1)/виключити(D=0) дисплей. Запалити(C=1)/погасити(C=0) курсор. Зображення курсору зробити мигаючим (B=1).	40мкс
0	0	0	0	0	1	S/C	R/L	*	*	Перемістити курсор (S/C=0) або зсунути дисплей (S/C=1) вправо (R/L=1) або вліво(R/L=0).	40мкс
0	0	0	0	1	DL	N	F	*	*	Встановити розрядність шини даних 4 біти (DL=0) або 8 біт (DL=1), кількість рядків дисплею – один (N=0)або два (N=1), шрифт - 5x7 точок (F=0) або 5x10 точок (F=1).	40мкс
0	0	0	1	ACG						Установка адреси CGRAM. Після цієї команди дані будуть записуватися/зчитуватися в/із CGRAM.	40мкс
0	0	1	ADD							Установка адреси DDRAM. Після цієї команди дані будуть записуватися/зчитуватися в/із DDRAM.	40мкс
0	0	BF	AC							Читання стану busy-флага (BF) і лічильника адреса.	1мкс
1	0	Дані								Запис даних в DDRAM або CGRAM.	40мкс
1	1	Дані								Читання даних із DDRAM або CGRAM.	40мкс

	0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F
0				0	@	P	`	F			Б	Ю	Ч	.	Д	¼
1			!	1	Q	O	a	q			Г	Я	Ш	,	Ц	½
2			"	2	B	R	b	r			Ё	Б	Ъ	"	Щ	⅔
3			#	3	C	S	c	s			Ж	В	Ы	!	З	¼
4			\$	4	D	T	d	t			Э	Г	Ь	÷	Ф	⅓
5			%	5	E	U	e	u			И	Ё	О	Х	Ц	⅔
6			&	6	F	V	f	v			Й	Ж	Ю	÷	Щ	¼
7			'	7	G	W	g	w			Л	Э	Я	І	'	⅓
8			(8	H	X	h	x			П	И	О	И	"	¼
9)	9	I	Y	i	y			У	Й	О	†	~	⅓
A			*	:	J	Z	j	z			Ф	К	α	↓	é	¼
B			+	;	K	[k]			Ч	Л	"	Н	♀	¼
C			,	<	L	φ	l	ε			Ш	М	№	Н	ü	¼
D			-	=	M]	m	ε			Ь	Н	¿	Н	♣	¼
E			.	>	N	^	n	ε			Ы	П	f	÷	○	¼
F			/	?	O	_	o	ε			Э	Т	£	°	○	■

