

МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ
Національний університет «Запорізька політехніка»

Інститут інформатики та радіоелектроніки
Факультет радіоелектроніки та телекомунікацій
(повне найменування інституту, факультету)

Кафедра радіотехніки та телекомунікацій
(повне найменування кафедри)

Пояснювальна записка

до дипломного проекту (роботи)

бакалавра

(ступінь вищої освіти)

на тему ДИСТАНЦІЙНЕ КЕРУВАННЯ ІНФОРМАЦІЙНОЮ
ПАНЕЛЛЮ

Виконав: студент(ка) 4 курсу, групи РТ-917

Спеціальності _____
172 «Телекомунікації та радіотехніка»

(код і найменування спеціальності)

Освітня програма (спеціалізація)
«Інформаційні мережі зв'язку»

Булавчик Вадим Павлович
(прізвище та ініціали)

Керівник Костенко В.О.
(прізвище та ініціали)

Рецензент Касьян М.М.
(прізвище та ініціали)

2021 рік

МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ
Національний університет «Запорізька політехніка»
 (повне найменування закладу вищої освіти)

Інститут, факультет Інститут інформатики та радіoeлектроніки, ФРЕТ
 Кафедра Радіотехніки та телекомунікацій
 Ступінь вищої освіти Бакалавр
 Спеціальність 172 «Телекомунікації та радіотехніка»
 (код і найменування)
 Освітня програма (спеціалізація) Інформаційні мережі зв'язку
 (назва освітньої програми (спеціалізації))

ЗАТВЕРДЖУЮЗавідувач кафедри РТТ к.т.н., доц. Морщавка С.В.« » травня 20 21 року

ЗАВДАННЯ
НА ДИПЛОМНИЙ ПРОЄКТ (РОБОТУ) СТУДЕНТА(КИ)

Булавчику Вадиму Павловичу

(прізвище, ім'я, по батькові)

1. Тема проєкту (роботи) Дистанційне керування інформаційною панеллюкерівник проєкту (роботи) Костенко Валер'ян Остапович, доцент

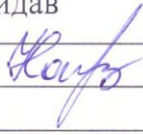
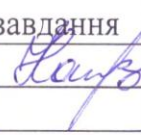
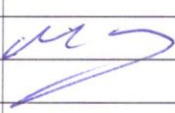
(прізвище, ім'я, по батькові, науковий ступінь, вчене звання)

затверджені наказом закладу вищої освіти від «30» квітня 2021 року № 1642. Строк подання студентом проєкту (роботи) 27 травня 2021 р.3. Вихідні дані до проєкту (роботи) Проектування алгоритмів, схем і програм керування інформаційною панеллю, яка реалізована на мікроконтролері «Atmega32-16AU»4. Зміст розрахунково-пояснювальної записки (перелік питань, які потрібно розробити) Зробити огляд існуючих аналогічних панелей, врахувати їх недоліки, запропонувати свій алгоритм керування

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

Презентація 16 слайдів

6. Консультанти розділів проекту (роботи)

Розділ	Прізвище, ініціали та посада Консультанта	Підпис, дата	
		завдання видав	прийняв виконане завдання
1-4	Костенко В.О. к.т.н., доцент		
Нормокон- троль	Мороз Г.В., ст. викладач		

7. Дата видачі завдання « 02 » квітня 20 21 року.

КАЛЕНДАРНИЙ ПЛАН

№ з/п	Назва етапів дипломного проекту (роботи)	Строк виконання етапів проекту (роботи)	Примітка
1	Пошук та вивчення літературних джерел	05.04-12.04	
2	Пошук та розгляд існуючих аналогів	12.04-19.04	
3	Синтез схеми електричної принципової	19.04-02.05	
4	Побудова алгоритмів і програм	02.05-8.05	
5	Розрахунок надійності	08.05-13.05	
6	Оформлення пояснювальної записки	13.05-20.05	
7	Перевірка ПЗ на доброчесність	20.05-21.05	
8	Створення презентації у Power Point	21.05-26.05	
9	Захист	4.06	

Студент(ка)

Керівник бакалаврської роботи

 Булавчик В.П.
(прізвище та ініціали)

 Костенко В.О.
(прізвище та ініціали)

РЕФЕРАТ

ПЗ: 150 сторінок, 19 рисунків, 17 таблиць, 10 джерел

Об'єкт дослідження – світлодіодний модуль відображення графічної інформації з керуванням по радіоканалу.

Мета роботи – розробка і дослідження світлодіодного модуля відображення графічної інформації з керуванням по радіоканалу у варіанті вуличного інформаційного табло для відображення текстової інформації, поточного часу і дати, температури навколишнього середовища.

Метод дослідження ґрунтується на аналізі побудови аналогічних пристроїв, реалізації їх внутрішніх програм і схемотехнічних рішень. За результатами проведених досліджень було виявлено ряд схемотехнічних і програмних недоліків досліджених пристроїв які були враховані при розробці даного пристрою.

В ході роботи над дипломним проектом був розроблений пакет конструкторської документації, що включає в себе: схеми електричні структурні, схеми електричні принципів, креслення деталей, складальне креслення, алгоритми роботи програм і програми.

ІНФОРМАЦІЙНЕ ТАБЛО, СТАТИЧНА ІНДИКАЦІЯ,
СВІТЛОДІОДНИЙ МАСИВ, ДРАЙВЕР СВІТЛОДІОДНИЙ,
МІКРОКОНТРОЛЕРИ, ГОДИННИК РЕАЛЬНОГО ЧАСУ, РАДІОКАНАЛ

ЗМІСТ

ПЕРЕЛІК СКОРОЧЕНЬ.....	7
ВСТУП	8
1 АНАЛІЗ ТЕХНІЧНОГО ЗАВДАННЯ, РОЗГЛЯД ІСНУЮЧИХ АНАЛОГІВ .	10
2 ПРОЕКТУВАННЯ СТРУКТУРНОЇ СХЕМИ, ОПИС ОСНОВНИХ БЛОКІВ СХЕМИ.....	12
2.1 Структурна схема пульта керування.....	12
2.2 Структурна схема світлодіодного модуля відображення графічної інформації з управлінням через радіоканал	14
3 СИНТЕЗ СХЕМИ ЕЛЕКТРИЧНОЇ ПРИНЦИПОВОЇ.....	20
3.1 Розробка схеми електричної принципової	20
3.1.1 Принципова схема пульта дистанційного керування світлодіодним модулем відображення графічної інформації	20
3.1.2 Принципова схема блоку контролера світлодіодного модуля відображення графічної інформації	28
3.1.3 Принципова схема блоку індикації світлодіодного модуля графічної інформації	31
3.2 Вибір і обґрунтування елементної бази	33
3.2.1 Конденсатори.....	34
3.2.2 Резистори	34
3.2.3 Комутаційні елементи	35
3.2.4 Діоди.....	35
3.2.5 Транзистори	37
3.2.6 Мікросхеми	37
3.2.7 Модуль радіоканалу.....	40
3.2.8 Роз'єми	41
3.2.9 Індикатор.....	42
3.2.10 Кварцові резонатори	42
4 ПОБУДОВА АЛГОРИТМІВ І ПРОГРАМ	43

4.1 Побудова алгяриту програми пульта управління	43
4.2 Побудова алгоритму програми блоку контролера світлодіодного модуля відображення графічної інформації	45
4.3 Побудова програмного коду для мікроконтролера пульта управління ..	46
4.4 Побудова програмного коду для мікроконтролера блоку контролера світлодіодного модуля відображення інформації.....	55
5 РОЗРАХУНОК НАДІЙНОСТІ	57
5.1 Теоретична частина.....	57
5.2 Розрахункова частина	60
ВИСНОВКИ.....	71
ПЕРЕЛІК ПОСИЛАНЬ	72
Додаток А	73
Додаток Б.....	74
Додаток В	132

ПЕРЕЛІК СКОРОЧЕНЬ

PKI	–	рідкокристалічний індикатор
ASCII	–	(American standard code for information interchange) американський стандартний код для обміну інформацією
EEPROM	–	Electrically Erasable Programmable Read-Only Memory
RS232	–	recommended standard 232
RTC	–	real time clock (годинник реального часу)
UART	–	universal asynchronous receiver/transmitter (універсальний асинхронний прийомопередавач)

ВСТУП

Все частіше на ринку зовнішньої реклами використовують інформаційні панелі – табло «Рухомий рядок». Це викликано рядом причин, таких як : висока інформативність, динамічність, яскравість, можливість швидкої зміни інформації. В багатолюдних місцях, там, де є трамвайні, тролейбусні, маршрутні зупинки, метро, майданчики, парки, де потенційний клієнт потребує інформації, рухомі рядки є ефективним інструментом реклами.

Більшість людей вважає, що рухомі рядки – це застарілий вид реклами, він своє віджив, він не кому не цікавий, велика ціна, а це лише світлодіодне табло, легше купити плазмову панель.

По-перше елементи рухомих рядків – світлодіоди, не так давно використовуються в масовому виробництві, раніше світлодіод служив джерелом індикації, зараз же світлодіод – джерело світла. Над яскраві світлодіоди та світлодіоди синього, білого кольору були запуснені в виробництво лише на початку 21 сторіччя. І лише в останні роки світлодіоди стали використовувати в рекламі.

Зараз кількість встановлених табло в великих містах України не така вже й велика, вони не стоять на кожному кроці, як лайтбокси, вивіски та інші «стандартні» види зовнішньої реклами. В інших країнах дуже часто можна побачити вулицю, обвішану рухомими рядками та світлодіодними екранами.

Плазмові панелі це добре. Але розміри панелі, її яскравість не дозволяють використовувати її в зовнішній рекламі. Ці панелі можна використовувати лише всередині приміщень. До того ж, плазмова панель – телевизор – монітор. З психологічної точки зору на такий вид реклами можуть і не звернути увагу, тому що цього всього вистачає в повсякденному житті, на роботі, телевизор як телевизор.

Інша справа яскраве, динамічне вуличне табло, яке несе в собі корисну інформацію, наприклад поточний час, дату, температуру навколишнього

середовища та ін. Люди дуже швидко звикають до такої інформації , наступного разу, проходячи повз, обов'язково звернуть увагу на табло.

1 АНАЛІЗ ТЕХНІЧНОГО ЗАВДАННЯ, РОЗГЛЯД ІСНУЮЧИХ АНАЛОГІВ

Світлодіодний модуль відображення графічної інформації з управлінням через радіоканал повинен мати наступні технічні параметри:

- змінюване поле масиву;
- вивід кириличних и латинських букв великого та малого регістрів, вивід цифр, розділових знаків та спеціальних символів;
- вивід температури навколишнього середовища;
- вивід поточної дати та часу;
- управління за допомогою персонального комп'ютера;
- дистанційне керування за допомогою пульта керування;
- розмір тексту, що вводиться – до 32000 символів;\
- висота символу 8 пікселів;
- видимість при прямих сонячних променях;
- робота в статичному режимі;
- оптимальна дистанція видимості 5 – 15 метрів.

Пристрій, що мною розробляється призначено для використання в зовнішній та внутрішній рекламі, може використовуватись як самостійний пристрій, або входити в склад складних рекламних конструкцій, може доповнювати старі вивіски з метою підвищення їх привабливості.

На початковому етапі проектування цього пристрою був проведений порівняльний аналіз ринку рекламної продукції з метою вибору аналогів. За результатами аналізу був вибраний легкодоступний на території України аналог виробництва фірми «Silima» модель RSXL 8-112. Цей аналог найбільш близько підходить за параметрами відносно пристрою, що розробляється.

Основні характеристики аналога фірми «Silima» модель RSXL 8-112 приведені нижче:

- висота символу 8 пікселів;
- вуличне застосування;

- вивід даних за допомогою персонального комп'ютера;
- пам'ять 32000 символів;
- ціна 650\$.

Проаналізувавши структуру побудови цього аналога була визначена певна кількість недоліків, котрі надалі були усунені. До недоліків можна віднести:

- використання динамічної індикації, що призводить до зниження яскравості світлодіодів (через велику частоту перемикання);
- збільшення робочого струму світлодіодів в 4-5 разів для збільшення їх яскравості;
- наявність тільки провідного зв'язку з джерелом відправки інформації (персональний комп'ютер);
- відносно велика ціна.

В нашому випадку головним недоліком є відсутність дистанційного керування.

2 ПРОЕКТУВАННЯ СТРУКТУРНОЇ СХЕМИ, ОПИС ОСНОВНИХ БЛОКІВ СХЕМИ

При розробці пристрою першим етапом є розробка структурної схеми, згідно якої будується електрична принципова схема.

2.1 Структурна схема пульта керування

Центральним блоком структурної схеми є мікроконтролер, який керує іншими елементами схеми. Повинен мати достатню кількість портів введення-виведення. Мати достатній об'єм флеш пам'яті програм, і володіти необхідною швидкодією.

Оскільки пристрій є портативним в пульті управління необхідний модуль живлення із влаштованим акумулятором. Він повинен забезпечувати безперебійну роботу пульта керування, заряд влаштованого акумулятора і роботу від мережі з використанням будь якого стандартизованого блока живлення з вихідною напругою 10-18 вольт. За результатами проведення випробувань блоків живлення найбільш відповідним є імпульсний модуль живлення виготовлений у вигляді мережевої вилки виробництва MeanWell модель ES25E15-150.

Для роботи пристрою, що проектується, без підключення комп'ютерної клавіатури, а також для керування пультом керування і навігації всередині програмної середи було вирішено використовувати влаштовану клавіатуру.

Для відображення введеного за допомогою комп'ютерної клавіатури інформаційного тексту, котрий буде передано в модуль відображення інформації, відображення введеного раніше тексту, котрий зберігається в пам'яті пульта керування необхідний модуль індикації. Також він потрібний для відображення внутрішньої програмної оболонки навігації усередині неї за допомогою обумовленої вище клавіатури. Модуль індикації повинен забезпечувати виведення букв російської, української мови а також латинські

символи і цифри десяткової системи числення. Для роботи у вечірній або нічний час має бути передбачене підсвічування дисплея, що вмикається і вимикається для економії заряду акумуляторної батареї.

Зовнішня EEPROM пам'ять пульта управління призначена в першу чергу для зберігання тексту, що вводиться, а також зберігання деяких програмних значень і змінних. Вона повинна мати достатній об'єм для зберігання трьох алфавітно-цифрових віртуальних файлів, один з яких в подальшому буде відправлено до модуля відображення. Для кожного повідомлення має бути передбачене місце для мінімум 32000 символів і програмних кодів. Має бути енергонезалежна.

У пульті керування передбачений системний годинник з можливістю встановлення часу в модулі відображення. З метою незалежності системного годинника і зменшення навантаження на мікроконтролер було вирішено використовувати не програмну реалізацію ходу годинника, а використовувати для цих цілей спеціалізований модуль годинника реального часу, який опитується в певні моменти часу вносячи мінімальне навантаження на центральний мікроконтролер. Забезпечує незалежний від електроживлення і інших системних задач точний хід часу.

Щоб організувати зв'язок по радіоканалу зі світлодіодним модулем відображення інформації встановлюється модуль прийому-передачі. Він повинен забезпечувати безпосередній канал радіозв'язку з кодуванням переданих даних з метою підвищення правильності прийнятих даних на приймальній стороні. Повинна забезпечуватися висока швидкість передачі даних. Для простоти організації механічних елементів антена повинна бути вбудована. Мінімальна дальність радіозв'язку повинна бути не менше 15 метрів.

2.2 Структурна схема світлодіодного модуля відображення графічної інформації з управлінням через радіоканал

Існує два варіанти побудови структурних схем світлодіодних модулів відображення інформації: динамічний і статичний.

Світлодіодні модулі відображення інформації з динамічним принципом відображення інформації дозволяють працювати світлодіодам дуже короткий проміжок часу (рис. 2.1). Це більше схоже на мерехтіння ніж на світіння. Навіть якщо виробник використовує дуже яскравий світлодіод, з максимальним кутом випромінювання, все одно всі ці переваги будуть видні дуже короткий проміжок часу. Таке табло має яскравість мінімум в 4-8 разів менше від потенційно можливої яскравості світлодіода. При сонячному світлі з такого табло неможливо прочитати інформацію, отже, рекламна компанія стає не ефективною! Для наглядності наведемо спрощену схему блоку індикації динамічного табло, дозвіл 128x8 пікселя, всього 1024 світлодіода.

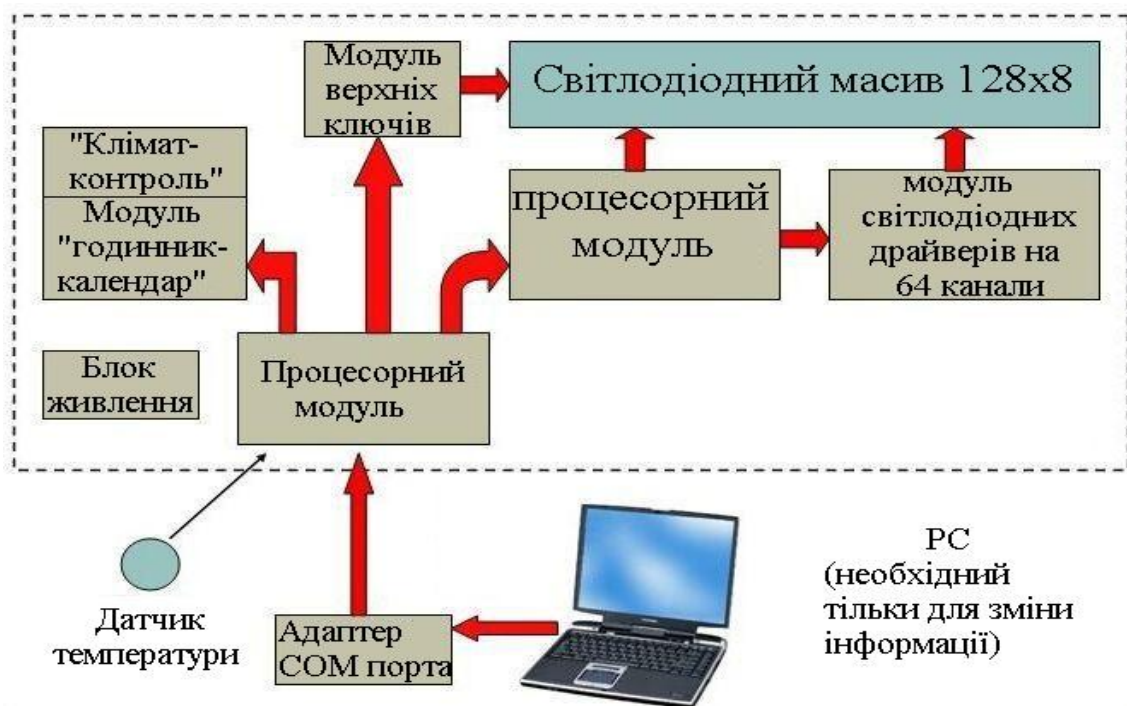


Рисунок 2.1 – Динамічна індикація

Світлодіоди в схемі з'єднані у вигляді матриці, яка має, 8 виходів рядків і 8 виходів стовпців. В буфер пам'яті завантажують інформацію і подають її на стовпці, при цьому на перший рядок, за допомогою електронних ключів, подають живлення. Згодом засвічується перший рядок, відповідно до завантажених даних в буфер пам'яті, який управляє стовпцями. Таким чином, комутуючи всі електронні ключі з частотою, наприклад 75 герц, і в кожен момент часу подаючи нову інформацію на стовпці, через підрядник формується зображення. Насправді ж в один момент часу світитися тільки один рядок світлодіодів, але так як людське око інерційне, то перемикання цих рядків з різними комбінаціями світіння, з частотою 75 герц, створює враження єдиного зображення, при цьому світлодіоди світяться в 1/8 від реальної яскравості. Цей недолік менш видно всередині приміщення, але для ефективної зовнішньої реклами він не допустимий.

Світлодіодні модулі відображення інформації виконані зі статичною схемою відображення значно складніші, і вимагають більш дорогих комплектуючих, а також друкованих плат підвищеної точності. Тому, в результаті, виходить дорожче не тому, що застосовані інші світлодіоди, а від того, що на її розробку і виготовлення потрібна велика кількість електронних компонентів, ресурсів і часу. В кінцевому підсумку світлодіоди не мерехтять, а працюють безперервно. Таке табло дуже добре видно при сонячному світлі (навіть при прямих променях)! Чим яскравіше табло, тим більше людей його побачать. Побачать навіть з далекої відстані.

На рисунку 2.2 приведена схема зі статичної індикацією. У статичних схемах живлення на світлодіоди подається постійно, а їх управління здійснюється окремим каналом буфера.

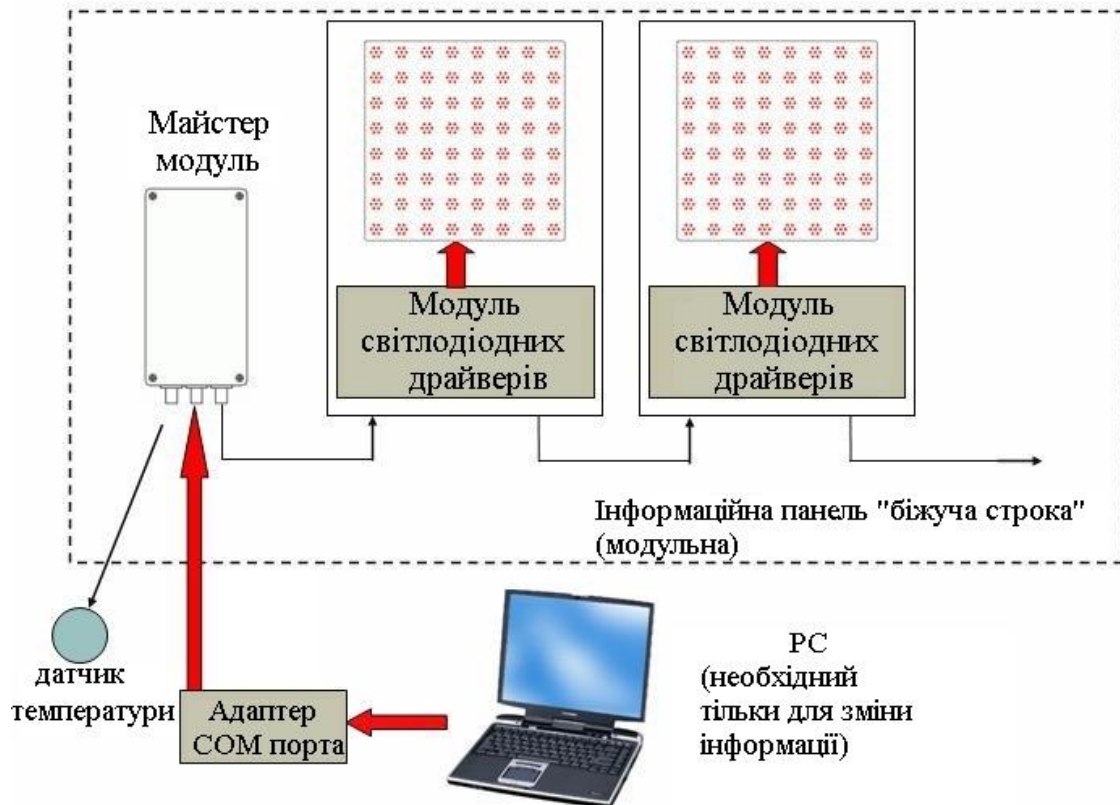


Рисунок 2.2 – Статична індикація

Порівняльна відмінність між статичною і динамічною індикацією:

а) нормальний струм живлення світлодіода зазвичай становить 20 мА, а так як в динаміці кожен світлодіод світитися всього 1/8 часу повного циклу розгортки, то грубо кажучи, яскравість світлодіода 1/8 від реальної його яскравості. У статистиці живлення на світлодіоди подається постійно і окремо на кожен світлодіод, при цьому яскравість реальна, гарантована виробником. І все ж в динаміці можна збільшити яскравість, живлячи світлодіоди підвищеними струмами в межах 100 мА, в цьому випадку світлодіод працює у форсованому режимі, при цьому відбувається швидка деградація кристала (зменшення яскравості в процесі роботи), але візуальна яскравість все одно нижче паспортної;

б) в динамічному режимі світлодіоди з'єднані в рядки і в стовпці, тому при виході з ладу одного з комутуючих ключів або одного з каналів буфера, не

працює вся рядок або стовпець світлодіодів. У статиці ж при виході каналу буфера з ладу не працює всього один світлодіод (піксель);

в) кількість електронних елементів в схемі динаміки менше ніж в статиці. Наприклад, біжучий рядок дозволом 8х96 пікселів. для динаміки досить, 8 електронних ключів, 12 зсувних регістрів (буферів). Для статики необхідно 96 зсувних регістрів (буферів). Економія на динаміці очевидна;

г) потужність, для однакових табло дозволом 8х96 пікселів, в 4-8 разів менше для динаміки і в 4-8 разів більше для статики відповідно;

д) в динамічних біжучих строках частота розгортки вважається нормальною 75Гц, при менших частотах помітно мерехтіння, швидко втомлюються очі. При цьому яскравість в 3-4 рази візуально менше потенційно можливої яскравості світлодіодів. Динамічну розгортку зазвичай використовують в невеликих біжучих строках для офісу і транспорту, а також в строках до 30см у висоту на вулиці при непрямому попаданні сонячних променів.