Рисунок 3 – Таблица кодов символов контролера рідкокристалічного індикатора

1.3 Приклад програми для роботи з рідкокристалічним дисплеєм

;Вивести на РК індикатор слово
"Привіт".

```
DAT EQU R0
adr EQU R1
Temp0 EQU R2
Temp0 EQU 0x02
```

```
CSEG
ORG 0x0000
JMP run
```

```
ORG 0x0033
run:
```

```
begin: CALL LCD_INIT
```

```
call str1
call word1
call str2
call word1
call str3
call word1
call str4
call word1
```

```
jmp begin
```

```
LCD_INIT:
```

```
mov dat, #0x00
mov adr, #0x08
call writ
call ldelay
```

```
mov dat, #00010000b
mov adr, #0x08
call writ
call ldelay
```

```
mov dat, #0x00
mov adr, #0x08
call writ
call ldelay
```

```
mov dat, #00100000b
mov adr, #0x08
call writ
call ldelay
```

```
mov dat, #0x00
mov adr, #0x08
call writ
call ldelay
```

```
mov dat, #01100000b
mov adr, #0x08
call writ
call ldelay
```

```
mov dat, #0x00
mov adr, #0x08
call writ
call ldelay
```

```
mov dat, #11100000b
mov adr, #0x08
call writ
call wldelay
call ldelay
```

```
mov dat, #00100000b
mov adr, #0x08
call writ
call ldelay
```

```
mov dat, #10000000b
mov adr, #0x08
call writ
```

```
RET call ldelay
```

```
word1:
// Виведення символу «П»
mov dat, #0xA1
mov adr, #0x08
call writ
call ldelay
```

```
mov dat, #0x00381
mov adr, #0x08
call writ
call ldelay
```

```
// Виведення символу «р»
mov dat, #0x71
mov adr, #0x08
call writ
call ldelay
```

```
mov dat, #0x01
mov adr, #0x08
call writ
call ldelay
```

```
// Виведення символу «и»
mov dat, #0xB1
mov adr, #0x08
call writ
call ldelay
```

```
mov dat, #0x81
mov adr, #0x08
call writ
call ldelay
```

```
// Виведення символу «в»
mov dat, #0xB1
mov adr, #0x08
call writ
call ldelay
```

```
mov dat, #0x31
mov adr, #0x08
call writ
call ldelay
```

```
// Виведення символу «і»
mov dat, #0x61
mov adr, #0x08
call writ
call ldelay
```

```
mov dat, #0x91
mov adr, #0x08
call writ
call ldelay
```

```
// Виведення символу «т»
mov dat, #0xB1
mov adr, #0x08
call writ
call ldelay
```

```
mov dat, #0xF1
mov adr, #0x08
call writ
call ldelay
```

```
ret
// Команда виводу інформації на
1 стрічку
str1:
```

```
mov dat, #10000000b
mov adr, #0x08
call writ
call ldelay
```

```
mov dat, #00000000b
mov adr, #0x08
call writ
```

```
call ldelay
```

```
ret
// Команда виводу інформації на
2 стрічку
str2:
```

```
mov dat, #11000000b
mov adr, #0x08
call writ
call ldelay
```

```
mov dat, #00000000b
mov adr, #0x08
call writ
call ldelay
```

```
ret
// Команда виводу інформації на
3 стрічку
str3:
```

```
mov dat, #10000000b
mov adr, #0x08
call writ
call ldelay
```

```
mov dat, #10100000b
mov adr, #0x08
call writ
call ldelay
```

```
ret
// Команда виводу інформації на
4 стрічку
str4:
```

```
mov dat, #11000000b
mov adr, #0x08
call writ
call ldelay
```

```
mov dat, #10100000b
mov adr, #0x08
call writ
call ldelay
```

```
ret
```

```
writ: SETB P3.6
mov P0, dat
mov P2, adr
MOV P2, #00h
```

```
ret
// Delay 46 us
```

```
Delay: MOV Temp0,
#0xFF DJNZ Temp0, $
```

```
RET
// Delay 17.9 ms
```

```
LDelay: MOV Temp0, #0xFF
```

```
LD1: DEC Temp0
PUSH _Temp0
MOV Temp0,
```

```
#0xFF DJNZ Temp0, $
POP _Temp0
CJNE Temp0, #0x00,
```

```
LD1
RET
// .50s
```

```
WLDelay: MOV Temp0, #0x19
```

```
WL2: DEC Temp0
PUSH _Temp0
CALL LDelay
POP _Temp0
```

```
WL2    CJNE    Temp0,    #0x00,  
RET  
END
```

2. Хід роботи

1. Вивчити команди ініціалізації та керування РКІ.
2. Розробити алгоритм індивідуального завдання до початку заняття.
3. Розробити програму і скопіювати її.
4. З допомогою програмних відладочних засобів Keil проаналізувати часові діаграми передачі команд та даних від контролера до РКІ.
5. Завантажити програму в стенд, переконатись в правильності її роботи, при негативному результаті виявити допущені помилки і виправити їх. Повторити завантаження програми в стенд.

Варіанти індивідуальних завдань

№	Текст індивідуального завдання
1	При нажатті кнопки «1» на першому рядку рідкокристалічного дисплея вивести номер варіанту, при нажатті кнопки «2» на другому рядку дисплея вивести ім'я, при нажатті кнопки «3» на третьому рядку дисплея вивести прізвище, при нажатті кнопки «4» на четвертому рядку дисплея вивести по-батькові. Всі символи виводити українськими літерами. При натисканні кнопки «0» дисплей очищується від усіх написів.
2	При нажатті кнопки «2» на першому рядку рідкокристалічного дисплея вивести номер варіанту, при нажатті кнопки «3» на другому рядку дисплея вивести ім'я, при нажатті кнопки «4» на третьому рядку дисплея вивести прізвище, при нажатті кнопки «5» на четвертому рядку дисплея вивести по-батькові. Всі символи виводити українськими літерами. При натисканні кнопки «1» дисплей очищується від усіх написів.
3	При нажатті кнопки «3» на першому рядку рідкокристалічного дисплея вивести номер варіанту, при нажатті кнопки «4» на другому рядку дисплея вивести ім'я, при нажатті кнопки «5» на третьому рядку дисплея вивести прізвище, при нажатті кнопки «6» на четвертому рядку дисплея вивести по-батькові. Всі символи виводити українськими літерами. При натисканні кнопки «2» дисплей очищується від усіх написів.
4	При нажатті кнопки «4» на першому рядку рідкокристалічного дисплея вивести номер варіанту, при нажатті кнопки «5» на другому рядку дисплея вивести ім'я, при нажатті кнопки «6» на третьому рядку дисплея вивести прізвище, при нажатті кнопки «7» на четвертому рядку дисплея вивести по-батькові. Всі символи виводити українськими літерами. При натисканні кнопки «3» дисплей очищується від усіх написів.
5	При нажатті кнопки «5» на першому рядку рідкокристалічного дисплея вивести номер варіанту, при нажатті кнопки «6» на другому рядку дисплея вивести ім'я, при нажатті кнопки «7» на третьому рядку дисплея вивести прізвище, при нажатті кнопки «8» на четвертому рядку дисплея вивести по-батькові. Всі символи виводити українськими літерами. При натисканні кнопки «4» дисплей очищується від усіх написів.
6	При нажатті кнопки «6» на першому рядку рідкокристалічного дисплея вивести номер варіанту, при нажатті кнопки «7» на другому рядку дисплея вивести ім'я, при нажатті кнопки «8» на третьому рядку дисплея вивести прізвище, при нажатті кнопки «9» на четвертому рядку дисплея вивести по-батькові. Всі символи виводити українськими літерами. При натисканні кнопки «5» дисплей очищується від усіх написів.
7	При нажатті кнопки «0» на першому рядку рідкокристалічного дисплея вивести номер варіанту, при нажатті кнопки «1» на другому рядку дисплея вивести ім'я, при нажатті кнопки «2» на третьому рядку дисплея вивести прізвище, при нажатті кнопки «3» на четвертому рядку дисплея вивести по-батькові. Всі символи виводити українськими літерами. При натисканні кнопки «5» дисплей очищується від усіх написів.

ЛАБОРАТОРНА РОБОТА №6

ТЕМА: Дослідження процесу аналого-цифрового перетворення в мікроконтролері на прикладі джойстика на змінних резисторах.

МЕТА: Вивчити методи оцифровування аналогових сигналів.

1. Теоретичні відомості

1.1 Схема для роботи з аналогово-цифровим перетворювачем

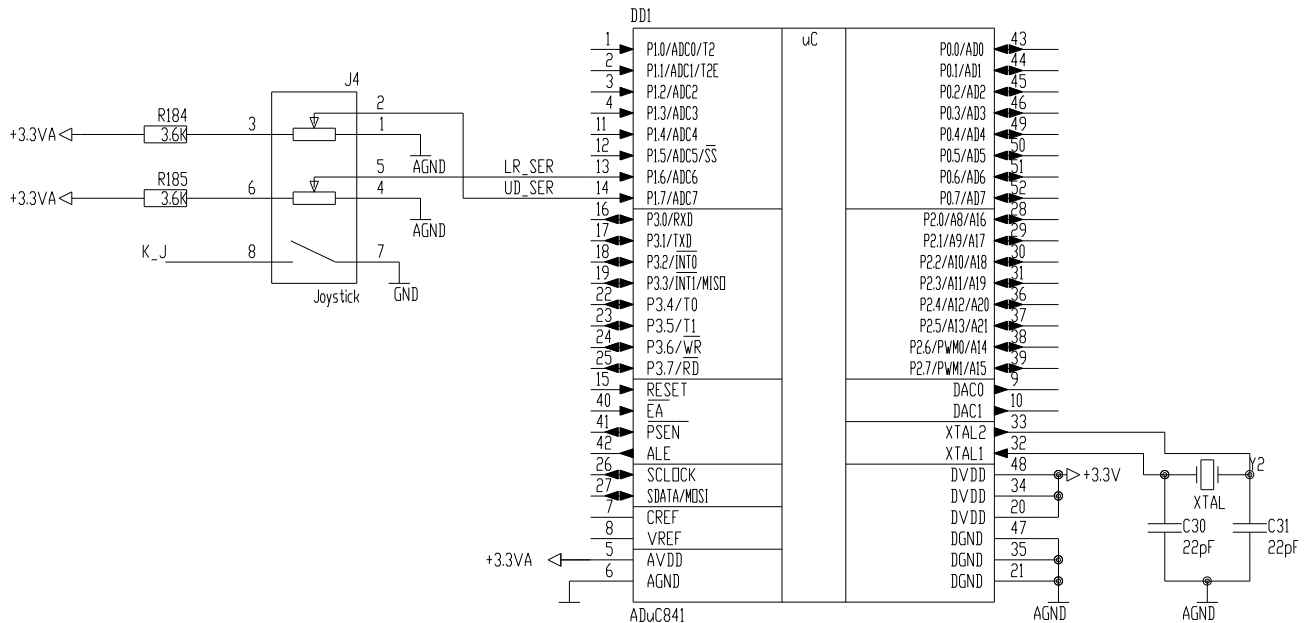


Рисунок 1 – Схема для роботи з аналогово-цифровим перетворювачем

1.2 Аналого-цифровий перетворювач

АЦП, аналого-цифровий перетворювач – пристрій, що перетворює вхідний сигнал в дискретний код (цифровий сигнал). Як правило, АЦП – електронний пристрій, що перетворює напругу в двійковий код.