Виходячи з перерахованих вище переваг і недоліків, для розроблювального пристрою була обрана структурна схема світлодіодного модуля відображення графічної інформації з керуванням по радіоканалу зі статичної індикацією, до складу якої повинні входити нижче перераховані блоки.

Для управління блоком контролера світлодіодного модуля відображення графічної інформації призначений центральний мікроконтролер.

У разі безпосереднього зв'язку світлодіодного модуля відображення інформації з персональним комп'ютером або ноутбуком через інтерфейс RS232 призначений блок інтерфейсу RS232. Блок інтерфейсу повинен перетворювати рівень сигналів інтерфейсу RS232 в безпечні і зрозумілі мікроконтролеру сигнали рівня TTL.

Зовнішня EEPROM пам'ять призначена для зберігання в ній тексту і системних команд з виведення додаткових даних, що приймаються з пульта управління або з персонального комп'ютера тексту і системних команд з

виведення додаткових даних. Повинна бути енергонезалежна, володіти достатнім об'ємом пам'яті, мати велику швидкість зв'язку з мікроконтролером.

Також як і в пульті управління світлодіодним модулем відображення графічної інформації, модуль годинника реального часу призначений для незалежного ходу годинника, ослаблення програмного навантаження на мікроконтролер. Призначений для виведення, за бажанням власника, поточного часу і (або) дати на масив відображення. Коригування ходу годинника, установка поточного часу і дати проводиться за допомогою пульта управління.

Температурний датчик призначений для перетворення температури повітря навколишнього середовища в цифровий вигляд для подальшого перетворення даного цифрового коду в значення температури, з подальшим виведенням даного значення реальної температури на світлодіодний масив.

Оскільки дані з мікроконтролера зі світлодіодними драйверами з'єднується за допомогою дротових ліній (загальний шлейф) який навіть при малих довжинах вносить значні втрати, а також за рахунок власної ємності спотворює фронти сигналів даних, необхідний буферний каскад посилення сигналів даних і синхронізації.

Модуль прийому-передачі призначений для зв'язку з пультом управління по спеціальному каналу зв'язку. Повинен володіти високою чутливістю, мати хорошу перешкодозахищеність. Швидкість прийому або передачі даних повинна бути не менше 9600 біт/сек., повинен мати вбудовану антену для спрощення побудови механічної конструкції. Здійснює прийом даних (текст і системні команди) з пульта управління. Найкращий варіант використання готового модуля з вбудованими протоколами передачі і підключається безпосередньо до мікроконтролера по інтерфейсу UART.

Модуль світлодіодного драйвера призначений для управління світлодіодним масивом. Він підключається через буферний каскад до мікроконтролеру. По суті він являє собою зсувний регістр з стабілізованим вихідним струмом на кожному з шістнадцяти виходів. До кожного виходу

світлодіодного драйвера підключається катод – один світлодіод або один ланцюжок.

Світлодіодний масив являє собою шістнадцять світлодіодів або ланцюжків які конструктивно розташовані в один рядок, аноди яких з'єднані на загальний плюс, а катоди підключені до світлодіодних драйверів.

З СИНТЕЗ СХЕМИ ЕЛЕКТРИЧНОЇ ПРИНЦИПОВОЇ

3.1 Розробка схеми електричної принципової

3.1.1 Принципова схема пульта дистанційного керування світлодіодним модулем відображення графічної інформації

Основними елементами схеми світлодіодного модуля відображення графічної інформації з керуванням по радіоканалу є мікроконтролери.

На сьогоднішній день основними виробниками мікроконтролерів є корпорації "Microchip", "Atmel", "Intel", "Texas Instruments", які виробляють безліч спеціалізованих мікроконтролерів і мікроконтролерів загального призначення. За даними статистики лідером з постачання мікроконтролерів на територію України є корпорація "Atmel", виходячи з цього більшість підприємств України використовують в своїх виробках мікроконтролери даної корпорації, враховуючи їх доступність і невисоку вартість. З огляду на сказане вище для розроблювального пристрою були обрані мікроконтролери корпорації "Atmel".

Основні критерії вибору мікроконтролера:

- кількість ліній введення- виведення даних;
- тип пам'яті (EPROM, Flash);
- об'єм пам'яті програм;
- об'єм ОЗУ;
- наявність спеціальних інтерфейсів (UART, USART, I2C, SPI);
- кількість ліній введення-виведення;
- наявність в мікроконтролері необхідних функціональних модулів;
- частота тактового генератора;
- робоча напруга.

Для розроблювального пристрою була встановлена мікроконтролер корпорації "Atmel" ATmega32-16AU (DD4). Оскільки він відповідає всім

вимогам для забезпечення надійної роботи пристрою, має достатній об'єм Flash пам'яті програм (необхідний мінімальний розмір пам'яті програм становить близько 14 кбіт), необхідну кількість ліній введення-виведення, необхідні стандартні інтерфейси I²C і USART для зв'язку з іншими елементами схеми такими як зовнішня EEPROM пам'ять і RTC (годинник реального ходу), модульний блок передачі даних:

- технічні характеристики мікроконтролера ATmega32 - 16AU;
- 32 лінії введення - виведення даних;
- 32 кбіт Flash пам'яті, 16 кбіт відведено для пам'яті програм;
- 2 кбіт SRAM;
- 1024 біт EEPROM;
- інтерфейси USART, I²C, SPI;
- частота тактового генератора до 16 МГц;
- робоча напруга – 4,0-5,0 В.

Структурна схема мікроконтролера ATmega32-16AU представлена на рисунку 3.1.

Для правильної роботи центрального мікроконтролера встановлені гарантовані одиниці: резистор підтяжки лінії скидання мікроконтролера R4, опір якого за специфікацією корпорації ATMEL повинен становити 10 кОм +/- 10% (рис. 3.2). Для роботи внутрішнього тактового генератора мікроконтролера з високою стабільністю частоти, що необхідно для правильної роботи деяких елементів, що підключаються до мікроконтролеру були встановлені кварцовий резонатор ZQ2 і конденсатори C3, C4 призначені для більш стабільного запуску внутрішнього генератора. Номінал кварцового резонатора може знаходитись в межах від 32,768 кГц до 16 МГц. Номінал визначається в залежності від необхідної швидкодії і для даної конструкції використовується кварцовий резонатор частотою 12 МГц. Ємність конденсаторів C3 і C4 стандартизована і становить 22пФ (рис. 3.3).

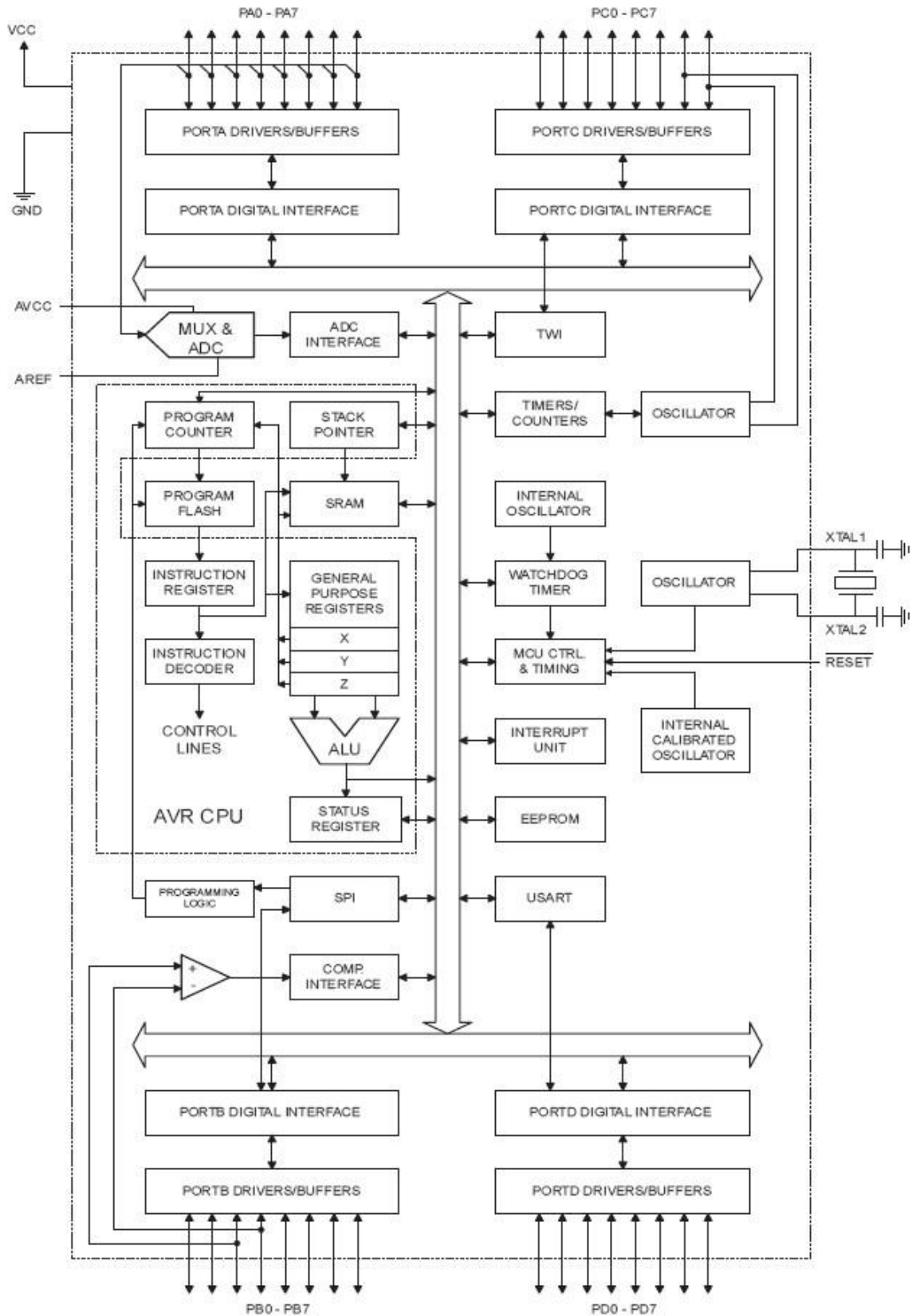


Рисунок 3.1 – Структурна схема мікроконтролеру ATmega 32-16AU

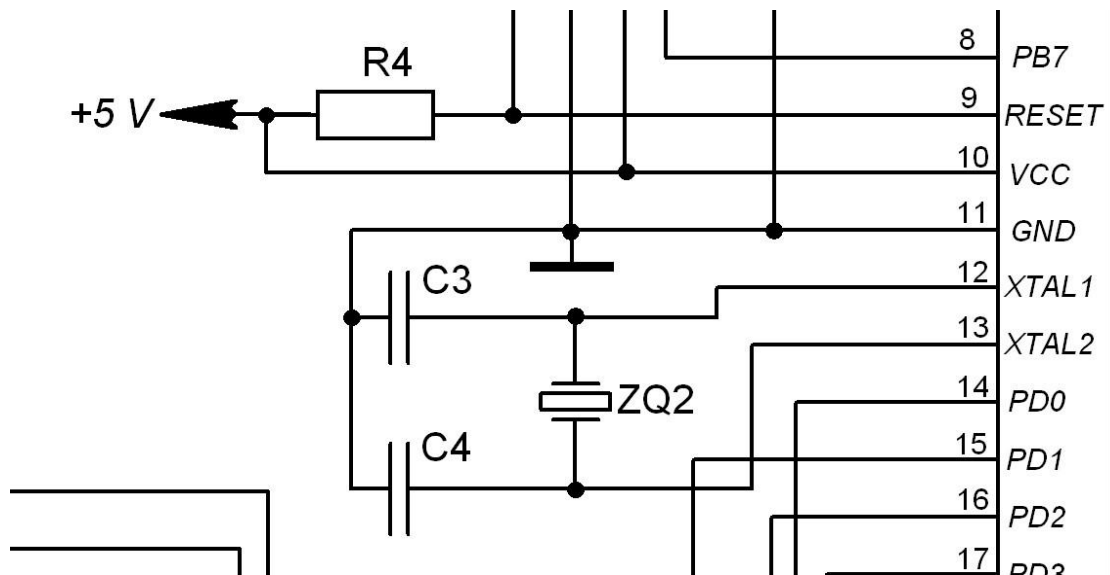


Рисунок 3.2 – Реалізація резистору підтяжки R4 і зовнішніх елементів тактового генератора C3, C4, ZQ2

На структурній схемі передбачений блок годинника реального часу. У процесі аналізу схемотехнічних рішень побудови подібних блоків було вирішено використовувати спеціалізовану мікросхему DD3 (рис. 3.2). В обв'язці даної мікросхеми передбачені кварцовий резонатор ZQ1 забезпечує тактові імпульси з частотою 32,768 кГц, також встановлюється елемент резервного живлення, акумуляторний елемент GB1 який забезпечує безперервний відлік часу в відсутності електроживлення.

Мікросхема DD2 (рис 3.3) є мікросхемою пам'яті, і призначена для зберігання введених даних, схема включення даної мікросхеми стандартизована і визначається за технічною документацією, при цьому включенні визначено адресу мікросхеми A0.

Резистори R5 і R6 (рис. 3.3) підтягують лінії даних і лінії синхроімпульсів інтерфейсу I²C до лінії живлення +5 Вольт, що забезпечує більш надійну роботу даного інтерфейсу, виключає ефект випадкового брязкоту рівня по цих лініях, оскільки швидкість обміну даними по цьому інтерфейсу відносно висока

і становить 400 Гц. Номінал даних резисторів, в яких використовується інтерфейс I²C, рекомендується виробниками мікросхем і складає 4,7 кОм.

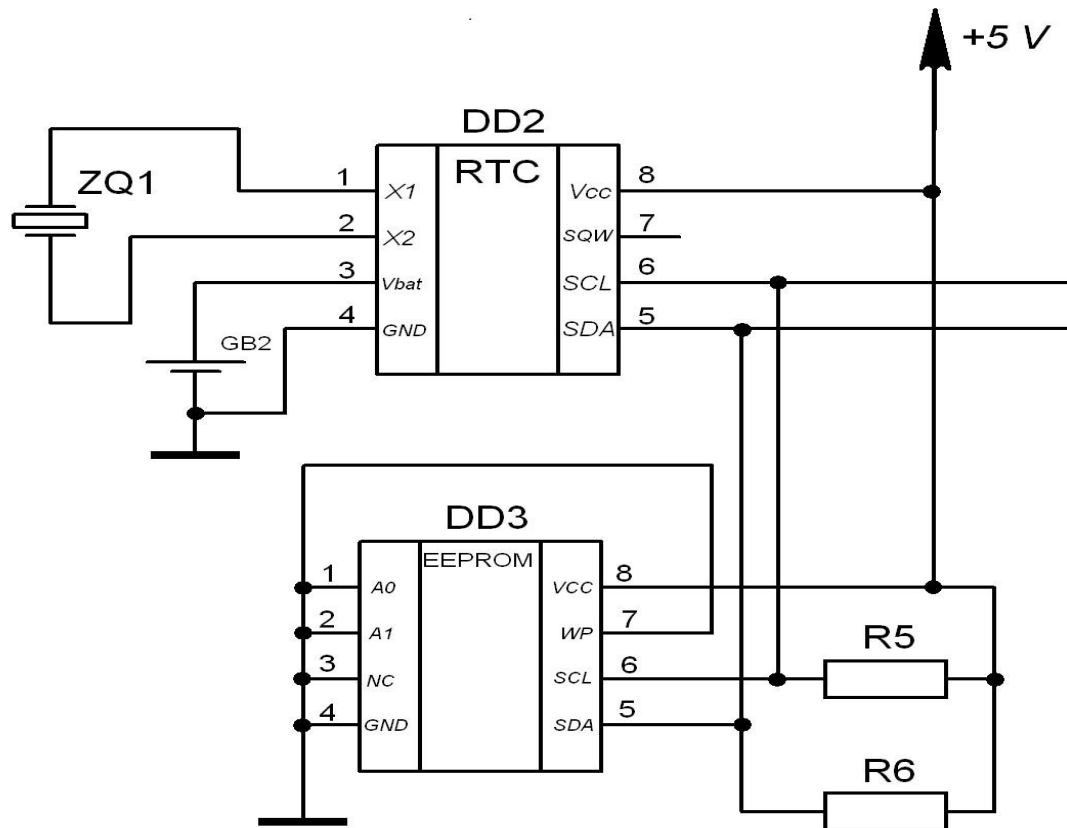


Рисунок 3.3 – Реалізація підключення зовнішньої пам'яті DD3 і годинника реального часу DD2, підключених до мікроконтролера по інтерфейсу I²C

Для відображення інформаційного тексту застосований рідкокристалічний індикатор з вбудованим драйвером HD44780. Для зменшення ліній зв'язку та економії ліній введення-виведення мікроконтролера використовується режим чотирьох бітного керування. Резистор R7 призначений для регулювання контрастності рідкокристалічного індикатора, в найпростішому випадку являє собою найпростіший дільник потенціалів, напруга знімається з середньої точки цього змінного резистора подається на рідкокристалічну матрицю всередині індикатора. Напруга +5 вольт необхідно

для живлення внутрішнього драйвера індикатора. Номінал резистора R7 обирається в межах 10 кОм-20 ком, рекомендований виробником і прийняте при розробці 10 кОм. Для роботи у вечірній і нічний час з пультом управління в індикаторі повинно бути передбачене підсвічування. У рідкокристалічних індикаторах застосовуються два види підсвічування: світлодіодна і люмінесцентна, враховуючи всі недоліки люмінесцентного підсвічування (необхідна напруга 100 вольт з частотою 400 Гц) вибір був зроблений в сторону світлодіодного підсвічування. Для організації її управління в схему введений польовий транзистор VT1 працює в ключовому режимі і керується мікроконтролером. Оскільки світлодіодне підсвічування розраховане на напругу 4,4 вольти а в схемі застосовано напругу 5 вольт встановлений обмежуючий резистор R8, значення якого визначається за формулою 3.1:

$$R8 = \frac{U - U_{\text{підсвіч}}}{I_{\text{підсвіч}}} \quad (3.1)$$

де $U_{\text{жив}}$ – напруга живлення основної схеми 5 В;

$U_{\text{підсвіч}}$ – напруга живлення підсвічування 4,4 В;

$I_{\text{підсвіч}}$ – струм, що споживання світлодіодного підсвічування 280 мА.

Підставивши значення отримуємо:

$$R8 = \frac{5-4,4}{0,28} = 2,14 \approx 2,2 \text{ Ом}, \quad (3.1)$$

Клавіатура є купленою та являє собою матрицю 3×4 з мембранних кнопок, підключається безпосередньо до мікроконтролера по семи лініях: три лінії стовпців і чотири лінії рядків. Ідентифікація натискання кнопок клавіатури здійснюється програмно в межах програмної оболонки.

У пульті керування передбачений модуль живлення (рис. 3.4).

Модуль живлення побудований з урахуванням всіх вимог які пред'являються до даного пристрою. Оскільки пульт керування повинен мати можливість працювати без наявності джерела живлення в ньому вбудована акумуляторна батарея. в зв'язку цим було встановлено вхідний стабілізатор напруги DA1 який забезпечує подачу безпечної для заряду літій-іонної акумуляторної батареї GB1 напруги. Акумуляторна батарея GB1 для нормальної і тривалої роботи пульта управління повинна мати ємність мінімум 800 мА / год. і забезпечувати вихідну напругу мінімум 6,2 Вольт. Напруга на виході DA1 має становити приблизно 10 вольт. Нижче наведені розрахунки резисторів R1 і R2 задають вихідну напругу цього стабілізатора:

$$V_{OUT} = 1,25 \times \left(1 + \frac{R1}{R2} \right), \quad (3.2)$$

де V_{OUT} – напруга на виході стабілізатора;

R2 – опір резистора R2 рекомендований 220 Ом.

В результаті отримуємо:

$$R1 = \left(\frac{V_{OUT} - 1,25}{1,25} \right) \times R2, \quad (3.3)$$

$$R1 = \left(\frac{10 - 1,25}{1,25} \right) \times 220 = 1540 \text{ Ом}.$$

Приймаємо значення опору відповідно ряду E24 $R1 = 1,5 \text{ кОм}$.

Оскільки всі елементи схеми розраховані на напругу 5 Вольт стабілізатор DA2 забезпечує на виході напругу 5 Вольт і струм до 1 Ампера.

Діод VD1 призначений для захисту пульта управління від неправильної подачі напруги, що може статися в результаті застосування не стандартизованого блоку живлення, який не пройшов технічні випробування.

Діоди VD2 і VD4 призначені для захисту стабілізатора DA1 від зворотної напруги акумуляторної батареї. Світлодіод VD3 призначений для відображення стану пульта управління: «Робота» → «Робота-заряд акумуляторної батареї» → «Заряд акумуляторної батареї». У режимі «Робота» світиться правий (зелений) кристал світлодіода. У режимі «Робота-заряд акумуляторної батареї» світяться обидва кристала світлодіода (зелений і червоний). У режимі «Заряд акумуляторної батареї» світиться лівий (червоний) кристал світлодіода.