Розрядність АЦП характеризується кількістю дискретних значень, які перетворювач може видати на виході, вимірюється в бітах. Наприклад, АЦП, здатний видавати 256 дискретних значень (0...255), має розрядність 8 біт, оскільки: $2^8=256$.

Розрядність може бути також визначена в величинах вхідного сигналу і виражена, наприклад, в вольтах. Розрядність за напругою дорівнює напрузі, що відповідає максимальному вихідному коду, який ділиться на кількість вихідних значень.

Мікроконтролер ADuC841 як і інші пристрої компанії ADI має високоточний АЦП і програмований ЦАП.

АЦП керується за допомогою регістрів спеціальних функцій ADCCON1..ADCCON3. Молодших 8 біт результату перетворення зберігаються в регістрі ADCDATA1, старших 4 біти – в молодшій тетраді регістру ADCDATA1. В старшу тетраду контролер записує номер поточного каналу.

Значення регістру ADCCON1 керує режимом запуску та роботи АЦП. Його адреса – 0xEF, початкове значення – 0x40, побітова адресація не підтримується.

Таблиця 1 – Біти регістру ADCCON1

Біт	Назва	Призначення
7	MD1	Ввімкнення АЦП. Якщо біт встановлений – АЦП ввімкнено
6	EXT_REF	Біт встановлений – АЦП працює від зовнішнього джерела опорної напруги, Біт скинений – АЦП працює від внутрішнього джерела опорної напруги.
5	CK1	Визначають подільник тактової частоти в частоту АЦП. Частота АЦП не може перевищувати 8.38MHz <div> <div>CK1</div> <div>CK0</div> <div>0</div> <div>0</div> <div>1</div> <div>1</div> <div>0</div> <div>1</div> <div>0</div> <div>1</div> </div>
4	CK0	
3	AQ1	
2	AQ0	
1	AQ0	
3	AQ1	Визначають кількість періодів АЦП, що витрачаються на захоплення сигналу пристроєм вибірки та зберігання (апертурний час). Для високоомних джерел сигналу значення мають збільшуватися. <div> <div>AQ1</div> <div>AQ0</div> <div>0</div> <div>0</div> <div>0</div> <div>1</div> <div>1</div> <div>0</div> <div>1</div> <div>1</div> </div>
2	AQ0	
1	AQ1	
0	AQ0	
0	AQ1	
1	T2C	Встановлений біт дозволяє запуск АЦП при переповненні таймера 2
0	EXC	Встановлений біт дозволяє запуск АЦП при низькому рівні на лінії CONVST

Молодша тетрада регістру ADCCON2 визначає обраний канал перетворення, старша керує запуском АЦП. Адреса регістру – 0xD8, початкове значення – 0x00, побітова адресація підтримується.

Таблиця 2 – Біти регістру ADCCON2

Біт	Назва	Призначення
7	ADCI	Переривання АЦП. Встановлюється контролером в кінці єдиного АЦП перетворення або в кінці блоку перетворень в режимі DMA
6	DMA	Вмикає режим DMA
5	CCONV	Запускає режим послідовних перетворень. Після закінчення кожного перетворення розпочинається нове, доти, доки не користувач не скине біт DMA.
4	SCONV	Запускає одне перетворення. Після його завершення біт автоматично скидається
3	CS3	Визначає активний канал АЦП <div> <div>CS3</div> <div>CS2</div> <div>0</div> <div>0</div> <div>0</div> <div>1</div> <div>0</div> <div>0</div> <div>0</div> <div>0</div> <div>0</div> <div>1</div> </div>
2	CS2	
1	CS1	
0	CS0	
0	CS0	

Регістр ADCCON3 керує процедурою калібрування і не розглядається в даній лабораторній роботі.

Перед тим, як запускати аналогово-цифрове перетворення слід ввімкнути АЦП – записати в регістр ADCCON1 байт ініціалізації. Наприклад ввімкнемо АЦП з такими параметрами:

- Опора – внутрішня (EXT_REF=0)
- Подільник – 2 (CK1=1 CK0=1) ($f_{ADC}=f_{CLK}/2=5.5299\text{МГц}$)
- Апертура – 4 періоди (AQ1=1 AQ0=1)

Запуск від спрацювання таймера 2 та зовнішнього сигналу заборонені (T2C=0 EXC=0). Таким чином байт ініціалізації рівний $10111100_2 = 188_{10} = \text{BC}_{16}$. Записуємо його в регістр ADCCON1.

Далі, щоб провести одиничне вимірювання, потрібно записати в ADCCON2 номер каналу і біти запуску. Так для запуску одиничного перетворення з 7 каналу в ADCCON2 слід записати байт 0x17.

Після запуску дані будуть готові через 17-20 тактів АЦП (в залежності від апертурного часу, встановленого бітами AQ1-AQ0). Якщо подільник частоти АЦП рівний двом (біти СК1=1 СК0=1), то це складе 34-40 тактів мікроконтролера. Взяти точний час завершення аналогово-цифрового перетворення можна опитуючи біт ADCI – в кінці циклу перетворення контролер встановить його.

Результат перетворення знаходитиметься в регістрах ADCDATAH:ADCDATAL.

1.3 Приклад програми для роботи з аналогово-цифровим перетворювачем

; В залежності від положення джойстика засвічується певна кількість світлодіодів.

```
ADCCON1    DATA    0EFH ;ADC CONTROL
ADCCON2    DATA    0D8H ;ADC CONTROL
ADCDATAL   DATA    0D9H ;ADC DATA LOW BYTE
ADCDATAH   DATA    0DAH ;ADC DATA HIGH BYTE
ADCI       BIT      0DFH ;ADCCON2.7 - ADC INTURRUPT FLAG
SCONV      BIT      0DCH ;ADCCON2.4 - SINGLE CONVERSION ENABLE
;DACCON    DATA    0FDH ;DAC CONTROL REGISTER
;CFG841    DATA    0AFH ;GENERAL FLASH/PWM CONTROL REGISTER
```

```
dat        EQU      R0
adr        EQU      R1
Temp1      EQU      R2
Temp2      EQU      R3
dat1       EQU      R4
Segm       EQU      0x02
```

```
CSEG
ORG 0x0000
jmp begin
ORG 0x0033
begin:
    mov     p1, 0xFF
    mov     ADCCON1, #10101100b
    call    off
    mov     TCON, #11111111b
    mov     TMOD, #11001100b

start:
    call    ACP1

    mov     a, th1
    cjne    a, #0, b10
    mov     a, tl1
    cjne    a, #10, analiz_10
    analiz_10:
        jc     b10
        call   sv0
        jmp    start
    b10:
        mov     a, th1
        cjne    a, #000000001b, analiz_475
    analiz_475:
        jc     b475
        call   sv1
    b475:
        mov     a, th1
        cjne    a, #000000011b, analiz_950
    analiz_950:
        jc     b950
        call   sv2
```

```

b950:
    mov a,th1
    cjne a,#00000101b,analiz_1425
analiz_1425:
    jc b1425
    call sv3

b1425:
    mov a,th1
    cjne a,#00000111b,analiz_1900
analiz_1900:
    jc b1900
    call sv4

b1900:
    mov a,th1
    cjne a,#00001001b,analiz_2375
analiz_2375:
    jc b2375
    call sv5

b2375:
    mov a,th1
    cjne a,#00001011b,analiz_2850
analiz_2850:
    jc b2850
    call sv6

b2850:
    mov a,th1
    cjne a,#00001100b,analiz_3325
analiz_3325:
    jc b3325
    call sv7

b3325:
    mov a,th1
    cjne a,#00001110b,analiz_3800
analiz_3800:
    jc b3800
    call sv8

b3800:
    mov a,th1
    cjne a,#00001111b,analiz_max
analiz_max:
    jc      notmax
    //      mov a,t11
    //      cjne a,#11111111b,notmax
    call sv8
    jmp start
notmax:
    jmp start
    //      mov dat,#11100111b
    //      mov adr,#0x7
    //      call strob
    //      jmp start

ACP0:
    mov     ADCCON2, #6h
    clr     ADCI
    setb    SCONV
    jnb     ADCI,$
    mov     b, adcdatal
    mov     a, adcdatah
    clr     acc.7
    clr     acc.6
    clr     acc.5
    clr     acc.4
    mov     Tl0, b
    mov     TH0, a
    mov     R5, TL0
    mov     R7, Th0
ret

ACP1:
    mov     ADCCON2, #7h
    clr     ADCI
    setb    SCONV