Резистор R3 призначений для обмеження напруги на червоний кристал світлодіода VD3, він розраховується за формулою 3.4:

$$R = \frac{U_{жив} - U_{паддіода}}{I_{діода}}, \quad (3.4)$$

де $U_{жив}$ – напруга на виході стабілізатора DA1 10В;

$U_{пад. діода}$ – напруга живлення червоного кристала 2В;

$I_{спож. діода}$ – струм, який споживає червоний кристал світлодіода 18 мА.

Підставивши значення отримуємо:

$$R3 = \frac{10 - 2}{0,018} = 444 \approx 470 \text{ Ом}, \quad (3.5)$$

Оскільки в ряді E24 не вказане значення 444 Ом приймаємо 470 Ом.

Конденсатори C1 і C5 призначені для зменшення пилкоподібних стрибків по лінії живлення, які можуть виникати при використанні неякісних блоків живлення.

Конденсатори C6-C10 призначені для гасіння імпульсних перешкод, які виникають в лініях живлення, компенсує індуктивність танталових конденсаторів C1 і C5 і провідників друкованої плати. Конструктивно вони повинні розташовуватися в безпосередній близькості до інших елементів (наприклад між лініями живлення окремих мікросхем).

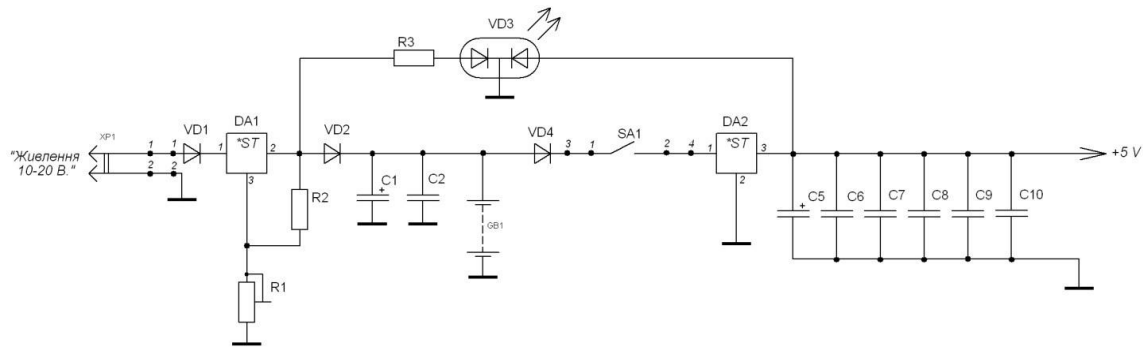


Рисунок 3.4 – Модуль живлення пульта керування

Для зв'язку пульта керування зі світлодіодним модулем відображення графічної інформації застосований модуль радіоканалу DD1. Для зручності роботи і зменшення витрат на розробку програмного забезпечення застосовується модуль з вбудованим протоколом передачі. Радіомодуль з'єднується безпосередньо з мікроконтролером за універсальним інтерфейсом прийому - передачі даних UART і забезпечує зв'язок зі швидкістю 9600 біт/сек.

3.1.2 Принципова схема блоку контролера світлодіодного модуля відображення графічної інформації

Принципова схема блоку контролера світлодіодного модуля відображення графічної інформації практично не відрізняється від схеми пульта управління. Центральним елементом схеми є мікроконтролер DD5 (рис. 3.5). Для роботи мікроконтролера встановлений резистор R6 (рис. 3.5) номіналом 10 кОм, він підтягує вивід скидання мікроконтролера до плюса живлення. Кварцовий резонатор ZQ2 забезпечує роботу тактового генератора мікроконтролера з частотою 12 МГц. Конденсатори C12 і C13 номіналом 22пФ, що забезпечують стабільний запуск тактового генератора (рис. 3.5). Мікросхема пам'яті DD4 об'ємом 512 кбіт з включенням адреси A0. Мікросхема годинника реального часу DD3 забезпечує незалежний від мікроконтролера хід часу.

Кварцовий резонатор ZQ1 з робочою частотою 32,768 кГц забезпечує відлік часу мікросхемою DD3. Елемент живлення GB1 призначений для незалежності ходу годинника від енергопостачання пристрою.

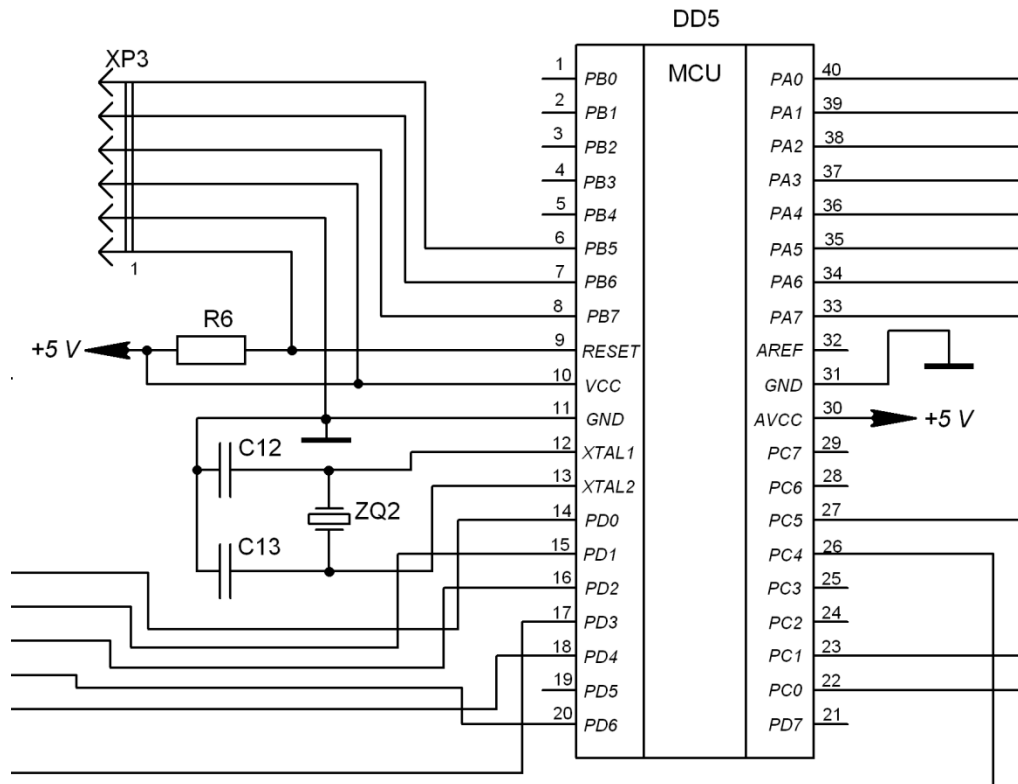


Рисунок 3.5 – Центральний вузол схеми блока контролера – мікроконтролер ATmega32-16 AU

Мікросхема реального ходу годинника DD3 і мікросхема EEPROM DD4 пов'язані з мікро контролером по інтерфейсу I²C. Резистори R7 і R8 забезпечують підтяжку ліній даних і тактових імпульсів, що забезпечує надійну роботу інтерфейсу від помилкових імпульсів.

Для зв'язку світлодіодного модуля відображення графічної інформації з персональним комп'ютером (дана функція є опціональною) на мікросхемі DD1 реалізований інтерфейс RS232 (рис. 3.6). Конденсатори C1, C3, C4, C8 є стандартизованими. Резистори R2 - R5 призначені для обмеження струму по лініях UART інтерфейсу, їх номінал становить 120 Ом. Для індикації поточної

температури передбачений роз'єм XP2 для підключення датчика зі зв'язком по інтерфейсу 1 – Wire, резистор R1 здійснює підтяжку лінії даних від випадкових подій.

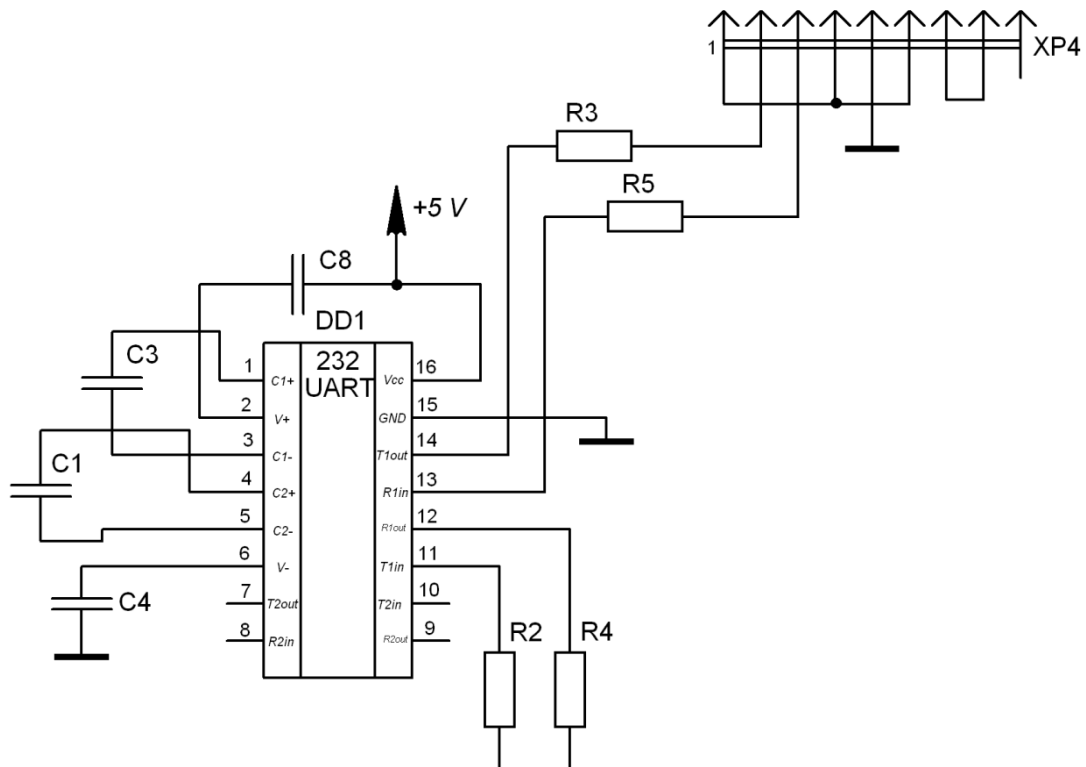


Рисунок 3.6 – Реалізація зв'язку блока контролера з персональним комп'ютером за допомогою інтерфейсу RS232

Оскільки блок індикації з модулем контролю з'єднується за допомогою шлейфа, для виключення втрат в ньому встановлені мікросхеми DD5 і DD6 які є двонаправленими трансиверами, схема їх включення наведена на рис. 3.7.

Конденсатор C2 призначений для фільтрації ліній живлення від імпульсних перешкод. Конденсатори C5-C7 і C9-C11 призначені для компенсації індуктивності від конденсатора C2, провідників друкованої плати, конструктивно розташовані на друкованій платі між виводами живлення мікросхем.

Модуль радіоканалу DD2 призначений для організації каналу зв'язку з пультом управління зі швидкістю зв'язку 9600 біт/сек. Підключається до мікроконтролера через інтерфейс UART.

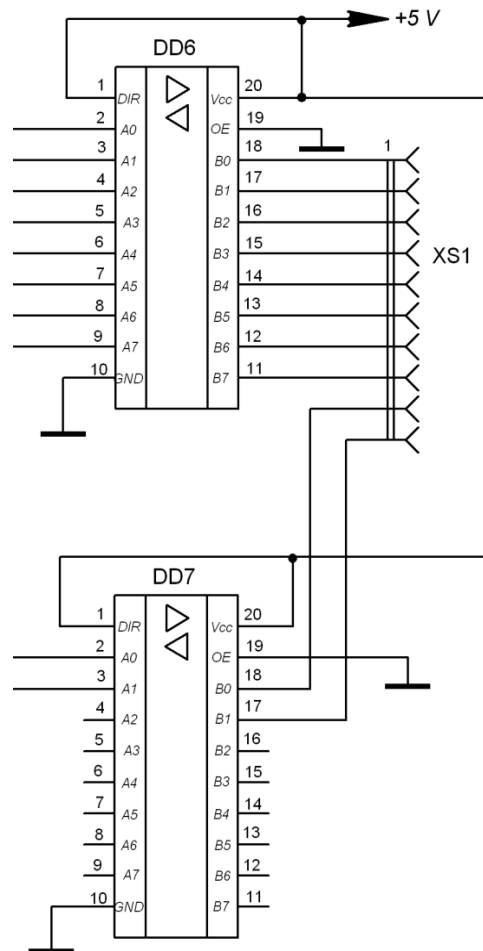


Рисунок 3.7 – Схема включення буферних трансиверів DD6 і DD7

3.1.3 Принципова схема блоку індикації світлодіодного модуля графічної інформації

Блок індикації складається з світлодіодів, аноди яких об'єднані між собою. Їх включення керується за допомогою світлодіодних драйверів, кожен вихід яких підключений до катода певного світлодіода.

Світлодіодні драйвери представляють 2×8 бітний зсувний регістр з польовим транзистором на кожному виході, за допомогою яких встановлюється вихідний струм на канал (рис. 3.8). Значення опору резистора визначається за графіком представленим на рисунку 3.7.

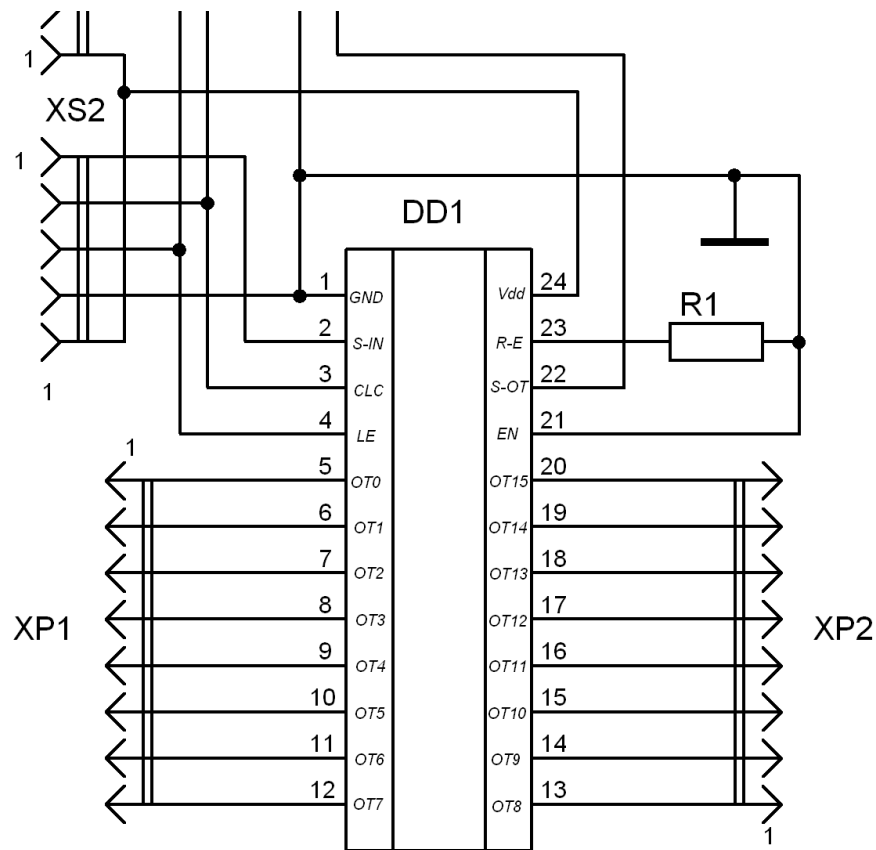


Рисунок 3.8 – Схема включення світлодіодного драйвера

Оскільки управління буде здійснюватися кожним світлодіодом окремо то обираємо струм виходу (з урахуванням заниження струму для продовження терміну служби світлодіодів, і зменшення деградації кристала світлодіода) рівний 16 мА. Виходячи з графіка представлено на рис. 3.9 вибираємо опір резисторів підтяжки світлодіодних драйверів $R1 = 1 \text{ кОм}$ (рис. 3.8).

Діод VD1 призначений для захисту від зворотної напруги, і зміщення нульової точки щодо землі, тому він повинен мати низьку напругу відкриття переходу, виходячи з цього застосований діод з переходом Шотки.

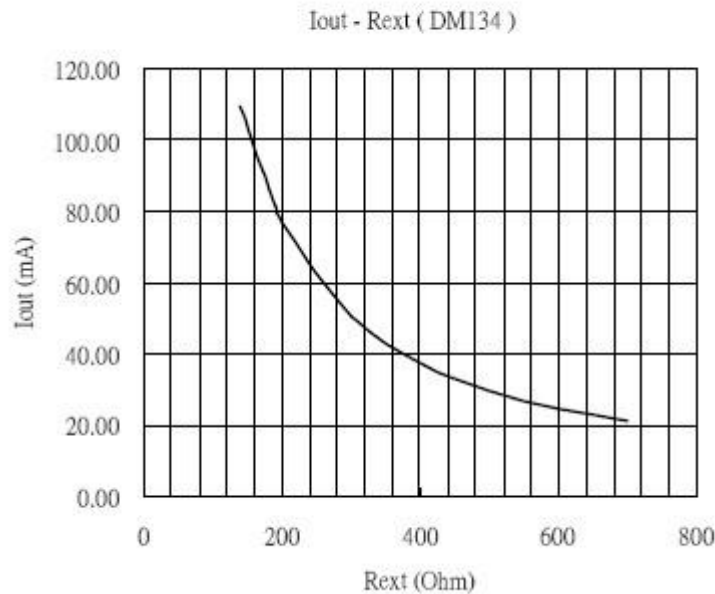
DM134

Рисунок 3.9 – Графік визначення опору резистора R1

3.2 Вибір і обґрунтування елементної бази

Одним з основних етапів розробки пристрою є вибір елементної бази.

Від обраного типу елементів залежать розміри пристрою, надійність приладу і його вартість. Можна вважати що, якщо елементна база обрано вірно то прилад, який розробляється буде мати високу надійність і невисоку вартість. Потрібно враховувати той фактор, що при застосуванні мікросхем надійність приладу значно зменшується тому, що їх коефіцієнти залежать від кількості елементів в них. Тому для підвищення надійності та зменшення ціни необхідно дуже ретельно обирати аналогові елементи, тобто використовувати елементи які мають високу надійність і малу ціну по відношенню до інших елементів (комунікаційні елементи, роз'єми). На сьогоднішній день з метою зменшення геометричних розмірів і ваги виробів застосовуються радіоелементи поверхневого монтажу. У пристрої, який розробляється мною, застосовані

елементи поверхневого монтажу типорозміру 1206 і мікросхеми в корпусах TQFP44, SO8.

3.2.1 Конденсатори

Конденсатори поділяються на постійні, регульовальні і змінні. У моєму приладі використовуються тільки постійні конденсатори оскільки у мене відсутні вузли в яких застосовуються змінні і підлаштувальні конденсатори (ланцюги з ручним регулюванням часових і частотних параметрів). Вибирають конденсатори по багатьом параметрам, основні з них це: місткість, робоча напруга і група ТКЕ.

Для розроблювального мною приладу я вибрав постійні керамічні конденсатори 1206-CE-5%, які мають порівняно з іншими меншу вагу і габарити, працюють при напрузі до 50 В і конденсатори танталові 1206-TAN-15% виробництва: Bourns, які значно менші за габаритами від їх аналогів і мають робочу напругу 16 В.

3.2.2 Резистори

Резистори також поділяються на постійні, регульовальні і змінні. Вибирають їх за значенням опору, потужності і відсотку відхилення.

У приладі, який розробляється мною, використовуються постійні метало-композитні резистори типу 1206-CRE-5% і підлаштувальні типу СП5-1, номінальне відхилення від заданого значення опору 5%. Даний тип резистору має малі габарити, максимальна робоча напруга 100 В, і ряд проміжних значень опору E24.

3.2.3 Комутаційні елементи

Комутаційні елементи використовують для перемикання між собою різних електричних сигналів в ланцюгах. У сучасній радіоелектроніці застосовують перемикачі, кнопки, клавіатури різних типів і форм, з різною кількістю режимів і функцій.

Мною застосований мікро перемикач типу RK2-14 який має максимальний комутаційний струм 1А і максимальну комутаційну напругу 250 В. В пульті управління використана мембранна клавіатура типу Ak 304-N-BBW має 12 кнопок включених матрицею 4x3.

3.2.4 Діоди

Діоди виготовляють з двох матеріалів: кремнієві і германієві.

Найчастіше застосовують кремнієві діоди оскільки вони мають меншу в порівнянні з германієвими порогову напругу відкриття діода. У пристрої, що розробляється мною, основним критерієм є напруга падіння на діодах. Провівши дослідження ринку напівпровідникової продукції були обрані діоди з бар'єром Шотки виробництва підприємства STEALTH™ 1N5819 з напругою падіння 0,2-0,4 В.

Для інформування оператора про режими роботи в пульті управління встановлений двоколірний світлодіод GNL-5013 GYW.

Світлодіоди масиву відображення. Основний елемент відображення інформації проєктованого пристрою є – світлодіод.

Так само як в комп'ютерах, частота процесора і обсяги оперативної пам'яті – основні характеристики ПК від яких залежить вартість, в рухомих рядках яскравість світлодіодів і кут їх випромінювання є визначальними параметрами.

Сила світла світлодіода вимірюється в канделах на метр квадратний, а кут випромінювання в градусах. Поняття кут світіння не існує, існує поняття – кут

половинної яскравості світлодіода – це кут при відхиленні на який від центральної осі світлодіода, яскравість зменшується в 2 рази.

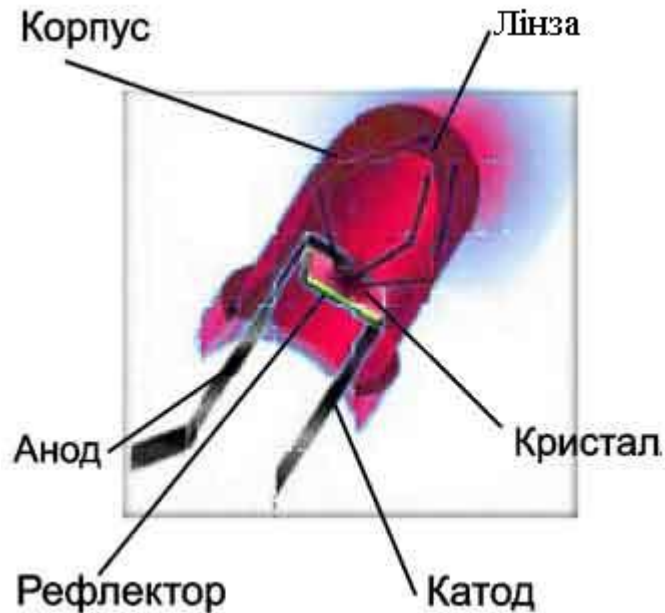


Рисунок 3.10 – Конструкція світлодіода

Як правило, світлодіоди виготовляються серіями. Вся серія має один і той же елемент, який випромінює світло – напівпровідниковий кристал. А ось збираючі лінзи застосовують різні, таким чином, отримують світлодіоди з половинним кутом яскравості 15, 30, 60, 120 градусів. Застосовуючи різні лінзи, всього лише збільшують або зменшують щільність світлового потоку.

Наведемо приклади параметрів світлодіодів фірми G-pog. При однаковій вартості маємо:

- червоні 5мм 3500mcd 18° (GNL-5013UEC-TL);
- червоні 5мм 1200 mcd 45° (GNL-5013UED-TL);
- червоні 5мм 650 mcd 60° (GNL-5053UEC-TL);
- червоні 5мм 250 mcd 140° (GNL-5073UEC-TL).

На сьогоднішній день для звичайних 5 мм світлодіодів максимум це 1-1,5 кандела при 60 градусах. Існують, звичайно, світлодіоди з трьома кристалами в

одному корпусі тощо, але їх вартість занадто велика. Так само існують фірми виробники, які самі завищують параметри своїх світлодіодів.

Проаналізувавши наведене вище були обрані світлодіоди GNL-5013UED-TL.

3.2.5 Транзистори

У пристрої, що розробляється мною, застосований малопотужний польовий транзистор IRLM2502.

3.2.6 Мікросхеми

У якості буферного каскаду використані мікросхеми 74HC245, які представляють собою трансивер з трьома станами, здатний працювати в обох напрямках в залежності від вибору підключення. Для забезпечення стабільної напруги живлення використані стабілізатори напруги DA1 - LM317 і DA2 - L7805.

Як вказано вище, в пульті управління і світлодіодному модулі для відображення графічної інформації з метою спрощення програми і для точного відліку часу застосовані спеціалізовані мікросхеми виробництва Maxim DS1307.

Основні характеристики DS1307:

- годинник реального часу;
- відлік секунд, хвилин, годин, дати, місяця, днів тижня і року, з високосним роком;
- звіт до 2100 року;
- 56-Byte оперативної пам'яті для зберігання даних;
- I²C Послідовний Інтерфейс;
- споживання менш ніж 500нА від резервної батареї;

Структурна схема мікросхеми DS1307 представлена на рисунку 3.11.

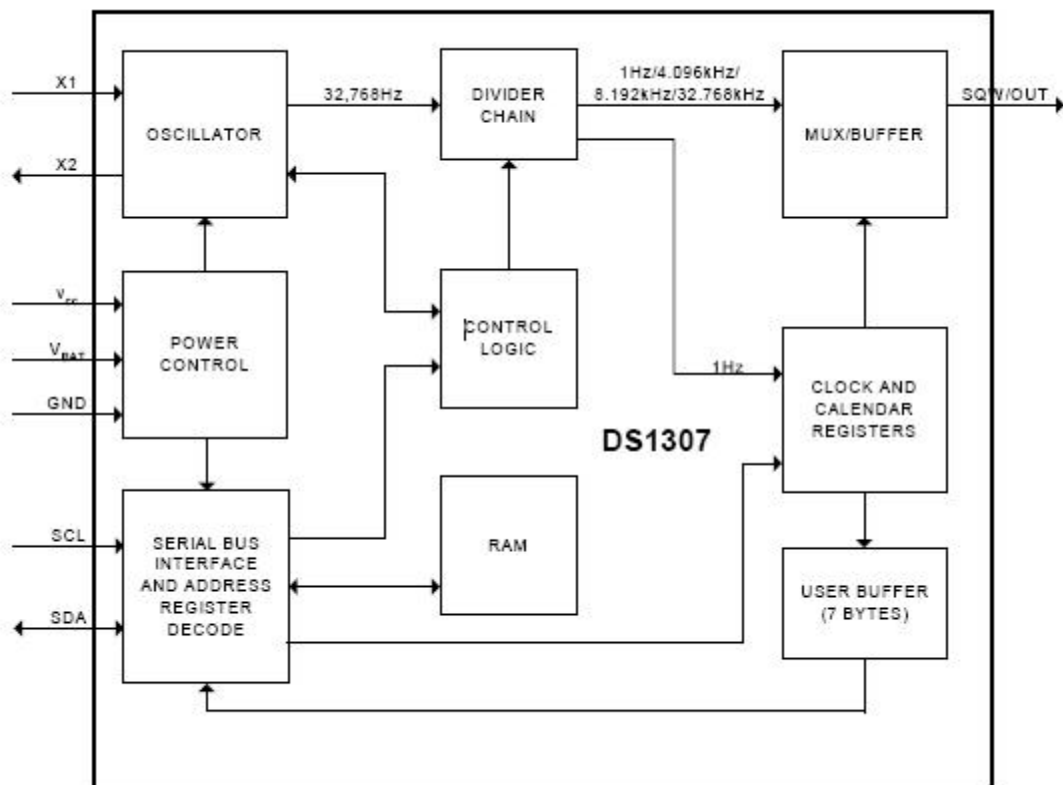


Рисунок 3.11 – Структурна схема мікросхеми DS1307

Зовнішня EEPROM пам'ять використана типу 24C512.

Основні технічні характеристики:

- режими низьковольтного живлення і стандартного:
 - а) 2.7 (VCC = 2.7V - 5.5V);
 - б) 1.8 (VCC = 1.8V - 3.6V);
- внутрішня організація пам'яті 65,536 x 8;
- послідовний інтерфейс I²C;
- тригери Шмітта, фільтровані входи з шумоподавлювачем;
- протокол двонаправлених даних;
- 1 МГц (5V), 400 кГц (2.7V) і 100 кГц (1.8V) сумісність;
- 128-байтний метод запису сторінки;
- висока надійність;
- гарантовано 100,000 циклів запису;

– зберігання даних: 40 років.

Структурна схема 24C512 приведена на рисунку 3.12.

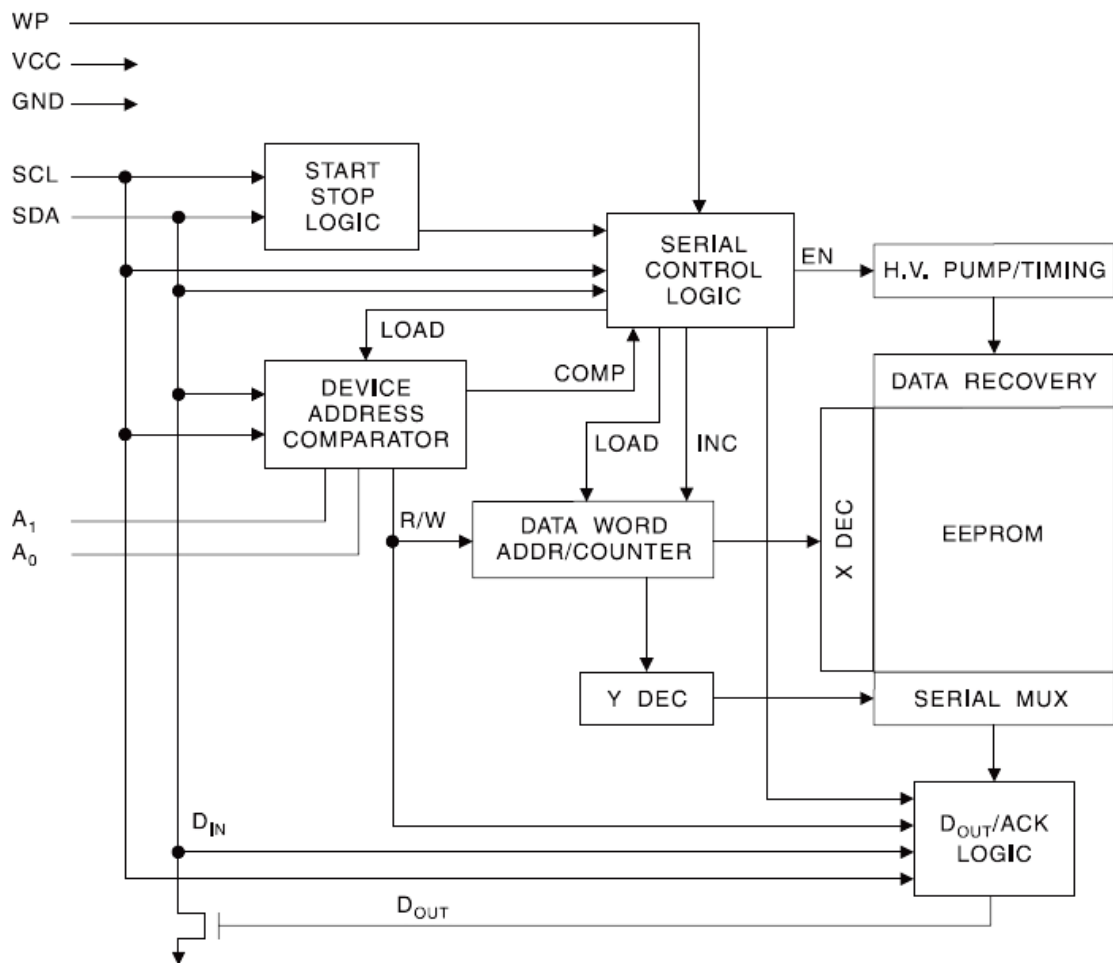


Рисунок 3.12 – Структурна схема 24C512

Для управління світлодіодами застосовані спеціалізовані світлодіодні драйвери DM134.

Структурна схема драйвера DM134 представлена на рисунку 3.13.

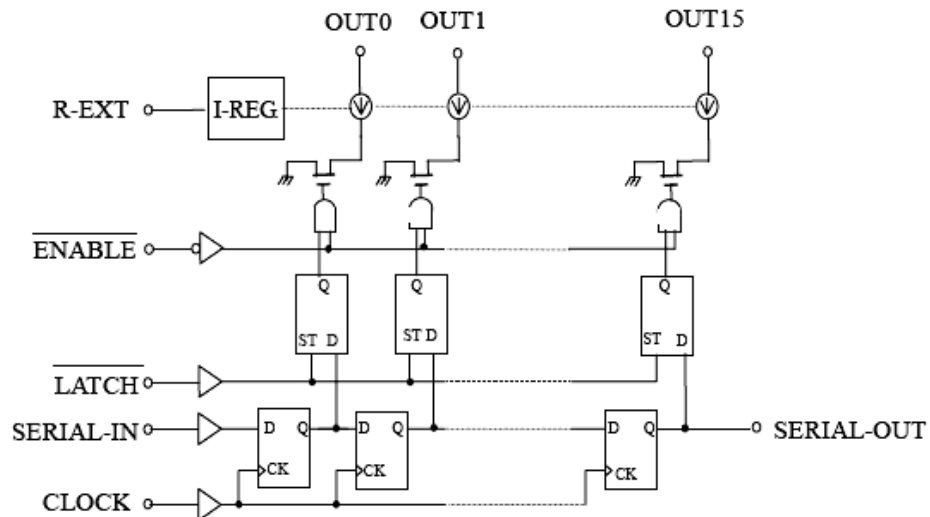


Рисунок 3.13 – Структурна схема драйвера DM134

Вони являють собою зсувний регістр з виходом організованим на польових транзисторах з ізольованим затвором. На кожному виході забезпечується чітко заданий струм в межах від 10мА до 90мА. Струм на виході задається резистором підключеним між землею і виходом R-EXT. Номінал даного резистора визначається за графіком представленим на рисунку 3.6.

3.2.7 Модуль радіоканалу

Для реалізації радіоканалу між пультом управління і світлодіодним модулем відображення графічної інформації застосовані спеціалізовані промислові модулі радіозв'язку Xbee підприємства Maxstream типу XB24-AWI-001.

Основні технічні параметри XB24-AWI-001:

- радіус дії в приміщенні 50 м;
- радіус дії у вільному просторі 100 м;
- максимальна вихідна потужність 1мВт;

- швидкість передачі даних по радіоканалу 250000 біт / с;
- швидкість передачі даних по інтерфейсу 1200 .. 115200 біт / с;
- чутливість 92 дБм;
- напруга живлення 2,8 .. 3,4 В;
- струм споживання в режимі передачі 45 мА;
- струм споживання в режимі прийому, 50 мА;
- струм споживання в режимі енергозбереження 10 мкА;
- робоча частота 2,4 ГГц;
- кількість каналів 16;
- кількість адрес в мережі 65000.

Вибір зроблено на використання даних радіомодулів оскільки вони мають невеликі габаритні розміри, вбудовані протоколи організації радіозв'язку, зручність в сполученні даного радіомодуля з безпосереднім мікроконтролером.

Структурна схема радіомодуля представлена на рисунку 3.14.

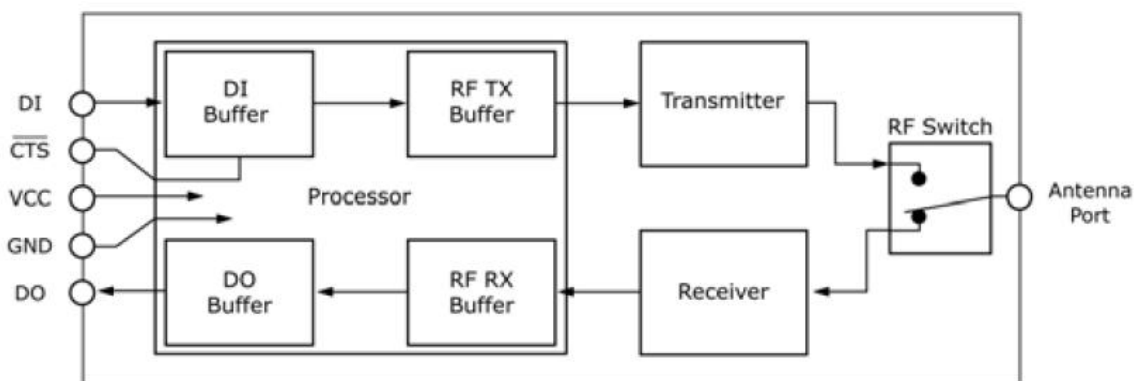


Рисунок 3.14 – Структурна схема радіомодуля XB24-AWI-001

3.2.8 Роз'єми

Для підключення пульта управління до зовнішнього модульного імпульсного джерела живлення використано стандартне гніздо живлення Power_con.

Для підключення стандартної комп'ютерної клавіатури застосовано стандартизоване гніздо PS / 2M-Conn.

Для організації з'єднання плат драйверів світлодіодів і плат модулів було прийнято використовувати штирові штекери PLS-40 і відповідні гнізда BPS-40.

3.2.9 Індикатор

Призначений для виведення текстової інформації на пульті дистанційного керування. Оскільки екран повинен відображати велику кількість інформації було вирішено використовувати індикатор ACM2004D.

3.2.10 Кварцові резонатори

У розробленому пристрої були застосовані кварцові резонатори виробництва GEYER наступного найменування: KX-49 12МГц і часові кварцові резонатори KX-38 32.768 кГц.

4 ПОБУДОВА АЛГОРИТМІВ І ПРОГРАМ

На першому етапі створення програми необхідно визначити послідовність дій, які необхідно виконати, щоб вирішити поставлене завдання, тобто розробити алгоритм.

Алгоритм – це точне розпорядження, що визначає процес переходу від вихідних даних до результату.

Алгоритм рішення задачі може бути представлений у вигляді словесного опису або графічно – у вигляді блок-схеми. Подання алгоритму у вигляді блок-схеми дозволяє усвідомити послідовність дій, які повинні бути виконані для вирішення завдання, переконатися в правильності розуміння поставленого завдання.

4.1 Побудова алгяритму програми пульта управління

Першою операцією роботи програми має бути введення і перевірка правильності введеного коду, це необхідно для захисту пристрою від чорної реклами шляхом установки некоректного тексту конкурентами. Далі має запускатися внутрішнє програмне меню, яке управляє всіма системними функціями, навігація в ньому повинна здійснюватися за допомогою дванадцяти кнопкової клавіатури.

Для встановлення системного часу в світлодіодному модулі відображення графічної інформації та на дисплеї пульта управління, для зручності користування пультом управління, необхідно ввести процедуру зміни часу по інтерфейсу I²C, яка повинна включати в себе процедуру опитування поточного значення часу і дати та виведення їх значень на екран в підпрограмі установки системного часу і дати. Далі необхідний блок установки, який всередині програмної оболонки буде перетворювати введені значення дати і часу в зрозумілий для годин реального часу формат, після чого повинна здійснюватися відправка значення часу і дати в годинник реального часу.

Для введення тексту за допомогою комп'ютерної клавіатури необхідно провести ініціалізацію підключення комп'ютерної клавіатури. При введенні з комп'ютерної клавіатури необхідна наявність зовнішньої, розділеної на сектори пам'яті. Вибір сектора пам'яті для зберігання даних необхідно вибирати програмно, за допомогою внутрішньої клавіатури.

Оскільки дані про натискання тієї чи іншої клавіші комп'ютерної клавіатури приходять на контролер в стандарті ST3, то необхідний блок перекодування даних з формату ST3 в формат даних зрозумілий мікроконтроллеру – ASCII. Після перекодування даних здійснюється вивід введеного символу на дисплей і запис байта в зовнішню незалежну пам'ять з початкової адреси обраного сектора, з додаванням одиниці до адреси записаного байта. По закінченню запису байта, програма повинна повернутися в початковий стан до приходу нового коду з клавіатури.

Для запису даних введених з комп'ютерної клавіатури в світлодіодний модуль відображення графічної інформації повинна бути реалізована процедура запису. На початку необхідно зробити вибір сектора пам'яті в який зроблений запис даних. Далі необхідно рахувати перший байт, що зберігається в зовнішній пам'яті і додати одиницю до адреси для зчитування подальшого байта, далі по умовному протоколу зв'язку пульта управління зі світлодіодним модулем передати лічений байт, після закінчення передачі програма повертається до зчитування наступного байта.

З метою забезпечення більшої зручності користувача необхідна організація зміни коду доступу. Процедура зміни коду доступу виконана за стандартною схемою. Перший етап видалення старого значення коду доступу. Другий етап включення і виключення коду доступу. І установка нового значення коду.

З метою перегляду введеного раніше тексту необхідна процедура читання даних з зовнішньої незалежної пам'яті. Для цього необхідно в першу чергу вибрати одну з розмічених раніше областей, далі рахувати певну кількість байтів для їх одночасного виведення на дисплей, потім реалізувати затримку,

достатню для прочитання масиву одного екрану, вивести наступний масив байтів даних, які прочитуються з зовнішньої пам'яті.

По закінченню всіх програмних процедур, програма повинна повертатися у вихідну точку.

4.2 Побудова алгоритму програми блоку контролера світлодіодного модуля відображення графічної інформації

Для здійснення безперервного виводу тексту на світлодіодний масив алгоритм повинен повторюватись циклічно.

З метою отримання даних, що передаються з пульта управління, які можуть надходити в будь-який момент часу, необхідне введення наскрізного переривання на прихід даних по інтерфейсу UART, які повинні бути занесені в зовнішню незалежну пам'ять.

Процедура виведення тексту починається зі зчитування байта даних з початкової адреси зовнішньої пам'яті.

Після завершення операції по зчитуванню байта необхідна процедура розпізнавання системних і текстових значень. Вони виконані у вигляді логічних подій. При зчитуванні байта виведення температури навколишнього середовища повинна відбуватися ініціалізація інтерфейсу даних 1 – Wire, оскільки було прийнято рішення використовувати датчики саме з цим інтерфейсом для зменшення кількості ліній зв'язку та втрат в них, оскільки довжина сполучних ліній з датчиком може досягати декількох метрів. Оскільки дані, що передаються датчиком не можуть бути використані безпосередньо, необхідний блок перекодування.

При зчитуванні байта виведення поточної дати, програма повинна звертатися по інтерфейсу I²C до годинника реального часу і робити зчитування значення поточного часу в форматі: годинник – хвилини, опускаючи при цьому значення секунд і дати. Лічений результат необхідно перетворити в зрозуміле для мікроконтролера значення.

При зчитуванні байта виведення дати необхідно рахувати значення годин реального ходу, опускаючи значення годин. Після чого необхідно перетворити отримане значення і вивести його на світлодіодне поле.

Якщо лічений байт не відноситься до виводу поточної дати, поточного часу або температури навколишнього середовища то відбувається декодування даного байта в текстовий. Після чого повинна відбуватися процедура запису декодованого символу в полі світлодіодних драйверів, відповідно до технічного опису виробника драйверів.

Алгоритм складений за наведеним вище описом наведено в Додатку А.

4.3 Побудова програмного коду для мікроконтролера пульта управління

Для написання програмного коду був використаний популярний компілятор для мікроконтролерів AVR FastAVR. Який на відміну від більшості подібних компіляторів генерує чистий асемблерний код придатний для компіляції будь-яким з відомих на сьогоднішній день компілятором. Програма FastAVR використовує мову Бейсік і має велику кількість вбудованих бібліотек для швидкого написання програмного коду.

Нижче наведені частини програми мікроконтролера пульта управління. Повна програма наведена в Додатку Б.

Оскільки всі випускаються серійно, то рідкокристалічні індикатори з вбудованими драйверами шрифтів не містять у своїй пам'яті деякі спеціальні символи і деякі символи української мови, ці символи були створені програмно за допомогою спеціального, вбудованого в компілятор, графічного редактора, цей код наведено на лістингу 4.1.

```

DefLcdChar 0, &h02, &h04, &h04, &h08, &h04, &h04, &h02, &h00 '{
DefLcdChar 1, &h08, &h04, &h04, &h02, &h04, &h04, &h08, &h00 '}'
DefLcdChar 2, &h0E, &h11, &h10, &h1C, &h10, &h11, &h0E, &h00 '€
DefLcdChar 3, &h00, &h00, &h0E, &h11, &h1C, &h11, &h0E, &h00 '€
DefLcdChar 4, &h0A, &h00, &h0E, &h04, &h04, &h04, &h0E, &h00 'İ
DefLcdChar 5, &h00, &h0A, &h00, &h0C, &h04, &h04, &h0E, &h00 'İ
DefLcdChar 6, &h0A, &h15, &h0A, &h15, &h0A, &h15, &h0A, &h00 'date
DefLcdChar 7, &h0E, &h15, &h15, &h17, &h11, &h11, &h0E, &h00 'time

```

Лістинг 4.1 – Програмна реалізація спеціальних символів для ЖКІ

Як описано в попередніх пунктах, в пульті управління було організовано внутрішнє меню вибору певних функцій. Програма даного меню організована за допомогою команд вибору і скидання кнопок вбудованої клавіатури. На лістингу 4.2 приведена частина програми організації даного меню.

```

'===== IInit_main =====
Sub IInit_main()
    Cls
    WaitMs 50
    Cursor Off
    Locate 1,7
    Lcd "RunLin "
    Locate 2,1
    Lcd "1-Word text 2-TX txt"
    Locate 3,1
    Lcd "3-Seting 4-Read"
    For wil=0 To 255
        Select_key()
        Init_clok_lcd()
        If wil=255 Then
            wil=0
        End If
    Next
End Sub
'===== Select keybord =====
Sub Select_key()

```

```

k=0
While k=0
    k=0
    k=Key()
    Init_clok_lcd()                                ' if key is pressed
Wend
    Select Case k
    Case 58      'key0
        Led()
    Case 33      'key1
        Key_1()
    Case 34      'key2
        Key_3()
    Case 35      'key3
        Key_4()
    Case 41      'key4
        Key_5()
    Case 59      'key#
        Key_reset()
    End Select
End Sub
'=====

```

Лістинг 4.2 – Організація внутрішнього меню

Зв'язок з годинником реального часу і зовнішньою пам'яттю здійснюється по інтерфейсу I²C. Для цього використані вбудовані в компілятор бібліотеки даного інтерфейсу. Організовано читання і запис даних в годинник реального часу; читання, запис і стирання даних у зовнішній незалежній пам'яті а також відсилання даних по інтерфейсу UART. Частина даних програм наведені в лістингу 4.3.