```

```

        jnb     ADCI,$
        mov     b, adcdatal
        mov     a, adcdatalh
        clr     acc.7
        clr     acc.6
        clr     acc.5
        clr     acc.4
        mov     TH1, b
        mov     TH1, a
        mov     R5, TL1
        mov     R6, Th1
ret
off:
        mov     dat,#11111111b
        mov     adr,#00000101b
        call    strob
;
        mov     dat,#00000000b
        mov     adr,#00000110b
        call    strob1
ret
strob:
        setb    P3.6
        mov     P0,datal
        mov     P2,adr
        mov     P2,#0x00
ret
strob1:
        setb    P3.6
        mov     P0,datal
        mov     P2,adr
        mov     P2,#0x00
ret
Delay:
        mov     Temp1, #0xFF
loop:
        dec     Temp1
        mov     Temp2, #0xFF
        djnz    Temp2, $
        cjne    Temp1, #0x00, loop
ret
sv0:
        mov     dat,#0xff
        mov     adr,#0x7
        call    strob
        ret
sv1:
        mov     dat,#11111110b
        mov     adr,#0x7
        call    strob
        ret
sv2:
        mov     dat,#11111100b
        mov     adr,#0x7
        call    strob
        ret
sv3:
        mov     dat,#11111000b
        mov     adr,#0x7
        call    strob
        ret
sv4:
        mov     dat,#11110000b
        mov     adr,#0x7
        call    strob
        ret
sv5:
        mov     dat,#11100000b
        mov     adr,#0x7
        call    strob
        ret
sv6:

```

```

mov dat,#11000000b
mov adr,#0x7
call strob
ret
sv7:
mov dat,#10000000b
mov adr,#0x7
call strob
ret
sv8:
mov dat,#00000000b
mov adr,#0x7
call strob
ret

Ldelay:
call delay
call delay
call delay
call delay
call delay
call delay
call delay
call delay
call delay
RET
END

```

2. Хід роботи

6. Вивчити алгоритм обробки даних аналого-цифровим перетворювачем.
7. Розробити алгоритм індивідуального завдання до початку заняття.
8. Розробити програму і скомпілювати її.
9. Завантажити програму в стенд, переконавшись в правильності її роботи, при негативному результаті виявити допущені помилки і виправити їх. Повторити завантаження програми в стенд.

Варіанти індивідуальних завдань

№	Текст індивідуального завдання
1	За допомогою джойстика керувати швидкістю переміщення точки по світлодіодній лінійці. Швидкість переміщення визначається положенням ручки джойстика відносно вертикальної осі, нижнє положення відповідає мінімальній швидкості (1 сегмент за 4 секунди), верхнє положення відповідає максимальній швидкості переміщення засвіченого світлодіода (1 сегмент за 0,5 секунди). Початковий стан: засвічений світлодіод переміщається по лінійці в напрямку знизу вверх з швидкістю відповідно до положення ручки джойстика.
2	За допомогою джойстика відповідно до положення його ручки по вертикальній осі засвічувати світлодіоди середнього стовпця світлодіодної матриці. При встановленні ручки джойстика в нижнє положення світлодіодна стовбець світлодіодів погашений, при встановленні ручки джойстика в верхнє положення засвічено всі світло діоди стовпця. Початковий стан: засвічена кількість світлодіодів відповідно до положення ручки джойстика.
3	За допомогою джойстика відповідно до положення його ручки по вертикальній осі рухати засвіченим світлодіодом. При встановленні ручки джойстика в крайнє нижнє положення на світлодіодній лінійці засвічений крайній нижній світлодіод, при встановленні ручки джойстика в крайнє верхнє положення засвічений крайній верхній світлодіод. Початковий стан: засвічений світлодіод відповідно до положення ручки джойстика.
4	За допомогою джойстика відповідно до положення його ручки по вертикальній осі змінювати місце виведення символу «*» на другому рядку рідкокристалічного дисплею. При встановленні ручки джойстика в крайнє нижнє положення символ «*» виводиться на