‘===== Read time =====

```
Sub GetTime()
    I2Cstart DS1307
    I2Cwrite 0
    second=I2Cread, Ack
    min=I2Cread, Ack
    hour=I2Cread, Ack
    dof=I2Cread, Ack
    day=I2Cread, Ack
    mon=I2Cread, Ack
    year=I2Cread, Nack
    I2Cstop
End Sub
```

‘=====

```
Sub Read_24lc512_tx()
Local tmp As Byte, ef As Byte
    adDD=ReadEE(adEE)
    adDD1=ReadEE(adEE1)
    iadDD=(adDD1*127)+adDD
    iadDD1=0
    WaitMs 15
    I2Cstart &hA0      ‘ 24C64 address
    I2Cwrite Msb(aDDR)  ‘ write adr MSB
    I2Cwrite aDDR
For tmp=0 To 15      ‘ write adr LSB
    For ef=0 To 249
        iadDD1=iadDD1+1
        tx=I2Cread, Nack
        PrintBin tx
        WaitMs 10
        ‘Swap(tx)
        ‘PrintBin tx
        ‘WaitMs 60
    If iadDD1=iadDD Then
        Lcd Chr(&hFF)
```

```

        tmp=I2Cread, Nack    ' dummy read - we need Nack!
        I2Cstop
    Exit Sub
End If
Next
Lcd Chr(&hFF)
Next
    tmp=I2Cread, Nack    ' dummy read - we need Nack!
    I2Cstop
    'Incr aDDR
End Sub
'=====

```

Лістинг 4.3 – Читання даних годин реального часу, зовнішньої пам'яті і відсилання даних по інтерфейсу UART

Реакція на натиснення кнопок клавіатури виконана як очікування натискання однієї з кнопок і подальший вибір відповідної дії. Приклад реалізації вибору дії на натискання кнопки «Один» з основного меню наведено на лістингу 4.4.

```

'===== Procedyre key1 =====
Sub Key_1()
Local tr As Byte, ty As Byte
k1:
    aDDR=&h0000
    arfDDR=0
    Cls
    WaitMs 50
    Cursor Off
    Locate 1,5
    Lcd "Run word text"
    Locate 2,1
    Lcd "1-File_1 2-File_2"
    Locate 3,1
    Lcd "3-File_3"

    For wil=0 To 255
        kp1=0
        While kp1=0

```

```

    kp1=0
    kp1=Key()
    'Init_clok_lcd()
Wend
    Select Case kp1
        Case 33
            Cls
            WaitMs 50
            Cursor On
            Locate 1,1
            Lcd "Text:"
            aDDR=&h0002      'write adres 24lc512
            Erase_24()
            aDDR=&h0002      'write adres 24lc512
            adEE1=&h0051
            adEE=&h0052
            lang=0
            Select_lang()
            ps_2()
        Case 34
            Cls
            WaitMs 50
            Cursor On
            Locate 1,1
            Lcd "Text:"
            aDDR=&h4000      'write adres 24lc512
            Erase_24()
            aDDR=&h4000
            adEE1=&h0053
            adEE=&h0054
            lang=0
            Select_lang()
            ps_2()
        Case 35
            Cls
            WaitMs 50
            Cursor On
            Locate 1,1
            Lcd "Text:"
            aDDR=&h8000      'write adres 24lc512
            Erase_24()
            aDDR=&h8000
            adEE1=&h0055
            adEE=&h0056
    
```

```

        lang=0
        Select_lang()
        ps_2()
    Case 57
        GoTo K1
    Case 59
        Key_reset()
End Select
If wil=255 Then
    wil=0
End If
Next

'=====

```

Лістинг 4.4 – Реалізація програмного коду реакції на
натискання кнопки «Один» з основного меню

Як було сказано вище для захисту пульта від несанкціонованого доступу впроваджена програмна перевірка коду доступу, частина цієї програми наведено на лістингу 4.5.

```

'===== Test Lok =====
Sub lok()
    Cls
    WaitMs 50
    Cursor Off
    Locate 1,1
    Lcd "Word kod"

    lok1 = ReadEE(&h0030)
    lok2 = ReadEE(&h0031)
    lok3 = ReadEE(&h0032)
    lok4 = ReadEE(&h0033)
    lk1=0
    lk2=0
    lk3=0
    lk4=0
    lk5=0
    kl1=0

```

```
kl2=0
kl3=0
kl4=0
```

```
While lk1=0
    lk1=Key()
    If lk1=lok1 Then
        kl1=1
        Locate 2,1
        Lcd"*"
    Else
        If lk1<>0 And lk1<>lok1 Then
            Locate 2,1
            Lcd"*"
        End If
    End If
End If
Wend
```

```
While lk2=0
    lk2=Key()
    If lk2=lok2 Then
        kl2=1
        Locate 2,2
        Lcd"*"
    Else
        If lk2<>0 And lk2<>lok2 Then
            Locate 2,2
            Lcd"*"
        End If
    End If
End If
Wend
```

```
While lk3=0
    lk3=Key()
    If lk3=lok3 Then
        kl3=1
        Locate 2,3
        Lcd"*"
    Else
        If lk3<>0 And lk3<>lok3 Then
            Locate 2,3
            Lcd"*"
        End If
    End If
End If
```

```

Wend

While lk4=0
    lk4=Key()
    If lk4=lok4 Then
        kl4=1
        Locate 2,4
        Lcd "*"
    Else
        If lk4<>0 And lk4<>lok4 Then
            Locate 2,4
            Lcd "*"
        End If
    End If
End If

Wend

While lk5=0
    lk5=Key()
    'NoKey()
    Locate 4,1
    Lcd "*-OK"
    Locate 4,16
    Lcd "#-res"
    If lk5=57 And kl1=1 And kl2=1 And kl3=1 And kl4=1
Then
        IIIinit_main()
    Else
        If lk5=59 Then
            Lok()
        Else
            If lk5<>0 And lk5<>57 And lk5<>59 Then
                lk5=0
            End If
        End If
    End If
End If

Wend

End Sub
'=====

```

Лістинг 4.5 – Частина програми перевірки коду доступу

4.4 Побудова програмного коду для мікроконтролера блоку контролера світлодіодного модуля відображення інформації

Програма для мікроконтролера блоку контролера написана на тому ж компіляторі що і попередня програма.

Повний текст програми наведено в Додатку В.

Для відображення символів, байти, що зчитуються з пам'яті повинні бути перетворені на відповідні дані для виведення їх на порт мікроконтролера з подальшим записом в світлодіодні драйвера. Уривок даного перетворення приведений в лістингу 4.6.

```
Sub OUTB(d As Byte)
  Local ii As Byte
  For ii=0 To 13
    BUFZ(ii)=Table1(d*14+ii)
  Next
End Sub
```

Лістинг 4.6 – Перетворення даних для виведення їх в порт мікроконтролера

Регламентована виробником процедура запису драйверів організована програмно, і здійснює запис даних з порту мікроконтролера в усі драйвера масиву. Нижче наведено код запису даних в драйвери, лістинг 4.7.

```
Sub WRITE()
  Local bb As Byte
  count=BUFZ(0)
  For j=1 To count
    bb=BUFZ(j)
    PORTA=bb
    Nop
    Nop
    Set Clock
    WaitUs 30
    Reset Clock
```

```

Next
End Sub

```

Лістинг 4.7 – Запис даних в світлодіодні драйвери

Щоб по закінченню перегляду тексту, перед новою процедурою виведення, світлодіодне поле повністю очищалося, введена підпрограма обнулення байтів даних масиву, вона приведена на лістингу 4.8.

```

Sub Nullleft()
  For k=0 To 31
    For i=0 To 30
      m=31-i:n=m:Decr n
      BUFF(m)=BUFF(n)
    Next
    PORTA=0
    For i=0 To 31
      Set Clock
      Nop
      Reset Clock
    Next
    Set LE
    Nop
    Reset LE
    WaitMs 2
    BUFF(0)=0
    For i=0 To 31
      m=31-i
      PORTA=BUFF(m)
      Set Clock
      Nop
      Reset Clock
    Next
    Set LE
    Nop
    Reset LE
    WaitMs 17
  Next
End Sub

```

Лістинг 4.8 – Підпрограма зсуву нуля до кінця рядка

5 РОЗРАХУНОК НАДІЙНОСТІ

5.1 Теоретична частина

5.1.1 Надійність – це властивість системи виконувати задані функції заданого часу, при заданих умовах експлуатації. Надійність характеризується якісними та кількісними показниками надійності.

Мета розрахунку:

- визначити кількісні показники надійності;
- виявити слабкі місця в системі;
- визначити шляхи підвищення надійності.

Вихідний матеріал для розрахунку:

- принципові схеми з переліком елементів
- статистичні дані про інтенсивність відмов елементів;
- задані умови експлуатації (значення поправочних коефіцієнтів).

Допуски при розрахунку:

- розрахунок ведеться для періоду нормальної експлуатації, коли величина постійна;
- відмова - події випадкові і незалежні;
- відмова одного елемента призводить до відмови всієї системи.

5.1.2 Розрахунок надійності. Математичні моделі для розрахунку значень експлуатаційної інтенсивності відмов:

$$\lambda_{\text{э}} = \lambda_0 \text{ПКі або } \lambda_{\text{н}} \text{ПКі}, \quad (5.1)$$

$$\lambda_{\text{э}} = \lambda_{\text{ос. г}} \text{ПКі або } \lambda_{\text{н.с.г}} \text{ПКі}, \quad (5.2)$$

де $\lambda_{\text{э}}$ – експлуатаційна інтенсивність відмови в реальних умовах;

λ_o – інтенсивність відмов в нормальних умовах;

ПКі – сума експлуатаційної інтенсивності відмов.

Нормальними умовами вважаються: температура навколишнього середовища 18-23 С; вологість 60-70%; атмосферний тиск 760 мм ртутного стовпа.

Модель (5.1) застосовують при розрахунку експлуатаційної інтенсивності відмов тих типів виробів, для яких в довіднику наведені значення λ_o , (λ_n).

Модель (5.2) застосовують при розрахунку експлуатаційної інтенсивності відмов тих типів виробів, для яких в довіднику відсутні значення λ_o , (λ_n) і для оцінки інтенсивності відмов групи виробів в цілому.

5.1.2.1 Складові, що входять до моделі для всіх класів ВЕТ:

– λ_o , $\lambda_{oc. \Gamma}$ – інтенсивність відмов типу (групи) ВЕТ, яка відповідає температурі навколишнього середовища 25° С і номінальному електричному навантаженню;

– λ_n , $\lambda_{n.c. \Gamma}$ – інтенсивність відмов типу (групи) ВЕТ, яка відповідає номінальному електричному навантаженню і максимально допустимій по ТУ температурі навколишнього середовища, яка відповідає цьому навантаженню;

– K_p (K_t) – коефіцієнт режиму, який залежить від електричного навантаження і (або) температури навколишнього середовища;

– K_e – коефіцієнт, який залежить від умов експлуатації;

– $K_{ам}$ – коефіцієнт, який залежить від наявності амортизації;

– $K_{к, обл}$ – коефіцієнт, який залежить від якості обслуговування РЕА.

5.1.2.2 Складові, що входять до моделі для конкретних класів ВЕТ.

5.1.2.2.1 Інтегральні мікросхеми:

– $K_{сл}$ – коефіцієнт, який враховує ступінь інтеграції і температуру кристала (корпусу);

– $K_{попр}$ – коефіцієнт, який враховує передбачуване зниження λ при експлуатації І²С в полегшених режимах і при проведенні спеціальних заходів щодо забезпечення надійності ВЕТ.

5.1.2.2.2 Напівпровідникові пристрої:

- K_{ϕ} – коефіцієнт, який враховує функціональне призначення приладу;
- $K_{д.н.}$ – коефіцієнт залежний від величини максимально допустимого по ТУ навантаження по потужності розсіювання (струму);
- K_{s1} – коефіцієнт залежний від величини відношення робочої напруги до максимально допустимої по ТУ;
- K_T – коефіцієнт, який залежить від навантаження по напрузі і температури переходу для потужних транзисторів НВЧ;
- K_f – коефіцієнт, залежний від частоти і потужності в імпульсі, для потужних транзисторів НВЧ.

5.1.2.2.3 Конденсатори:

- K_c – коефіцієнт, залежний від величини номінальної ємності;
- $K_{п.с}$ – коефіцієнт, який залежить від величини послідовного активного опору в схемі між конденсатором і джерелом живлення, для окисел напівпровідникових конденсаторів.

5.1.2.2.4 Резистори:

- K_r – коефіцієнт, залежний від величини номінального опору;
- K_m – коефіцієнт, залежний від величини номінальної потужності, для метало-композитних резисторів;
- K_{s1} – коефіцієнт, залежний від величини відношення робочої напруги на резисторі до напруги максимально допустимому за ТУ, для змінних не дротяних резисторів.

Ймовірністю називається ймовірність того, що протягом заданого часу, після закінчення якого не буде жодної відмови. Залежність $P(t) = f(t)$ – це є основний закон надійності. Основний закон надійності – ймовірність безвідмовної роботи зменшується згодом по експоненті закону надійності (рис.5.1).

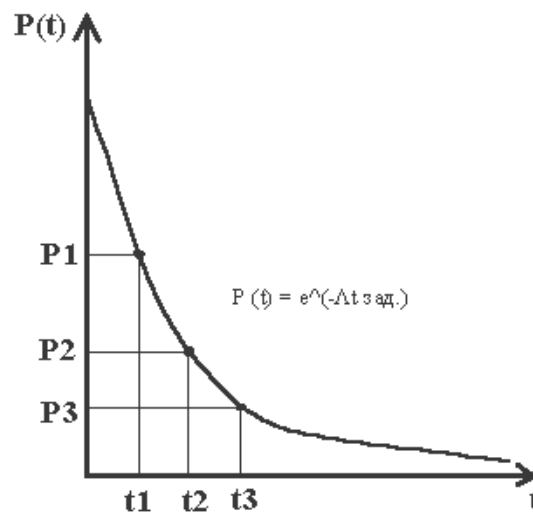


Рисунок 5.1 – Експонентний закон надійності

де $P(t)$ – це ймовірність, того, що протягом заданого часу в системі не буде жодної відмови;

$t_{\text{зад.}}$ – заданий час напрацювання;

e – основа натуральних логарифмів.

5.2 Розрахункова частина

5.2.1 Розрахунок значення експлуатаційної інтенсивності відмов конденсаторів керамічних 1206-CE-5% і танталових 1206-TAN-15% вир-ва: Bourns.

Значення інтенсивності відмов ємностей в умовах експлуатації розраховують за моделлю:

$$\lambda_e = \lambda_{\text{ос.г}} \times K_p \times K_e \times K_e, \quad (5.3)$$

Для керамічних конденсаторів:

$$\lambda_e = 0,013 \times \times 2 \times 0,15 \times 0,6 = 0,00234 \times 10^{-6},$$

Для танталових конденсаторів:

$$\lambda_e = 0,05 \times \times 2 \times 0,26 \times 2 = 0,052 \times 10^{-6}.$$

Значення складових частин формули знаходяться по методичному посібнику. Результати розрахунку заносимо в таблицю 5.1.

Таблиця 5.1 – Значення експлуатаційної інтенсивності відмов конденсаторів

позначення	Тип	$\lambda_{ос.г} \times 10^{-6}$	Ке	Кр	Ке	$\lambda_э \times 10^{-6}$
C1	1206-TAN47mF-15%	0,05	2	0,26	2	0,052
C2	1206-CE100nF-5%	0,013	2	0,15	0,71	0,002769
C3	1206-CE22pF-5%	0,013	2	0,15	0,71	0,002769
C4	1206-CE22pF-5%	0,013	2	0,15	0,71	0,002769
C5	1206-TAN47mF-15%	0,05	2	0,26	2	0,052
C6	1206-CE100 nF -5%	0,013	2	0,15	0,71	0,002769
C7	1206-CE100 nF -5%	0,013	2	0,15	0,71	0,002769
C8	1206-CE100 nF -5%	0,013	2	0,15	0,71	0,002769
C9	1206-CE100 nF -5%	0,013	2	0,15	0,71	0,002769
C10	1206-CE100 nF -5%	0,013	2	0,15	0,71	0,002769
C1	1206-CE1mF-5%	0,013	2	0,15	0,71	0,002769
C2	1206-TAN47mF-15%	0,05	2	0,26	2	0,052
C3	1206-CE1mF-5%	0,013	2	0,15	0,71	0,002769
C4	1206-CE1mF-5%	0,013	2	0,15	0,71	0,002769
C5	1206-CE100 nF -5%	0,013	2	0,15	0,71	0,002769

Продовження таблиці 5.1

C6	1206-CE100 nF -5%	0,013	2	0,15	0,71	0,002769
C7	1206-CE100 nF -5%	0,013	2	0,15	0,71	0,002769
C8	1206-CE1mF-5%	0,013	2	0,15	0,71	0,002769
C9	1206-CE100 nF -5%	0,013	2	0,15	0,71	0,002769
C10	1206-CE100 nF -5%	0,013	2	0,15	0,71	0,002769
C11	1206-CE100 nF -5%	0,013	2	0,15	0,71	0,002769
C12	1206-CE22pF-5%	0,013	2	0,15	0,71	0,002769
C13	1206-CE22pF-5%	0,013	2	0,15	0,71	0,002769
Всього						0,21138

5.2.2 Розрахунок значення експлуатаційної інтенсивності відмов мікросхем. Для мікросхем значення інтенсивності відмов в умовах експлуатації розраховують за моделлю:

$$\Lambda_e = \lambda_{oc,r} \times K_{СП} \times K_e, \quad (5.4)$$

$$\lambda_e = 0,52 \times 1,8 \times 2,4 = 2,2464 \times 10^{-6}.$$

Значення складових частин формули знаходяться по методичному посібнику. Результати розрахунку заносимо в таблиці 5.2.

Таблиця 5.2 – Значення експлуатаційної інтенсивності відмов мікросхем

позначення	Тип	$\lambda_{oc,r} \times 10^{-6}$	$K_{СП}$	K_e	$\lambda_e \times 10^{-6}$
DA1	LM317	0,1	1,04	1,3	0,1352
DA2	L7805	0,1	1,04	1,3	0,1352
DD1	XB24-AWI-001	0,52	1,8	2,4	2,2464
DD2	DS1307	0,21	1,27	1,7	0,45339
DD3	24C512	0,21	1,27	1,7	0,45339
DD4	ATmega32	0,43	1,6	1,9	1,3072

Продовження таблиці 5.2

DD1	ST232	0,21	1,27	1,7	0,45339
DD2	XB24-AWI-001	0,52	1,8	2,4	2,2464
DD3	DS1307	0,21	1,27	1,7	0,45339
DD4	24C512	0,21	1,27	1,7	0,45339
DD5	ATmega32	0,43	1,6	1,9	1,3072
DD6	74HC245N	0,21	1,27	1,7	0,45339
DD7	74HC245N	0,21	1,27	1,7	0,45339
ГКІЮ.					
2xDD1 – DD8	DM314	0,18	1,27	1,6	$\sum 5,852$
Всього					14,156993

5.2.3 Розрахунок значення експлуатаційної інтенсивності відмов індикатора ACM2004D. ACM2004D - рідко - кристалічний алфавітно-цифровий індикатор.

Для індикатора типу ACM2004D значення інтенсивності відмов в умовах експлуатації розраховують за моделлю:

$$\lambda_i = \lambda_{oc.r} \times K_{\lambda}, \quad (5.5)$$

$$\lambda_{\lambda} = 0,25 \times 2 = 0,5 \times 10^{-6}.$$

Значення складових частин формули знаходяться по методичному посібнику. Результати розрахунку заносимо в таблицю 5.3.

Таблиця 5.3 – Значення експлуатаційної інтенсивності відмов індикатора

Позначення	Тип	$\lambda_{oc.r} \times 10^{-6}$	K_{λ}	$\lambda_{\lambda} \times 10^{-6}$
HG1	ACM2004D	0,25	2	0,5
ВСЬОГО				0,5

5.2.4 Розрахунок значення експлуатаційної інтенсивності відмов резисторів типу 1206-CRE-5% вир-ва: Bourne і підлаштувальних резисторів СП5-1В. 1206-CRE-5% – резистор постійний керамічний метало-композитний.

Для резисторів типу 1206-CRE-5% і СП5-1 значення інтенсивності відмов в умовах експлуатації розраховують за моделлю:

$$\lambda_3 = \lambda_{oc.2} \times K_p \times K_3 \times K_r \times K_m, \quad (5.6)$$

$$\lambda_3 = 0,01 \times 10^{-6} \times 2 \times 0,57 \times 0,7 \times 0,7 = 0,005586 \times 10^{-6}.$$

Значення: $\lambda_{oc.г}$, K_p , K_e , K_r , K_{s1} , K_m знаходимо згідно довідкового посібника з розрахунку інтенсивності відмов. Результати розрахунку заносимо в таблицю 5.4.

Таблиця 5.4 – Значення експлуатаційної інтенсивності відмов резисторів

позначення	Тип	$\lambda_{oc.г} \times 10^{-6}$	K_e	K_p	K_m	K_r	$\lambda_3 \times 10^{-6}$
R1	СП5 – 1 – 1,5кОм	0,03	2	0,6	0,7	0,7	0,01764
R2	1206-CRE-220R-5%	0,01	2	0,57	0,7	0,7	0,005586
R3	1206-CRE-1кR-5%	0,01	2	0,57	0,7	0,7	0,005586
R4	1206-CRE-10кR-5%	0,01	2	0,57	0,7	0,7	0,005586
R5	1206-CRE-4,7кR-5%	0,01	2	0,57	0,7	0,7	0,005586
R6	1206-CRE-4,7кR-5%	0,01	2	0,57	0,7	0,7	0,005586
R7	СП5 – 1 -- 10кОм	0,03	2	0,6	0,7	0,7	0,01764
R8	1206-CRE-2,2R-5%	0,01	2	0,57	0,7	0,7	0,005586
R1	1206-CRE-4,7кR5%	0,01	2	0,57	0,7	0,7	0,005586
R2	1206-CRE-120R-5%	0,01	2	0,57	0,7	0,7	0,005586
R3	1206-CRE-120R-5%	0,01	2	0,57	0,7	0,7	0,005586
R4	1206-CRE-120R-5%	0,01	2	0,57	0,7	0,7	0,005586

Продовження таблиці 5.4

R5	1206-CRE-120R-5%	0,01	2	0,57	0,7	0,7	0,005586
Позначення	Тип	$\lambda_{oc.r} \times 10^{-6}$	Ке	Кр	Км	Кг	$\lambda_3 \times 10^{-6}$
R6	1206-CRE-10кR-5%	0,01	2	0,57	0,7	0,7	0,005586
R7	1206-CRE-4,7кR-5%	0,01	2	0,57	0,7	0,7	0,005586
R8	1206-CRE-4,7кR-5%	0,01	2	0,57	0,7	0,7	0,005586
R1	1206-CRE-1кR-5%	0,01	2	0,57	0,7	0,7	0,005586
R2	1206-CRE-1кR5%	0,01	2	0,57	0,7	0,7	0,005586
R3	1206-CRE-1кR-5%	0,01	2	0,57	0,7	0,7	0,005586
R4	1206-CRE-1кR-5%	0,01	2	0,57	0,7	0,7	0,005586
R5	1206-CRE-1кR-5%	0,01	2	0,57	0,7	0,7	0,005586
R6	1206-CRE-1кR-5%	0,01	2	0,57	0,7	0,7	0,005586
R7	1206-CRE-1кR-5%	0,01	2	0,57	0,7	0,7	0,005586
R8	1206-CRE-1кR-5%	0,01	2	0,57	0,7	0,7	0,005586
Всього							0,15817

5.2.5 Розрахунок значення експлуатаційної інтенсивності відмов діодів. Значення інтенсивності відмов діодів в умовах експлуатації розраховують за моделлю:

$$\lambda_3 = \lambda_{oc.z} \times K_p \times K_3 \times K_\phi \times K_{s1} \times K_{он}, \quad (5.7)$$

$$\lambda_3 = 0,12 \times 10^{-6} \times 0,151 \times 1 \times 1 \times 0,5 \times 1,5 = 0,01359 \times 10^{-6}.$$

Результати розрахунку заносимо в таблицю 5.5.

Таблиця 5.5 – Значення експлуатаційної інтенсивності відмов діодів

Позначення	Тип	$\lambda_{oc.г} \times 10^{-6}$	K_p	K_ϕ	$K_{дн}$	K_{s1}	K_ϵ	$\lambda_3 \times 10^{-6}$
VD1	1N5819	0,12	0,151	1	1	0,5	1,5	0,01359
VD2	1N5819	0,12	0,151	1	1	0,5	1,5	0,01359
VD3	GNL-5013 GYW	0,05	0,151	1	1	0,7	1,5	0,00793
VD4	1N5819	0,12	0,151	1	1	0,5	1,5	0,01359
2xVD1	1N5819	0,12	0,151	1	1	0,5	1,5	0,01359
4xVD1 – VD64	GNL-5013UED	0,05	0,151	1	1	0,7	1,5	$\Sigma 2,0294$
Всього								2,09169

5.2.6 Розрахунок значення експлуатаційної інтенсивності відмов транзисторів. Значення інтенсивності відмов транзистора в умовах експлуатації розраховують за моделлю:

$$\lambda_3 = \lambda_{oc.г} \times K_p \times K_3 \times K_\phi \times K_{s1} \times K_{дн}, \quad (5.8)$$

$$\lambda_3 = 0,29 \times 10^{-6} \times 0,24 \times 1 \times 0,8 \times 1,5 \times 0,5 = 0,042 \times 10^{-6}.$$

Результати розрахунку заносимо в таблицю 5.6.

Таблиця 5.6 – Значення експлуатаційної інтенсивності відмови транзистора

Познач..	Тип	$\lambda_{oc.г} \times 10^{-6}$	K_p	K_ϕ	$K_{дн}$	K_ϵ	K_{s1}	$\lambda_3 \times 10^{-6}$
VT1	IRLML25-02	0,29	0,24	1	0,8	1,5	0,5	0,042
Всього								0,042

5.2.7 Розрахунок значення експлуатаційної інтенсивності відмов роз'ємів. Значення інтенсивності відмов роз'ємів в умовах експлуатації розраховують за моделлю:

$$\lambda_{\Sigma} = \lambda_{oc.г} \times K_p \times K_{kk} \times K_{kc} \times K_e, \quad (5.9)$$

$$\lambda_{\Sigma} = 0,002 \times 10^{-6} \times 3 \times 0,58 \times 2,02 \times 0,39 \times 1,5 = 0,0016 \times 10^{-6}.$$

Результати розрахунку заносимо в таблицю 5.7.

Таблиця 5.7 – Значення експлуатаційної інтенсивності відмов роз'ємів

Познач.	Тип	$\lambda_{oc.г} \times 10^{-6}$	K_p	K_{kk}	K_{kc}	K_e	$\lambda_{\Sigma} \times 10^{-6}$
XP1	Power_con	0,0023	0,58	2,02	0,39	1,5	0,0016
XP2	PLS-40-06	0,0007	0,35	1,4	0,39	1,1	0,00014
XP3	PS/2M-Con	0,0023	0,58	2,02	0,39	1,5	0,0016
XP1	Power_con	0,0023	0,58	2,02	0,39	1,5	0,0016
XP2	MW – 03M	0,0023	0,58	2,02	0,39	1,5	0,0016
XP3	PLS-40 - 06	0,0007	0,35	1,4	0,39	1,1	0,00014
XP4	DB9M	0,0023	0,58	2,02	0,39	1,5	0,0016
XS1	BH-20	0,0023	0,58	2,02	0,39	1,5	0,0016
XS1	BH-20R	0,0023	0,58	2,02	0,39	1,5	0,0016
XS	15xPLS-40	0,0007	0,35	1,4	0,39	1,1	0,0021
XP	15xBPS-40	0,0007	0,35	1,4	0,39	1,1	0,0021
Всього							0,01568

5.2.8 Розрахунок значення експлуатаційної інтенсивності відмов комутаційних виробів. Значення інтенсивності відмов комутаційних виробів в умовах експлуатації розраховують за моделлю:

$$\lambda_{\Sigma} = \lambda_{oc.g} \times K_p \times K_{kk} \times K_e, \quad (5.10)$$

$$\lambda_{\Sigma} = 0,0045 \times 10^{-6} \times 0,6 \times 1,6 \times 1,2 = 0,005184 \times 10^{-6}.$$

Значення складових частин формули знаходяться по методичному посібнику.

Результати розрахунку заносимо в таблицю 5.8.

Таблиця 5.8 – Значення експлуатаційної інтенсивності відмов комутаційних виробів

Позначення	Тип	$\lambda_{oc.g} \times 10^{-6}$	K_p	K_{kk}	K_e	$\lambda_{\Sigma} \times 10^{-6}$
SA1	RK2-14	0,0045	0,6	1,6	1,2	0,005184
KP1	Ak 304-N-BBW	0,0018	0,6	1,6	1,01	0,005184
Всього						0,006929

5.2.9 Розрахунок значення експлуатаційної інтенсивності відмов кварцових резонаторів. Значення інтенсивності відмов кварцових резонаторів в умовах експлуатації розраховують за моделлю:

$$\lambda_{\Sigma} = \lambda_{oc.g} \times K_p \times K_{\Sigma} \times K_{\phi} \times K_{sl} \times K_{\partial n}, \quad (5.11)$$

$$\lambda_{\Sigma} \times 4 = 0,12 \times 10^{-6} \times 0,151 \times 1 \times 1 \times 0,5 \times 1,5 = 0,01359 \times 10^{-6} \times 4 = 0,05436 \times 10^{-6}.$$

5.2.10 Розрахунок сумарної інтенсивності відмов приладу проведемо за допомогою наступного виразу:

$$\lambda_{\Sigma} = \sum_{I=1}^N \lambda_{\Sigma} = 17,1828 \times 10^{-6}, \quad (5.12)$$

5.2.11 Розрахунок середнього часу приробітку виробу на одну відмову при безперервній роботі протягом 24 годин проводимо за допомогою наступної формули:

$$T_{сер} = 1/\lambda_{np}, \quad (5.13)$$

$$\lambda_{np} = \frac{1}{17,1828 \times 10^{-6}} = 0,05819 \times 10^6 \text{ год} = 6,6 \text{ рок.}$$

5.2.12 Розрахунок ймовірності безвідмовної роботи приладу проведемо за формулою:

$$P = e^{-\lambda t}, \quad (5.14)$$

5.2.13 Розрахунок кроку t для графіка $P = f(t)$ проведемо за допомогою наступного виразу:

$$h = T_{сер}/5, \quad (5.15)$$

де: h – крок, година;

$T_{сер}$ – середній час підробітку виробу на одну відмову, годину.

$$h = 0,05819 \times 10^6 / 5 = 0,011638 \times 10^6 \text{ год} \approx 0,0116 \times 10^6.$$

Вихідні дані для побудови графіка $P = f(t)$ заносимо в таблицю 5.9. По таблиці 5.9 будемо графік наведений на рисунку 5.2.

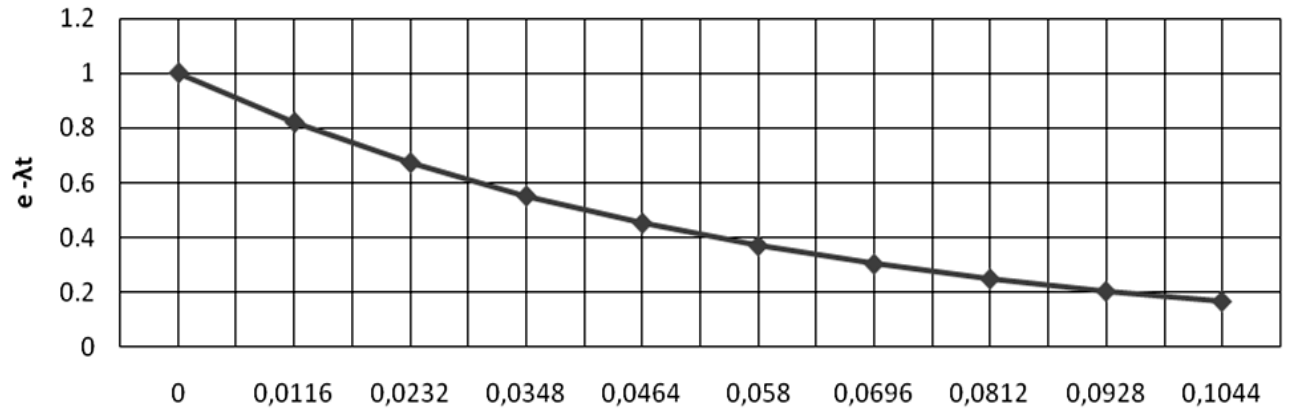


Рисунок 5.2 – Графік ймовірності безвідмовної роботи пристрою

Таблиця 5.9 – Розрахунок ймовірності безвідмовної роботи

$t \cdot 10^6$	0	0,0116	0,0232	0,0348	0,0464	0,058	0,0696	0,0812	0,0928	0,1044
$\lambda \cdot t$	0	0,199	0,3986	0,5979	0,7972	0,9966	1,1959	1,3952	1,5945	1,7939
$e^{-\lambda t}$	1	0.820	0.673	0.552	0.453	0.371	0.304	0.250	0.205	0.168

ВИСНОВКИ

У процесі виконання дипломного проекту був розроблений світлодіодний модуль відображення графічної інформації з керуванням по радіоканалу.

Розроблений пристрій має наступні характеристики:

- змінне поле масиву – так (практично без обмежень);
- вивід кириличних і латинських букв великого і малого регістру, вивід цифр, розділових знаків і спеціальних символів – так;
- вивід температури повітря навколишнього середовища – $10 \dots + 85^{\circ}\text{C}$;
- вивід поточного часу і дати – так (до 2099 року);
- управління за допомогою персонального комп'ютера – так (опціонально);
- управління дистанційне за допомогою пульта управління – так;
- довжина тексту, що вводиться – до 32000 символів;
- висота символу – 8 пікселів;
- оптимальна дистанція огляду – 5-15 метрів.

Були проведені електричні розрахунки деяких вузлів схеми. Спроектовано схеми електричні: структурні і принципові. Побудовано алгоритми роботи програм і написано машинні коди основних програм.

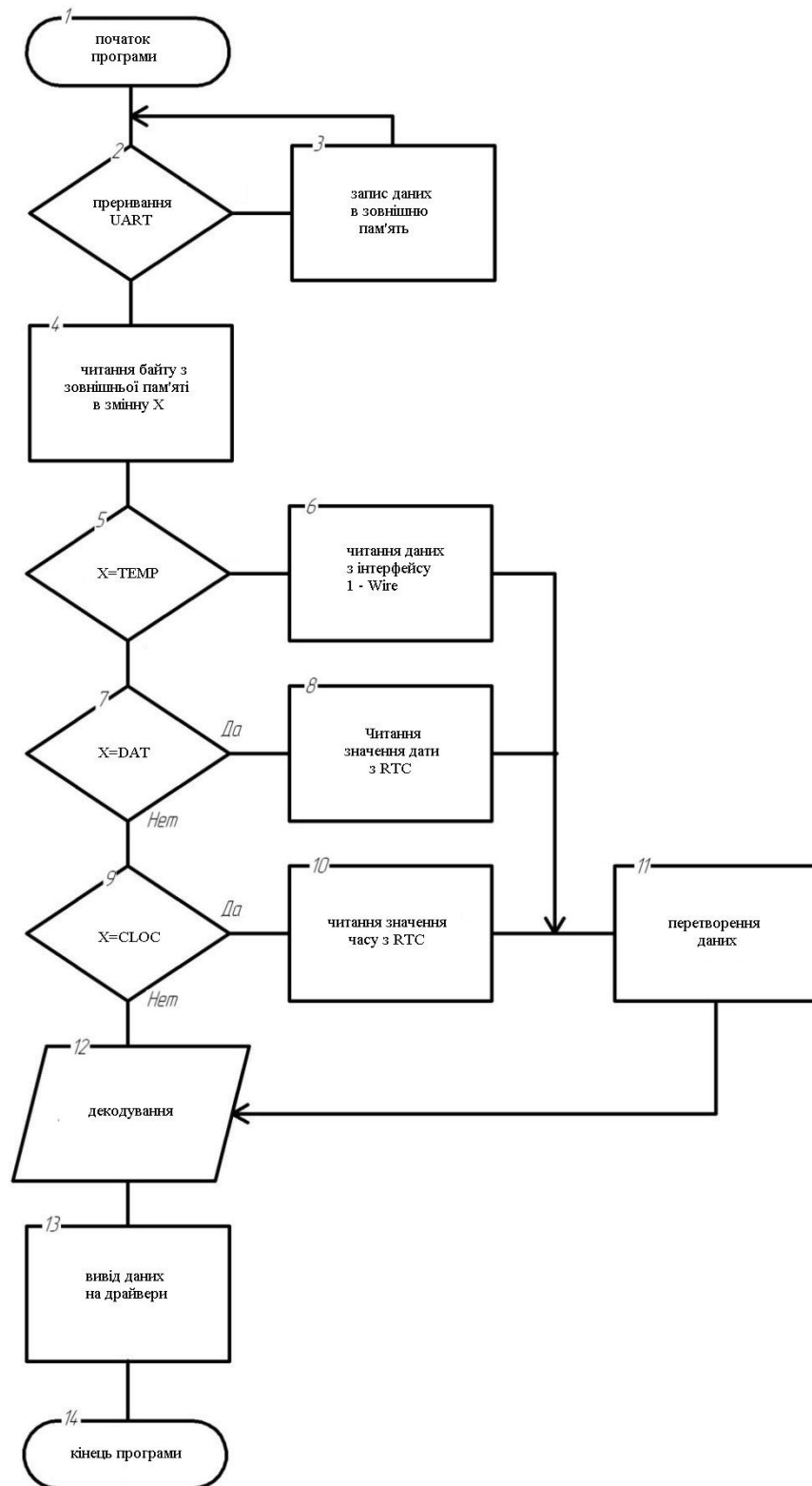
Проведено розрахунок надійності пристрою і визначено ймовірний безвідмовний термін роботи пристрою, який становить 6,6 років, що є хорошим показником для пристроїв даного типу.

ПЕРЕЛІК ПОСИЛАНЬ

1. Белов А.В. Конструирование устройств на микроконтроллерах [Текст] / А.В. Белов. – Санкт Петербург: Наука и Техника, 2004. – 428 с.
2. Турута Е.Ф. Зарубежные транзисторы: справочник [Текст] / Е.Ф. Турута. – Санкт Петербург: Горячая Линия-Телеком, 2004. – 756 с.
3. Евстифеев А.В. Микроконтроллеры AVR семейств Tiny и Mega фирмы Atmel [Текст] / А.В. Евстифеев. – М.: Додэка XXI, 2005. – 560 с.
4. Трамперт В. AVR-RISC микроконтроллеры [Текст] / В.Трамперт. – Киев: МК – Пресс, 2006. – 464 с.
5. Карпуков Л.М., Ключко В.В., Місюра А.О. Методичні вказівки до виконання дипломних проектів та магістерських робіт для студентів спеціальностей 7.160105 – "Захист інформації в комп'ютерних системах і мережах" та 7.160103 – "Системи захисту від несанкціонованого доступу" усіх форм навчання [Текст] / Л.М. Карпуков, В.В. Ключко, А.О. Місюра. – Запоріжжя: ЗНТУ, 2007. – 50 с.
6. Ревич Ю. Практическое программирование микроконтроллеров Atmel AVR [Текст] / – Санкт Петербург: БХВ-Петербург, 2008. – 379 с.
7. Зарубежные микросхемы, транзисторы, тиристоры, диоды + SMD. 0...9: справочник. // По материалам VRL 2008 – Санкт Петербург: Наука и техника, 2008. – 672 с.
8. Зарубежные микросхемы, транзисторы, тиристоры, диоды+SMD. A...Z. Том 1 (A-R): справочник // По материалам VRL 2008 – Санкт Петербург: Наука и техника, 2008. – 816 с.
9. SP – DM134_5_6A.010.datasheet // SILICON TOUCH TECHNOLOGY INC., 2003 – 18 p.
10. 11160_24c512 datasheet. Cheyenne Mtn. Blvd. Colorado Springs, CO 80906, 2007. – 22 p.

Додаток А

БЛОК-СХЕМА АЛГОРИТМУ ПРОГРАМИ БЛОКУ КОНТРОЛЕРА



Додаток Б

Лістинг програми мікроконтролера (Пульт керування)

```

('*****
This program For RunLin pult
This program is the propert of
Danilov Igor Gennady, developed in an
assistance with pridpriyatiem "Elensi"
and registered under the trade mark of
LedArtis tm.
*****')

$Device= m32                ' used device
$Stack = 120                ' stack depth
$Clock = 16                 ' used cristal
$Baud=9600                  ' baynd
$Key Rows=PORTC &h0f, Cols=PORTC &hf0,150    ' keybord 4x3
$Lcd=PORTA.3, RS=PORTA.2, EN=PORTA.1, 20, 4  ' lcd
$I2C SDA=PORTD.7, SCL=PORTD.6                ' I2C
$PcKey Data=PORTB.0, Clock=PORTB.1           ' PS/2 keybord

'===== Declare Sub =====
Declare Sub Select_key()
Declare Sub Init_clok_lcd()
Declare Sub GetTime()
Declare Sub Key_1()
Declare Sub Key_2()
Declare Sub key_3()
Declare Sub Key_4()
Declare Sub Key_reset()
Declare Sub IInit_main()
Declare Sub Lok()
Declare Sub Lok_set()
Declare Sub ps_2()
Declare Sub Stroc_LCD()
Declare Sub Bac_spas()
Declare Sub Enter()
Declare Sub Left_ps()
Declare Sub Down()
Declare Sub Right_ps()
Declare Sub Up()
Declare Sub ps_2_shift()
Declare Sub Cap()
Declare Sub Led()
Declare Sub Read_24lc512()
Declare Sub Write_24lc512()
Declare Sub No_shift()
Declare Sub Erase_24()

```

```

Declare Sub Key_5()
Declare Sub Shift_on()
Declare Sub Rus_no_shift()
Declare Sub Rus_shift()
Declare Sub Ukr_no_shift()
Declare Sub Ukr_shift()
Declare Sub Select_lang()
Declare Sub PC_read()
Declare Sub Erase_24_bs()
Declare Sub way()
Declare Sub Write_ef()
Declare Sub f_key()
Declare Sub Clok_set()
Declare Sub Write_clok()
Declare Sub RTC_Init()
Declare Sub IN1()
Declare Sub Transmit_cloc()
Declare Sub Transmit()
Declare Sub Read_24lc512_tx()
'=====
DefLcdChar 0, &h02, &h04, &h04, &h08, &h04, &h04, &h02, &h00 ' {
DefLcdChar 1, &h08, &h04, &h04, &h02, &h04, &h04, &h08, &h00 ' }
DefLcdChar 2, &h0E, &h11, &h10, &h1C, &h10, &h11, &h0E, &h00 ' C
DefLcdChar 3, &h00, &h00, &h0E, &h11, &h1C, &h11, &h0E, &h00 ' e
DefLcdChar 4, &h0A, &h00, &h0E, &h04, &h04, &h04, &h0E, &h00 ' İ
DefLcdChar 5, &h00, &h0A, &h00, &h0C, &h04, &h04, &h0E, &h00 ' i
DefLcdChar 6, &h0A, &h15, &h0A, &h15, &h0A, &h15, &h0A, &h00 ' date
DefLcdChar 7, &h0E, &h15, &h15, &h17, &h11, &h11, &h0E, &h00 ' time

'=====
'----- Peremennue -----
'=====
Dim i As Byte, j As Byte, k As Byte, m As Byte, c As Byte, d As Byte
Dim day As Byte, mon As Byte, year As Byte, dof As Byte
Dim hour As Byte, min As Byte, Minus As Byte, isec As Byte, iadDD As Word
Dim second As Byte, wil As Byte, kp1 As Byte, kp2 As Byte, kp3 As Byte
Dim kp4 As Byte, kp5 As Byte, kpres As Byte, ee_adr As Byte, iadDD1 As Word
Dim lok1 As Byte, lok2 As Byte, lok3 As Byte, lok4 As Byte, shi As Word
Dim lk1 As Byte, lk2 As Byte, lk3 As Byte, lk4 As Byte, kk As Byte, kn As Byte
Dim kl1 As Byte, kl2 As Byte, kl3 As Byte, kl4 As Byte, lk5 As Byte
Dim lokk1 As Byte, lokk2 As Byte, lokk3 As Byte, lokk4 As Byte, lokk5 As Byte
Dim keydata As Word, kd As Byte, keydata_ASCII As Byte, keydata1 As Word
Dim keyps As Byte, lcd_st As Word, lcd_ms As Word, st1 As Integer, bs As Word
Dim SH As Byte, cl As Byte, led1 As Byte, aDDR As Word, arDDR As Word
Dim kdata As Word, DAT As Byte, dd As Word, i1 As Byte, err As Word
Dim erer As Word, eer As Word, kshift As Word, krot As Word, ter As Byte
Dim lang As Word, lang_set As Word, lang_set_sh As Word, ppc As Word
Dim adDDR As Word, ef_rd As Word, ef_bit As Byte, adDD1 As Byte, adEE1 As Byte
Dim day1 As Byte, mon1 As Byte, year1 As Byte, dof1 As Byte, adDD As Byte
Dim hour1 As Byte, min1 As Byte, Minus1 As Byte, isec1 As Byte, adEE As Word
Dim second1 As Byte, kp9 As Byte, fun As Byte, bf As Byte, rt As Byte, tx As Byte
Dim tx1 As Byte, sa As Byte, sd As Byte, flag_red As Byte, arfDDR As Word