	крайньому лівому знакомісці другого рядка. При встановленні ручки джойстика в крайнє верхнє положення символ «*» виводиться на крайньому правому знакомісці другого рядка. Початковий стан: символ виведений на знакомісці відповідно до положення ручки джойстика.
5	За допомогою джойстика відповідно до положення його ручки по горизонтальній осі засвічувати стовпці світлодіодів на матриці світлодіодів. При встановленні ручки джойстика в крайнє ліве положення світлодіодна матриця погашена, при встановленні ручки джойстика в крайнє праве положення засвічено всі стовпці світлодіодів. Початковий стан: засвічена кількість стовпців світлодіодів відповідно до положення ручки джойстика.
6	На статичному індикаторі реалізувати біжучу стрічку, на якій по черзі засвічуються символи на 1, 2, 3 і 4 знакомісті відповідно починаючи з лівого, виводиться (“1”, “2”, “3”, “4”), зміна символів визначається положенням ручки джойстика відносно горизонтальної осі, ліве крайнє положення – символи змінюються з частотою 1 символ за 1с., крайнє праве положення – символи змінюються з частотою 8 символів за 1с. Початковий стан: символи змінюються з швидкістю відповідно до положення ручки джойстика.
7	На рідкокристалічному дисплеї виводити в першому рядку символи від «1» до «10» на відповідних знакомісцях в залежності від положення ручки джойстика на горизонтальній осі. Крайнє ліве положення ручки джойстика відповідає пустому рідкокристалічному дисплею. Крайнє праве положення ручки джойстика відповідає засвіченим усім символам від «1» до «10». Початковий стан: кількість символів виводяться відповідно до положення ручки джойстика.
8	На матричному індикаторі засвічувати з певним періодом символ “+”. Тривалість світіння символу залежить від положення ручки джойстика відносно вертикальної осі. Крайнє верхнє положення – час відображення символу 0,5с., інтервал погашеного індикатора 0,5с., крайнє нижнє – час відображення символу 4с., інтервал погашеного індикатора 4с. початковий стан: символ відображається з частотою відповідно до положення ручки джойстика.
9	На матричному індикаторі засвічувати певну кількість рядків відповідну до положення ручки джойстика по вертикальній осі. Верхнє крайнє положення – засвічено всі рядки матричного індикатора, нижнє крайнє положення – весь матричний індикатор погашено. Початковий стан: засвічено певну кількість рядків відповідно до положення ручки джойстика.
10	За допомогою джойстика керувати швидкістю переміщення точки по світлодіодній лінійці. Швидкість переміщення визначається положенням ручки джойстика відносно горизонтальної осі, крайнє ліве положення відповідає мінімальній швидкості (1 сегмент за 4 секунди), крайнє праве положення відповідає максимальній швидкості переміщення засвіченого світлодіода (1 сегмент за 0,5 секунди). Початковий стан: засвічений світлодіод переміщається по лінійці в напрямку зверху вниз з швидкістю відповідно до положення ручки джойстика.
11	За допомогою джойстика відповідно до положення його ручки по горизонтальній осі засвічувати світлодіоди. При встановленні ручки джойстика в крайнє ліве положення світлодіодна лінійка погашена, при встановленні ручки джойстика в крайнє праве положення засвічено всі світлодіоди. Початковий стан: засвічена кількість світлодіодів відповідно до положення ручки джойстика.
12	За допомогою джойстика відповідно до положення його ручки по горизонтальній осі рухати засвіченим світлодіодом. При встановленні ручки джойстика в крайнє ліве положення на світлодіодній лінійці засвічений крайній нижній світлодіод, при встановленні ручки джойстика в крайнє праве положення засвічений крайній верхній світлодіод. Початковий стан: засвічений світлодіод відповідно до положення ручки джойстика.
13	За допомогою джойстика відповідно до положення його ручки по горизонтальній осі змінювати місце виведення символу «!» на третьому рядку рідкокристалічного дисплею.

	При встановленні ручки джойстика в крайнє ліве положення символ «!» виводиться на крайньому лівому знакомісті третього рядка. При встановленні ручки джойстика в крайнє праве положення символ «!» виводиться на крайньому правому знакомісті третього рядка. Початковий стан: символ виведений на знакомісті відповідно до положення ручки джойстика.
14	За допомогою джойстика відповідно до положення його ручки по вертикальній осі засвічувати рядки світлодіодів на матриці світлодіодів. При встановленні ручки джойстика в нижнє положення світлодіодна матриця погашена, при встановленні ручки джойстика в верхнє положення засвічено всі рядки світлодіодів. Початковий стан: засвічена кількість рядків світлодіодів відповідно до положення ручки джойстика.
15	На статичному індикаторі реалізувати біжучу стрічку, на якій по черзі засвічуються символи на 4, 3, 2 і 1 знакомісті відповідно починаючи з правого, виводиться (“9”, “8”, “7”, “6”), зміна символів визначається положенням ручки джойстика відносно вертикальної осі, верхнє крайнє положення – символи змінюються з частотою 1 символ за 1с., крайнє нижнє – положення символи змінюються з частотою 8 символів за 1с. Початковий стан: символи змінюються з швидкістю відповідно до положення ручки джойстика.
16	На рідкокристалічному дисплеї виводити в червертому рядку символи від «1» до «10» на відповідних знакомістях в залежності від положення ручки джойстика на вертикальній осі. Крайнє нижнє положення ручки джойстика відповідає пустому рідкокристалічному дисплею. Крайнє верхнє положення ручки джойстика відповідає засвіченим усім символам від «1» до «10». Початковий стан: кількість символів виводяться відповідно до положення ручки джойстика.
17	На матричному індикаторі засвічувати з певним періодом символ “0”. Тривалість світіння символу залежить від положення ручки джойстика відносно горизонтальної осі. Праве крайнє положення – час відображення символу 0,5с., інтервал погашеного індикатора 0,5с., крайнє ліве – час відображення символу 4с., інтервал погашеного індикатора 4с. початковий стан: символ відображається з частотою відповідно до положення ручки джойстика.
18	На матричному індикаторі засвічувати певну кількість стовпців відповідну до положення ручки джойстика по горизонтальній осі. Ліве крайнє положення – засвічено всі стовпці матричного індикатора, праве крайнє положення – весь матричний індикатор погашено. Початковий стан: засвічено певну кількість стовпців відповідно до положення ручки джойстика.

СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ

1. Рябенський В.М., Жуйков В.Я., Гулий В.Д. Цифрова схемотехніка: Навчальний посібник / Львів : Новий Світ-2000, 2009 – 736 с.
2. Кравець В.О., Сокол Є.І., Рисований О.М. Комп'ютерна схемотехніка. Підручник. – Х.: НТУ "ХПІ", 2007. – 480 с.
3. Рисований О.М., Грушенко М.В. Цифрові пристрої та мікропроцесори. Архітектура та програмне забезпечення: Навчальний посібник. – Х.: ХУПС, 2005. – 384 с.
4. Рисований О.М., Стасєв Ю.В. Комп'ютерна схемотехніка / За ред. О.М. Рисованого: Навчальний посібник. – Х.: ХУПС, 2007. – 332 с.
5. Бойко В.І., Гуржій А.М. та ін. Схемотехніка електронних систем: У 3кн. Кн. 3. Мікропроцесори та мікроконтролери. К. : Вища школа., 2004.- 339 с.
6. Якименко Ю.І., Терещенко Т.О., Сокол Є.І., Петегеря Ю.С. Мікропроцесорна техніка. К. "Кондор", 2004.- 440 с.
7. Угрюмов Е.П. Цифровая схемотехника. – СПб, 2000 – 528 с.
8. Калабеков В.А. Цифровые устройства и микропроцессорные системы: Учебник для техникумов связи. – М. : Горячая линия – Телеком, 2000. – 336 с.: ил.
9. Цифровая и вычислительная техника: Учебник для вузов / Евреинов Э.В., Бутыльский Ю.Т., Мамзелев И.А. и др. ; Под. ред. Э.В. Евреинова – М. : Радио и связь. – 1991 – 464 с.
10. Паламар М.І., Стрембіцький М.О., Паламар А.М. Проектування комп'ютеризованих вимірювальних систем і комплексів. Навчальний посібник. Тернопіль: ТНТУ. 2019. 150 с.
11. Паламар А. Програмно-апаратний комплекс для дистанційного моніторингу стану джерел безперебійного електроживлення. Матеріали VII науково-технічної конференції Тернопільського національного технічного університету імені Івана Пулюя "Інформаційні моделі, системи та технології", 11-12 грудня 2019 р. Тернопіль: ТНТУ, 2019. с. 132.
12. Паламар М.І., Паламар А.М. Методичні вказівки до лабораторних робіт та курсового проектування робіт з дисципліни: "Комп'ютерна схемотехніка" для студентів денної та заочної форми навчання напряму підготовки 6.05010201 «Комп'ютерна інженерія». Тернопіль: ТНТУ, 2015. 95 с.
13. Паламар М.І., Чайковський А.В., Пастернак Ю.В., Стрембіцький М.О., Паламар А.М. Проектування пристроїв і вузлів інформаційно-вимірювальних систем та створення програмного забезпечення на базі навчально-налагоджувального стенда. Методичні вказівки до лабораторних і практичних робіт з дисциплін «Проектування інформаційно-вимірювальних систем», «Мікропроцесори і ЕОМ», «Проектування приладів і систем на основі мікроконтролерів». Тернопіль: ТНТУ, 2014. 76 с.
14. Дистанційний курс "Комп'ютерна схемотехніка", доступний за адресою <https://dl.tntu.edu.ua/bounce.php?course=1990>.