```

```

Dim ad_run As Byte, end_i As Byte
'=====

'===== Constante =====
Const DS1307=&hd0
'=====

'===== Ykazanie portov =====
PORTA.7=0
PORTD.2=0
DDRD.2=1
DDRA.7=1
adDD=0
adDD1=0
'=====

'===== MainLoop: =====
Do
    'WriteEE(&h0035, &ha3)
    sa=ReadEE(&h0035)
    If sa=&ha1 Then
        Lok()
    Else
        IInit_main()
    End If
Loop
'=====

'===== IInit_main =====
Sub IInit_main()
    Cls
    WaitMs 50
    Cursor Off
    Locate 1,7
    Lcd "RunLin "
    Locate 2,1
    Lcd "1-Word text 2-TX txt"
    Locate 3,1
    Lcd "3-Seting 4-Read"
    For wil=0 To 255
        Select_key()
        Init_clok_lcd()
        If wil=255 Then
            wil=0
        End If
    Next
End Sub
'=====

'===== Select keybord =====
Sub Select_key()
    k=0

```

```

While k=0
    k=0
    k=Key()
    Init_clok_lcd()          ' if key is pressed
Wend
    Select Case k
    Case 58      'key0
        Led()
    Case 33      'key1
        Key_1()
    Case 34      'key2
        Key_3()
    Case 35      'key3
        Key_4()
    Case 41      'key4
        Key_5()
    Case 59      'key#
        Key_reset()
    End Select

End Sub

'=====
'===== Indikation clock =====
Sub Init_clok_lcd()
    GetTime()
    Locate 4,1
    Lcd Hex(day);"/";Hex(mon)
    Locate 4,16
    Lcd Hex(hour);":";Hex(min)
End Sub
'=====

'===== Read time =====
Sub GetTime()
    I2Cstart DS1307
    I2Cwrite 0
    second=I2Cread, Ack
    min=I2Cread, Ack
    hour=I2Cread, Ack
    dof=I2Cread, Ack
    day=I2Cread, Ack
    mon=I2Cread, Ack
    year=I2Cread, Nack
    I2Cstop
End Sub
'=====

'===== Procedyre key1 =====
Sub Key_1()
    Local tr As Byte, ty As Byte
k1:
    aDDR=&h0000

```

```

arfDDR=0
Cls
WaitMs 50
Cursor Off
Locate 1,5
Lcd "Run word text"
Locate 2,1
Lcd "1-File_1 2-File_2"
Locate 3,1
Lcd "3-File_3"

For wil=0 To 255
    kp1=0
    While kp1=0
        kp1=0
        kp1=Key()
        'Init_clok_lcd()
    Wend
    Select Case kp1
        Case 33
            Cls
            WaitMs 50
            Cursor On
            Locate 1,1
            Lcd "Text:"
            aDDR=&h0002 'write adres 24lc512
            Erase_24()
            aDDR=&h0002 'write adres 24lc512
            adEE1=&h0051
            adEE=&h0052
            lang=0
            Select_lang()
            ps_2()
        Case 34
            Cls
            WaitMs 50
            Cursor On
            Locate 1,1
            Lcd "Text:"
            aDDR=&h4000 'write adres 24lc512
            Erase_24()
            aDDR=&h4000
            adEE1=&h0053
            adEE=&h0054
            lang=0
            Select_lang()
            ps_2()
        Case 35
            Cls
            WaitMs 50
            Cursor On
            Locate 1,1

```

```

        Lcd "Text:"
        aDDR=&h8000 'write adres 24lc512
        Erase_24()
        aDDR=&h8000
        adEE1=&h0055
        adEE=&h0056
        lang=0
        Select_lang()
        ps_2()
    Case 57
        GoTo K1
    Case 59
        Key_reset()
End Select
If wil=255 Then
    wil=0
End If
Next

=====
Key_5_:

    tr=0
    ty=0
    Cls
    WaitMs 50
    Cursor Off
    Locate 1,5
    Lcd "Delet File:"
    Locate 2,7
    Lcd "1-File 1"
    Locate 3,7
    Lcd "2-File 2"
    Locate 4,7
    Lcd "3-File 3"
    While tr=0
        tr=0
        tr=Key()
        'Init_clok_lcd()
    Wend
    Select Case tr
        Case 33
            Cls
            WaitMs 50
            Cursor Off
            Locate 1,1
            Lcd "DELET"
            aDDR=&h0002 'write adres 24lc512
            Erase_24()
        Case 34
            Cls
            WaitMs 50

```

```

        Cursor Off
        Locate 1,1
        Lcd "DELET"
        aDDR=&h4000      'write adres 24lc512
        Erase_24()
    Case 35
        Cls
        WaitMs 50
        Cursor On
        Locate 1,1
        Lcd "DELET"
        aDDR=&h8000      'write adres 24lc512
        Erase_24()
    Case 59
        Key_1()
End Select
WaitMs 100
GoTo k1

End Sub

'===== Procedyre key2 =====

'=====

'===== Procedyre key3 =====
Sub Key_3()
k3:
    Cls
    WaitMs 50
    Cursor Off
    Locate 1,2
    Lcd "Transmition RunLin"
    Locate 2,1
    Lcd "1-File_1 2-File_2"
    Locate 3,1
    Lcd "3-File_3 4-Clock"
For wil=0 To 255
    kp3=0
    While kp3=0
        kp3=0
        kp3=Key()
        'Init_clok_lcd()
    Wend
    Select Case kp3
        Case 33
            Cls
            WaitMs 50
            Locate 1,1
            Lcd "File 1 Trasmite: "
            Locate 2,3
            aDDR=&h0001

```

```

        adEE1=&h0051
        adEE=&h0052
        Transmit()      'write adres 24lc512
    Case 34
        Cls
        WaitMs 50
        Locate 1,1
        Lcd "File 2 Trasmite: "
        Locate 2,3
        aDDR=&h3fff
        adEE1=&h0053
        adEE=&h0054
        Transmit()      'write adres 24lc512
    Case 35
        Cls
        WaitMs 50
        Locate 1,1
        Lcd "File 3 Trasmite: "
        Locate 2,3
        aDDR=&h7fff
        adEE1=&h0055
        adEE=&h0056
        Transmit()      'write adres 24lc512
    Case 41
        Cls
        WaitMs 50
        Locate 1,1
        Lcd "Set clock to RunLin:"
        Locate 2,4
        Transmit_cloc()
    Case 57
        GoTo k3
    Case 59
        Key_reset()
End Select
If wil=255 Then
    wil=0
End If
Next
End Sub
'=====

Sub Read_24lc512_tx()
Local tmp As Byte, ef As Byte
    adDD=ReadEE(adEE)
    adDD1=ReadEE(adEE1)
    iadDD=(adDD1*127)+adDD
    iadDD1=0
    WaitMs 15
    I2Cstart &hA0      ' 24C64 address
    I2Cwrite Msb(aDDR)  ' write adr MSB
    I2Cwrite aDDR

```

```

For tmp=0 To 15      ' write adr LSB
  For ef=0 To 249
    iadDD1=iadDD1+1
    tx=I2Cread, Nack
    PrintBin tx
    WaitMs 10
    If iadDD1=iadDD Then
      Lcd Chr(&hFF)
      tmp=I2Cread, Nack
      I2Cstop
      Exit Sub
    End If
  Next
  Lcd Chr(&hFF)
Next
  tmp=I2Cread, Nack
  I2Cstop
  'Incr aDDR
End Sub

Sub Transmit()
  Local tv As Byte
  Local mes As Bit, fg As Byte
  Local tmp As Byte, ef As Byte
  adDD=ReadEE(adEE)
  adDD1=ReadEE(adEE1)
  iadDD=(adDD1*127)+adDD
  iadDD1=0
  PORTD.2=1
  WaitMs 10
  ad_run=&hae
  end_i=&ha3
  mes=0
  PrintBin ad_run
  WaitMs 15
  I2Cstart &hA0      ' 24C64 address
  I2Cwrite Msb(aDDR)  ' write adr MSB
  I2Cwrite aDDR
For tmp=0 To 15      ' write adr LSB
  For ef=0 To 249
    iadDD1=iadDD1+1
    tx=I2Cread, Nack
    PrintBin tx
    WaitMs 10
    If iadDD1=iadDD Then
      Lcd Chr(&hFF)
      tmp=I2Cread, Nack
      I2Cstop
      GoTo h
    End If
  Next
  Lcd Chr(&hFF)

```

Next

```

        tmp=I2Cread, Nack
        I2Cstop
h:      WaitMs 10
        PrintBin end_i
        WaitMs 10
        PrintBin end_i
        WaitMs 100
            Locate 3,4
            Lcd "Transmition OK"
            PORTD.2=0
jk:
        fg=0
        While fg=0
            fg=0
            fg=Key()
        Wend
        Select Case fg
            Case 57
                Key_3()
            Case 59
                Key_reset()
            Case Else
                GoTo jk
        End Select
End Sub

```

```

Sub Transmit_cloc()
    Local tvC As Byte, ad_runc As Byte, end_c As Byte
    Local mesc As Bit, lk As Byte, tmp1 As Byte
    ad_runc=&hA1
    end_c=&h75
    end_i=&ha3
    GetTime()

    PrintBin ad_runc
    WaitMs 10

    tmp1=second And &hF0
    Swap(tmp1)
    PrintBin tmp1
    WaitMs 10
    tmp1=second And &h0F
    PrintBin tmp1
    WaitMs 10
    Lcd Chr(&hff); Chr(&hff)

    tmp1=min And &hF0
    Swap(tmp1)
    PrintBin tmp1

```

```

WaitMs 10
tmp1=min And &h0F
PrintBin tmp1
WaitMs 10
Lcd Chr(&hff); Chr(&hff)

```

```

tmp1=hour And &hF0
Swap(tmp1)
PrintBin tmp1
WaitMs 10
tmp1=hour And &h0F
PrintBin tmp1
WaitMs 10
Lcd Chr(&hff); Chr(&hff)

```

```

PrintBin dof
WaitMs 10
Lcd Chr(&hff); Chr(&hff)

```

```

tmp1=day And &hF0
Swap(tmp1)
PrintBin tmp1
WaitMs 10
tmp1=day And &h0F
PrintBin tmp1
WaitMs 10
Lcd Chr(&hff); Chr(&hff)

```

```

tmp1=mon And &hF0
Swap(tmp1)
PrintBin tmp1
WaitMs 10
tmp1=mon And &h0F
PrintBin tmp1
WaitMs 10
Lcd Chr(&hff); Chr(&hff)

```

```

tmp1=year And &hF0
Swap(tmp1)
PrintBin tmp1
WaitMs 10
tmp1=year And &h0F
PrintBin tmp1
WaitMs 10
Lcd Chr(&hff); Chr(&hff)

```

```

PrintBin end_c
WaitMs 10
    Locate 3,2
    Lcd "Clock transmit OK"

```

jk1:

```
lk=0
```

```

While lk=0
    lk=0
    lk=Key()
Wend
Select Case lk
    Case 57
        Key_3()
    Case 59
        Key_reset()
    Case Else
        GoTo jk1
End Select
End Sub

'===== Procedyre key4 =====
Sub Key_4()
k4:
    Cls
    WaitMs 50
    Cursor Off
    Locate 1,2
    Lcd "Seting"
    Locate 2,1
    Lcd "1-Lok 2-Clock"
For wil=0 To 255
    kp4=0
    While kp4=0
        kp4=0
        kp4=Key()
    Wend
    Select Case kp4
        Case 33
            Lok_set()
        Case 34
            Cls
            WaitMs 50
            Locate 1,1
            Lcd "Set clock "
            Clok_set()
        Case 57
            GoTo k4
        Case 59
            Key_reset()
    End Select
    If wil=255 Then
        wil=0
    End If
Next
End Sub

'=====

Sub Clok_set()

```

Local f As Byte, syt As Byte

 GetTime()

 f=0

 RTC_Init()

 hour1=FromBcd(hour)

 min1=FromBcd(min)

 second1=FromBcd(second)

 day1=FromBcd(day)

 mon1=FromBcd(mon)

 year1=FromBcd(year)

 dof1=FromBcd(dof)

 IN1()

go:

 Select Case f

 Case 0

 GoTo hh 'hayr

 Case 1

 GoTo mm 'min

 Case 2

 GoTo ss 'sec

 Case 3

 GoTo de 'date

 Case 4

 GoTo mo 'mesyac

 Case 5

 GoTo ye 'yar

 Case 6

 GoTo doff 'dey

 Case 7

 f=0

 GoTo go

 End Select

hh:

For wil=0 To 255

Locate 2,3

 kp9=0

 While kp9=0

 kp9=0

 kp9=Key()

 Wend

 Select Case kp9

 Case 58 'key0

 GoTo go

 Case 33 'key1

 GoTo go

 Case 34 'key2

 GoTo go

 Case 35 'key3

 GoTo go

 Case 41 'key4

 GoTo go

 Case 42

 'key5

```

        GoTo go
    Case 43      'key6
        GoTo go
    Case 51
        Incr hour1
        Incr hour1
    Case 49
        Decr hour1
        Incr hour1
    Case 50
        Incr f
        GoTo go
    Case 57
        Write_clok()
        GoTo go
    Case 59
        Key_4()
End Select
If hour1=25 Then
    hour1=1
End If
If hour1=0 Then
    hour1=24
End If
Decr hour1
Locate 2, 3
If hour1<10 Then
    Lcd "0"
End If
Lcd hour1
If wil=255 Then
    wil=0
End If
Next
mm:
For wil=0 To 255
Locate 2, 6
    kp9=0
    While kp9=0
        kp9=0
        kp9=Key()
    Wend
    Select Case kp9
        Case 58      'key0
            GoTo go
        Case 33      'key1
            GoTo go
        Case 34      'key2
            GoTo go
        Case 35      'key3
            GoTo go
        Case 41      'key4

```

```

        GoTo go
Case 42                                'key5
        GoTo go
Case 43                                'key6
        GoTo go
Case 51
        Incr min1
        Incr min1
Case 49
        Decr min1
        Incr min1
Case 50
        Incr f
        GoTo go
Case 57
        Write_clok()
        GoTo go
Case 59
        Key_4()
End Select
If min1=61 Then
        min1=1
End If
If min1=0 Then
        min1=60
End If
Decr min1
Locate 2, 6
If min1<10 Then
        Lcd "0"
End If
Lcd min1
If wil=255 Then
        wil=0
End If
Next
ss:
For wil=0 To 255
Locate 2, 9
        kp9=0
        While kp9=0
                kp9=0
                kp9=Key()
        Wend
        Select Case kp9
                Case 58                'key0
                        GoTo go
                Case 33                'key1
                        GoTo go
                Case 34                'key2
                        GoTo go
                Case 35                'key3

```

```

        GoTo go
Case 41      'key4
        GoTo go
Case 42                                'key5
        GoTo go
Case 43      'key6
        GoTo go
Case 51
        Incr second1
        Incr second1
Case 49
        Decr second1
        Incr second1
Case 50
        Incr f
        GoTo go
Case 57
        Write_clok()
        GoTo go
Case 59
        Key_4()
End Select
If second1=61 Then
        second1=1
End If
If second1=0 Then
        second1=60
End If
Decr second1
Locate 2, 9
If second1<10 Then
        Lcd "0"
End If
Lcd second1
If wil=255 Then
        wil=0
End If
Next
de:
For wil=0 To 255
Locate 3, 3
        kp9=0
        While kp9=0
                kp9=0
                kp9=Key()
        Wend
        Select Case kp9
                Case 58      'key0
                        GoTo go
                Case 33      'key1
                        GoTo go
                Case 34      'key2

```

```

        GoTo go
    Case 35      'key3
        GoTo go
    Case 41      'key4
        GoTo go
    Case 42              'key5
        GoTo go
    Case 43      'key6
        GoTo go
    Case 51
        Incr day1
        Incr day1
    Case 49
        Decr day1
        Incr day1
    Case 50
        Incr f
        GoTo go
    Case 57
        Write_clok()
        GoTo go
    Case 59
        Key_4()
End Select
If day1=33 Then
    day1=1
End If
If day1=0 Then
    day1=32
End If
Decr day1
Locate 3, 3
If day1<10 Then
    Lcd "0"
End If
Lcd day1
If wil=255 Then
    wil=0
End If
Next
mo:
For wil=0 To 255
Locate 3, 6
    kp9=0
    While kp9=0
        kp9=0
        kp9=Key()
    Wend
    Select Case kp9
        Case 58      'key0
            GoTo go
        Case 33      'key1

```

```

        GoTo go
Case 34      'key2
        GoTo go
Case 35      'key3
        GoTo go
Case 41      'key4
        GoTo go
Case 42                                     'key5
        GoTo go
Case 43      'key6
        GoTo go
Case 51
        Incr mon1
        Incr mon1
Case 49
        Decr mon1
        Incr mon1
Case 50
        Incr f
        GoTo go
Case 57
        Write_clok()
        GoTo go
Case 59
        Key_4()
End Select
If mon1=14 Then
    mon1=1
End If
If mon1=0 Then
    mon1=13
End If
Decr mon1
Locate 3, 6
If mon1<10 Then
    Lcd "0"
End If
Lcd mon1
If wil=255 Then
    wil=0
End If
Next
ye:
For wil=0 To 255
Locate 3, 11
    kp9=0
    While kp9=0
        kp9=0
        kp9=Key()
    Wend
    Select Case kp9
        Case 58      'key0

```

```

        GoTo go
    Case 33      'key1
        GoTo go
    Case 34      'key2
        GoTo go
    Case 35      'key3
        GoTo go
    Case 41      'key4
        GoTo go
    Case 42              'key5
        GoTo go
    Case 43      'key6
        GoTo go
    Case 51
        Incr year1
        Incr year1
    Case 49
        Decr year1
        Incr year1
    Case 50
        Incr f
        GoTo go
    Case 57
        Write_clok()
        GoTo go
    Case 59
        Key_4()
End Select
If year1=101 Then
    year1=1
End If
If year1=0 Then
    year1=100
End If
Decr year1
Locate 3, 11
If year1<10 Then
    Lcd "0"
End If
Lcd year1
If wil=255 Then
    wil=0
End If
Next
doff:
For wil=0 To 255
Locate 3, 18
    kp9=0
    While kp9=0
        kp9=0
        kp9=Key()
    Wend

```

```

Select Case kp9
    Case 58      'key0
        GoTo go
    Case 33      'key1
        GoTo go
    Case 34      'key2
        GoTo go
    Case 35      'key3
        GoTo go
    Case 41      'key4
        GoTo go
    Case 42      'key5
        GoTo go
    Case 43      'key6
        GoTo go
    Case 51
        Incr dof1
    Case 49
        Decr dof1
    Case 50
        Incr f
        GoTo go
    Case 57
        Write_clok()
        GoTo go
    Case 59
        Key_4()
End Select
If dof1=8 Then
    dof1=1
End If
If dof1=0 Then
    dof1=7
End If
Locate 3, 18
Lcd dof1
If wil=255 Then
    wil=0
End If
Next
End Sub

Sub Write_clok()
    I2Cstart DS1307
    I2Cwrite 0
    I2Cwrite Bcd(second1)
    I2Cwrite Bcd(min1)
    I2Cwrite Bcd(hour1)
    I2Cwrite Bcd(dof1)
    I2Cwrite Bcd(day1)
    I2Cwrite Bcd(mon1)
    I2Cwrite Bcd(year1)

```

```

        I2Cstop
End Sub

Sub IN1()
    Cls
    WaitMs 10
    Cursor On
    Locate 1, 2
    Lcd "Seting clock"
    Locate 4, 14
    Lcd "#-Es"
    Locate 2,3
    If hour1<10 Then
        Lcd "0"
    End If
    Lcd hour1
    Locate 2, 6
    If min1<10 Then
        Lcd "0"
    End If
    Lcd min1
    Locate 2, 9
    If second1<10 Then
        Lcd "0"
    End If
    Lcd second1
    Locate 3, 3
    If day1<10 Then
        Lcd "0"
    End If
    Lcd day1
    Locate 3, 6
    If mon1<10 Then
        Lcd "0"
    End If
    Lcd mon1
    Locate 3, 11
    If year1<10 Then
        Lcd "0"
    End If
    Lcd year1
    Locate 3, 18
    Lcd dof1
    Locate 3,5
    Lcd "/"
    Locate 3,8
    Lcd "/20"
    Locate 2,5
    Lcd ":"
    Locate 2,8
    Lcd ":"
    Locate 3,15

```

```

    Lcd "D: ";dof1
    Locate 4,1
    Lcd Chr(&hdb);"-7"
    Locate 4,18
    Lcd "9-";Chr(&hdc)
    Locate 4,5
    Lcd "8-S"
    Locate 4,10
    Lcd "*-W"
End Sub

Sub RTC_Init()
Nop
Nop
    I2Cstart DS1307
    I2Cwrite 0
    isec=I2Cread,Nack
    I2Cstop

    isec=isec And &h7F
    I2Cstart DS1307
    I2Cwrite 0
    I2Cwrite isec
    I2Cstop

    I2Cstart DS1307
    I2Cwrite 7
    I2Cwrite 144'147
    I2Cstop
End Sub
'===== Key reset procedyre
Sub Key_reset()
    IInit_main()
End Sub
'=====

'===== Test Lok =====
Sub lok()
    Cls
    WaitMs 50
    Cursor Off
    Locate 1,1
    Lcd "Word kod"

lok1 = ReadEE(&h0030)
lok2 = ReadEE(&h0031)
lok3 = ReadEE(&h0032)
lok4 = ReadEE(&h0033)
lk1=0
lk2=0
lk3=0

```

```
lk4=0
lk5=0
kl1=0
kl2=0
kl3=0
kl4=0
```

```
While lk1=0
    lk1=Key()
    If lk1=lok1 Then
        kl1=1
        Locate 2,1
        Lcd"*"
    Else
        If lk1<>0 And lk1<>lok1 Then
            Locate 2,1
            Lcd"*"
        End If
    End If
Wend
```

```
While lk2=0
    lk2=Key()
    If lk2=lok2 Then
        kl2=1
        Locate 2,2
        Lcd"*"
    Else
        If lk2<>0 And lk2<>lok2 Then
            Locate 2,2
            Lcd"*"
        End If
    End If
Wend
```

```
While lk3=0
    lk3=Key()
    If lk3=lok3 Then
        kl3=1
        Locate 2,3
        Lcd"*"
    Else
        If lk3<>0 And lk3<>lok3 Then
            Locate 2,3
            Lcd"*"
        End If
    End If
Wend
```

```
While lk4=0
    lk4=Key()
    If lk4=lok4 Then
```

```

        kl4=1
        Locate 2,4
        Lcd"*"
    Else
        If lk4<>0 And lk4<>lok4 Then
            Locate 2,4
            Lcd"*"
        End If
    End If
Wend

While lk5=0
    lk5=Key()
    'NoKey()
    Locate 4,1
    Lcd "*-OK"
    Locate 4,16
    Lcd "#-res"
    If lk5=57 And kl1=1 And kl2=1 And kl3=1 And kl4=1 Then
        IIInit_main()
    Else
        If lk5=59 Then
            Lok()
        Else
            If lk5<>0 And lk5<>57 And lk5<>59 Then
                lk5=0
            End If
        End If
    End If
End If
Wend

End Sub
'=====

'===== Seting new lok =====
Sub Lok_set()
    Local th As Byte
    sd=ReadEE(&h0035)
    lokk1=0
    lokk2=0
    lokk3=0
    lokk4=0
    lokk5=0
    Cls
    WaitMs 50
    Cursor On
    Locate 1,1
    Lcd "Set lok"
    Locate 2,1
    Lcd "Word new cod: "
    Locate 4, 1
    Lcd "*-Off/On"

```

```

    If sd=&ha1 Then
        Locate 3,9
        Lcd "On"
    End If
    If sd=&ha3 Then
        Locate 3,9
        Lcd "Of"
    End If
    Locate 2,15
n:
    While lokk1=0
        lokk1=0
        lokk1=Key()
    Wend

    Select Case lokk1
        Case 58      'key0
            Locate 2,15
            Lcd "0"
        Case 33      'key1
            Locate 2,15
            Lcd "1"
        Case 34      'key2
            Locate 2,15
            Lcd "2"
        Case 35      'key3
            Locate 2,15
            Lcd "3"
        Case 41      'key4
            Locate 2,15
            Lcd "4"
        Case 42      'key5
            Locate 2,15
            Lcd "5"
        Case 43      'key6
            Locate 2,15
            Lcd "6"
        Case 49      'key7
            Locate 2,15
            Lcd "7"
        Case 50      'key8
            Locate 2,15
            Lcd "8"
        Case 51      'key9
            Locate 2,15
            Lcd "9"
        Case 57      'key*
            GoTo f
        Case 59      'key#
            GoTo n
    End Select
b:

```

```

While lokk2=0
    lokk2=0
    lokk2=Key()
Wend
Select Case lokk2
    Case 58      'key0
        Locate 2,16
        Lcd "0"
    Case 33      'key1
        Locate 2,16
        Lcd "1"
    Case 34      'key2
        Locate 2,16
        Lcd "2"
    Case 35      'key3
        Locate 2,16
        Lcd "3"
    Case 41      'key4
        Locate 2,16
        Lcd "4"
    Case 42      'key5
        Locate 2,16
        Lcd "5"
    Case 43      'key6
        Locate 2,16
        Lcd "6"
    Case 49      'key7
        Locate 2,16
        Lcd "7"
    Case 50      'key8
        Locate 2,16
        Lcd "8"
    Case 51      'key9
        Locate 2,16
        Lcd "9"
    Case 57      'key*
        GoTo f
    Case 59      'key#
        GoTo b
End Select

```

k_:

```

While lokk3=0
    lokk3=0
    lokk3=Key()
Wend
Select Case lokk3
    Case 58      'key0
        Locate 2,17
        Lcd "0"
    Case 33      'key1
        Locate 2,17
        Lcd "1"

```

```

Case 34      'key2
             Locate 2,17
             Lcd "2"
Case 35      'key3
             Locate 2,17
             Lcd "3"
Case 41      'key4
             Locate 2,17
             Lcd "4"
Case 42      'key5
             Locate 2,17
             Lcd "5"
Case 43      'key6
             Locate 2,17
             Lcd "6"
Case 49      'key7
             Locate 2,17
             Lcd "7"
Case 50      'key8
             Locate 2,17
             Lcd "8"
Case 51      'key9
             Locate 2,17
             Lcd "9"
Case 57      'key*
GoTo f
Case 59      'key#
GoTo k_
End Select

```

 $j_:$

```

While lokk4=0
    lokk4=0
    lokk4=Key()
Wend
Select Case lokk4
    Case 58      'key0
        Locate 2,18
        Lcd "0"
    Case 33      'key1
        Locate 2,18
        Lcd "1"
    Case 34      'key2
        Locate 2,18
        Lcd "2"
    Case 35      'key3
        Locate 2,18
        Lcd "3"
    Case 41      'key4
        Locate 2,18
        Lcd "4"
    Case 42      'key5
        Locate 2,18

```

```

        Lcd "5"
Case 43      'key6
        Locate 2,18
        Lcd "6"
Case 49      'key7
        Locate 2,18
        Lcd "7"
Case 50      'key8
        Locate 2,18
        Lcd "8"
Case 51      'key9
        Locate 2,18
        Lcd "9"
Case 57      'key*
GoTo f
Case 59      'key#
GoTo j_
End Select

```

```

For wil=0 To 255
    Cursor Off
    Locate 4,14
    Lcd "#-OK"
    lokk5=Key()

```

```

Select Case lokk5
Case 57      'key*
f:

```

```

        If sd=&ha1 Then
            sd=&ha3
            Locate 3,9
            Lcd "Of"
            WriteEE(&h0035,sd)
        End If

```

```

        If sd=&ha3 Then
            sd=&ha1
            Locate 3,9
            Lcd "On"
            WriteEE(&h0035,sd)
        End If

```

```

Case 59      'key#
        If lokk1<>0 And lokk2<>0 And lokk3<>0 And lokk4<>0 Then
            WriteEE(&h0030,lokk1)
            WriteEE(&h0031,lokk2)
            WriteEE(&h0032,lokk3)
            WriteEE(&h0033,lokk4)
        End If
        Key_4()

```

End Select

 If wil=255 Then

 wil=0

 End If

Next

End Sub

‘=====

Sub ps_2()

‘If shi=1 Then

‘ GoTo pss

‘End If

Cursor On

bf=0

fun=0

kd=0

PcKeySend(&hf7)

‘ PcKeySend(&h12)

lcd_st=1

lcd_ms=6

st1=0

Stroc_LCD()

While kd=0

pss:

 rt=0

 ppc=0

 If keydata=&h08 Then

 Key_1()

 End If

 If keydata=&h14 Then

 cl=0

 PcKeySend(&hED)

 PcKeySend(&b00000100)

 Ps_2_shift()

 GoTo pss

 End If

 PC_read()

 Select Case ppc

 Case &h39 ‘f5

 GoTo efect

 Case &hfd ‘f6

 GoTo efect

 Case &hfe ‘f7

 GoTo efect

 Case &hff ‘f8

 GoTo efect

 Case &h31 ‘bs’’’’’’

 Bac_spas()

 GoTo pss

 Case &h32 ‘enter’’’’’’

```

        Enter()
        GoTo gt
    Case &h33 'left'''
        left_ps()
        GoTo pss
    Case &h34 'riht'''
        Right_ps()
        GoTo pss
    Case &h35 'Down
        Down()
        GoTo pss
    Case &h36 'Up'''
        Up()
        GoTo pss
    Case &h3a 'F4
        Incr lang
        Select_lang()
        GoTo pss
    Case &hf7
        ppc=&hf7
        GoTo gr
    Case &hf8
        ppc=&hf8
        GoTo gr
    Case &hf9
        ppc=&hf9
        GoTo gr
    End Select
gt:
    ppc=ppc+lang_set
gr:
    No_shift()
    Rus_no_shift()
    kdata=ppc
    If kdata<>0 Then
        Write_24lc512()
        Incr aDDR
    End If
    GoTo pss
Wend
efect:
    If fun=0 Then
        bf=0
    End If
    If fun=ppc Then
        bf=1
    End If
    If fun<>ppc Then
        bf=2
    End If
    Select Case bf
        Case 0

```

```

        fun=ppc
        f_key()
        kdata=fun
        Write_24lc512()
        Incr aDDR
    Case 1
        f_key()
        kdata=fun
        Write_24lc512()
        Incr aDDR
        fun=0
    Case 2
        f_key()
        kdata=fun
        Write_24lc512()
        Incr aDDR
        fun=ppc
        f_key()
        kdata=fun
        Write_24lc512()
        Incr aDDR
    End Select
    GoTo pss

qq:
End Sub

=====

===== Up key =====
Sub Up()
    Decr st1
k31:
    If st1=3 And lcd_ms<=18 Then '4 to 3
        Locate 4, lcd_ms
    Else
        If st1=3 And lcd_ms>=19 Then
            st1=2
            GoTo k31
        Else
            If st1=2 Then '3 to 2
                Locate 3, lcd_ms
            Else
                If st1=1 Then '2 to 1
                    Locate 2, lcd_ms
                Else
                    If st1=0 And lcd_ms<7 Then '2 to 4
                        st1=3
                        GoTo k31
                    Else
                        If st1=0 And lcd_ms>=7 Then '1 to 4
                            Locate 1, lcd_ms
                        Else
                            If st1<0 Then

```

```

                                st1=3
                                GoTo k31
                                End If
                                End If
                                End If
                                End If
                                End If
                        End If
                End If
        End Sub
        '=====

        '===== Caps Lock =====
        Sub Cap()
            keydata = 0
            While keydata = 0
                keydata = 0
                keydata=PcKey()
            Wend
        End Sub
        '=====

        '===== Down key =====
        Sub Down()
            Incr st1
k21:
            If st1=0 And lcd_ms>=7 Then '1
                Locate 1, lcd_ms
            Else
                If st1=0 And lcd_ms<7 Then
                    st1=1
                    GoTo k21
                Else
                    If st1=1 Then '2
                        Locate 2, lcd_ms
                    Else
                        If st1=2 Then '3
                            Locate 3, lcd_ms
                        Else
                            If st1=3 And lcd_ms<=18 Then '4
                                Locate 4, lcd_ms
                            Else
                                If st1=3 And lcd_ms>=19 Then
                                    st1=0
                                    GoTo k21
                                Else
                                    If st1=4 Then
                                        st1=0
                                        GoTo k21
                                    End If
                                End If
                            End If
                        End If
                    End If
                End If
            End If
        End Sub

```

```

                                End If
                                End If
                        End If
End Sub
'=====

'===== Right key =====
Sub Right_ps()
Incr aDDR
Stroc_LCD()
End Sub
'=====

'===== Left key =====
Sub Left_ps()
    Decr aDDR
    lcd_ms=lcd_ms-2
    Stroc_LCD()
    If lcd_ms=0 Then
        st1=st1-1
        lcd_ms=20
    End If
    If st1=0 And lcd_ms=6 Then
        st1=3
        lcd_ms=18
    End If
    If st1=0 Then '1
        Locate 1, lcd_ms
    Else
        If st1=1 Then '2
            Locate 2, lcd_ms
        Else
            If st1=2 Then '3
                Locate 3, lcd_ms
            Else
                If st1=3 Then '4
                    Locate 4, lcd_ms
                End If
            End If
        End If
    End If
End Sub
'=====

'===== ENTER key =====
Sub Enter()
    Incr st1
k11:
    If st1=0 Then '1
        Cls
        WaitMs 50
    End If
End Sub
'=====

```

```

Cursor On
Locate 1,1
Lcd "Text:"
Select_lang()
f_key()
lcd_ms=7
Locate 1, lcd_ms
    Else
        If st1=1 Then '2
            lcd_ms=1
            Locate 2, lcd_ms
        Else
            If st1=2 Then '3
                lcd_ms=1
                Locate 3, lcd_ms
            Else
                If st1=3 Then '4
                    lcd_ms=1
                    Locate 4, lcd_ms
                Else
                    If st1=4 Then
                        st1=0
                        GoTo k11
                    End If
                End If
            End If
        End If
    End If
ppc=&h5e
End Sub
'=====

'===== Backspace key =====
Sub Bac_spas()
    ter=0
    lcd_ms=lcd_ms-2
    Stroc_LCD()
    If lcd_ms=0 Then
        st1=st1-1
        lcd_ms=20
    End If
        If st1=0 And lcd_ms=6 Then
            st1=3
            lcd_ms=18
        End If
    If st1=0 Then '1
        Locate 1, lcd_ms
    Else
        If st1=1 Then '2
            Locate 2, lcd_ms
        Else
            If st1=2 Then '3

```

```

        Locate 3, lcd_ms
        Else
        If st1=3 Then '4
            Locate 4, lcd_ms
        End If
        End If
        End If

    End If
    Lcd " "
    Decr lcd_ms
    Stroc_LCD()
    Decr aDDR
    arDDR=aDDR
    Erase_24_bs()
End Sub
'=====

'===== Locate cyrsor =====
Sub Stroc_LCD()
    keydata=0
    keydata1=0

    Incr lcd_ms
    If st1=0 And lcd_ms=21 Then '1 to 2
        st1 = 1
        lcd_ms=1
    End If
    If st1=1 And lcd_ms=21 Then '2 to 3
        st1=2
        lcd_ms=1
    End If
    If st1=2 And lcd_ms=21 Then '3 to 4
        st1=3
        lcd_ms=1
    End If
    If st1=3 And lcd_ms=19 Then '4 to 1
        If err=1 Then
            Wait 7
        End If
        kdata=ppc
        If kdata<>0 Then
            Write_24lc512()
            Incr aDDR
        End If
        st1=0
        WaitMs 500
        Cls
        WaitMs 50
        Cursor Blink
        Locate 1,1
        Lcd "Text:"
    End If
End Sub

```

```

        Select_lang()
        lcd_ms=7
    End If
    Cursor On
If ter<>1 Then
    If st1=0 Then
        Locate 1, lcd_ms
    Else
        If st1=1 Then
            Locate 2, lcd_ms
        Else
            If st1=2 Then
                Locate 3, lcd_ms
            Else
                If st1=3 Then
                    Locate 4, lcd_ms
                End If
            End If
        End If
    End If
End If
ter=0
End Sub
'=====

'===== PS/2 keybord SHIFT =====
Sub ps_2_shift()
Local cz As Byte, fun1 As Byte
fun1=fun
cz=0
    keydata=0
    ppc=0
qe:
While cz=0
qw:
    fun=0
    Cursor On
    ppc=0
        Cap()
    fun=fun1
    If keydata=&h08 Then
        Key_1()
    End If
    If keydata=&h14 Then
        PcKeySend(&hED)
        PcKeySend(&b00000000)
        Exit Sub
        GoTo qe
    End If
    PC_read()
    Select Case ppc
        Case &h39 'f5

```

```

        GoTo efect12
    Case &hfd 'f6
        GoTo efect12
    Case &hfe 'f7
        GoTo efect12
    Case &hff 'f8
        GoTo efect12
Case &h31 'bs''''''
        Bac_spas()
        GoTo qw
Case &h32 'enter''''''
        Enter()
        GoTo gh
Case &h33 'left'''
        left_ps()
        GoTo qw
Case &h34 'riht'''
        Right_ps()
        GoTo qw
Case &h35 'Down
        Down()
        GoTo qw
Case &h36 'Up'''
        Up()
        GoTo qw
Case &h3a 'F4
        Incr lang
        Select_lang()
Case &hf7
        ppc=&hfa
        GoTo gu
Case &hf8
        ppc=&hfb
        GoTo gu
Case &hf9
        ppc=&hfc
        GoTo gu
End Select
gh:
    ppc=ppc+lang_set_sh
gu:
    Shift_on()
    Rus_shift()
    kdata=ppc
    If kdata<>0 Then
        Write_24lc512()
        Incr aDDR
    End If
    keydata=0
    GoTo qw
Wend
efect12:

```

```

If fun=0 Then
    bf=0
End If
If fun=ppc Then
    bf=1
End If
If fun<>ppc Then
    bf=2
End If
fun1=fun
Select Case bf
    Case 0
        fun=ppc
        f_key()
        kdata=fun
        Write_24lc512()
        Incr aDDR
    Case 1
        f_key()
        kdata=fun
        Write_24lc512()
        Incr aDDR
        fun=0
    Case 2
        f_key()
        kdata=fun
        Write_24lc512()
        Incr aDDR
        fun=ppc
        f_key()
        kdata=fun
        Write_24lc512()
        Incr aDDR
End Select
keydata=0
GoTo qw

kw:
End Sub
=====
Sub f_key()
    Select Case fun
        Case &h39 'f5
            Lcd Chr(&hc8)
            Stroc_LCD()
        Case &hfd 'f6
            Lcd Chr(&hc9)
            Stroc_LCD()
        Case &hfe 'f7
            Lcd Chr(&hd9)
            Stroc_LCD()
        Case &hff 'f8
            Lcd Chr(&hda)

```

```

                Stroc_LCD()
            End Select
        End Sub
        '===== Podsvetka =====
        Sub Led()
            Incr led1
k123:
            If led1=0 Then
                PORTA.7=1
            Else
                If led1=1 Then
                    PORTA.7=0
                Else
                    If led1>1 Then
                        led1=0
                        GoTo k123
                    End If
                End If
            End If
        End Sub
        '=====

        Sub Write_24lc512()

            I2Cstart &hA0      ' 24lc512 address
            I2Cwrite Msb(aDDR) ' write adr MSB
            I2Cwrite aDDR      ' write adr LSB
            I2Cwrite kdata
            I2Cstop
            WaitMs 6          ' wait to write
            adDD=adDD+1
            If adDD=127 Then
                adDD1=adDD+1
                adDD=0
            End If
            WriteEE(adEE,adDD)
            WriteEE(adEE1,adDD1)
        End Sub
        '=====

        '=====
        Sub Read_24lc512()
            Local tmp As Word, hh_ As Word
            err=1
            kdata=0
            WaitMs 10
            I2Cstart &hA0      ' 24C64 address
            I2Cwrite Msb(aDDR) ' write adr MSB
            I2Cwrite aDDR      ' write adr LSB
            For hh_=0 To 3900
                ' I2Cwrite aDDR

```

```

        ppc=I2Cread, Nack
        fun=ppc
        No_shift()
        Shift_on()
        Rus_no_shift()
        Rus_shift()
        f_key()
        ppc=0
Next
        tmp=I2Cread, Nack
        I2Cstop
        'Incr aDDR
        err=0
End Sub
'=====

Sub Erase_24()
Local ier As Word
        eer=0
        erer=0
        ier=0
For ier=0 To 30
        I2Cstart &hA0      ' 24IC512 address
        I2Cwrite Msb(aDDR)  ' write adr MSB
        I2Cwrite aDDR
        For erer=0 To 127
                eer=0
                I2Cwrite eer
                Incr aDDR
        Next
        I2Cstop
        WaitMs 6
Next
End Sub

Sub Erase_24_bs()
Local ier As Word
        eer=0
        erer=0
        ier=0
        I2Cstart &hA0      ' 24IC512 address
        I2Cwrite Msb(arDDR) ' write adr MSB
        I2Cwrite arDDR
        I2Cwrite eer
        I2Cstop
        WaitMs 6

End Sub

Sub Key_5()
Local kp5 As Byte
k5:

```

```

wil=0
Cls
WaitMs 50
Cursor Off
Locate 1,5
Lcd "Read fail"
Locate 2,7
Lcd "1-File 1"
Locate 3,7
Lcd "2-File 2"
Locate 4,7
Lcd "3-File 3"
lang=0
For wil=0 To 255
    kp5=0
    While kp5=0
        kp5=0
        kp5=Key()
    Wend
    Select Case kp5
        Case 33
            st1=0
            Cls
            WaitMs 50
            Cursor On
            Locate 1,1
            Lcd "Text: "
            kdata=0
            lcd_ms=7
            aDDR=&h0000 'write adres 24lc512
            Read_24lc512()
        Case 34
            st1=0
            Cls
            WaitMs 50
            Cursor On
            Locate 1,1
            Lcd "Text: "
            kdata=0
            lcd_ms=7
            aDDR=&h3fff 'write adres 24lc512
            Read_24lc512()
        Case 35
            st1=0
            Cls
            WaitMs 50
            Cursor On
            Locate 1,1
            Lcd "Text: "
            kdata=0
            lcd_ms=7
            aDDR=&h7fff 'write adres 24lc512

```

```

        Read_24lc512()
    Case 57
        GoTo K5
    Case 59
        Key_reset()
End Select
If wil=255 Then
    wil=0
End If
Next
End Sub

'=====
'=====
'===== PS/2 keyboard =====
Sub No_shift()
    Select Case ppc
    '
    Case &h08 'esc
    '
        Key_1()
    Case &h01 'q
        Lcd "q"
        Stroc_LCD()
    Case &h02 'w
        Lcd "w"
        Stroc_LCD()
    Case &h03 'e
        Lcd "e"
        Stroc_LCD()
    Case &h04 'r
        Lcd "r"
        Stroc_LCD()
    Case &h05 't
        Lcd "t"
        Stroc_LCD()
    Case &h06 'y
        Lcd "y"
        Stroc_LCD()
    Case &h07 'u
        Lcd "u"
        Stroc_LCD()
    Case &h08 'i
        Lcd "i"
        Stroc_LCD()
    Case &h09 'o
        Lcd "o"
        Stroc_LCD()
    Case &h0a 'p
        Lcd "p"
        Stroc_LCD()
    Case &h0b '['
        Lcd "["
        Stroc_LCD()

```

```

Case &h0c ']'
    Lcd "]"
    Stroc_LCD()
Case &h0d '\
    Lcd "\"
    Stroc_LCD()
Case &h0e 'a
    Lcd "a"
    Stroc_LCD()
Case &h0f 's
    Lcd "s"
    Stroc_LCD()
Case &h10 'd
    Lcd "d"
    Stroc_LCD()
Case &h11 'f
    Lcd "f"
    Stroc_LCD()
Case &h12 'g
    Lcd "g"
    Stroc_LCD()
Case &h13 'h
    Lcd "h"
    Stroc_LCD()
Case &h14 'j
    Lcd "j"
    Stroc_LCD()
Case &h15 'k
    Lcd "k"
    Stroc_LCD()
Case &h16 'l
    Lcd "l"
    Stroc_LCD()
Case &h17 ';'
    Lcd ";"
    Stroc_LCD()
Case &h18 '"'
    Lcd "\""
    Stroc_LCD()
Case &h19 'z
    Lcd "z"
    Stroc_LCD()
Case &h1a 'x
    Lcd "x"
    Stroc_LCD()
Case &h1b 'c
    Lcd "c"
    Stroc_LCD()
Case &h1c 'v
    Lcd "v"
    Stroc_LCD()
Case &h1d 'b

```

```

        Lcd "b"
        Stroc_LCD()
Case &h1e 'n
        Lcd "n"
        Stroc_LCD()
Case &h1f 'm
        Lcd "m"
        Stroc_LCD()
Case &h20 ',
        Lcd ","
        Stroc_LCD()
Case &h21 '.
        Lcd "."
        Stroc_LCD()
Case &h22 '/
        Lcd "/"
        Stroc_LCD()
Case &h23 '" '
        Lcd " "
        Stroc_LCD()
Case &h24 '`
        Lcd "`"
        Stroc_LCD()
Case &h25 '1
        Lcd "1"
        Stroc_LCD()
Case &h26 '2
        Lcd "2"
        Stroc_LCD()
Case &h27 '3
        Lcd "3"
        Stroc_LCD()
Case &h28 '4
        Lcd "4"
        Stroc_LCD()
Case &h29 '5
        Lcd "5"
        Stroc_LCD()
Case &h2a '6
        Lcd "6"
        Stroc_LCD()
Case &h2b '7
        Lcd "7"
        Stroc_LCD()
Case &h2c '8
        Lcd "8"
        Stroc_LCD()
Case &h2d '9
        Lcd "9"
        Stroc_LCD()
Case &h2e '0
        Lcd "0"

```

```

        Stroc_LCD()
Case &h2f ' -
    Lcd "-"
    Stroc_LCD()
Case &h30 '='
    Lcd "="
    Stroc_LCD()
Case &hf7 'F9
    Lcd Chr(&hcf)
    Stroc_LCD()
Case &hf8 'F10
    Lcd Chr(6)
    Stroc_LCD()
Case &hf9 'F11
    Lcd Chr(7)
    Stroc_LCD()
End Select
End Sub
=====
Sub Shift_on()
    Select Case ppc
    '
    '
    Case &h08 'esc
        Key_1()
    Case &h3c 'q
        Lcd "Q"
        Stroc_LCD()
    Case &h3d 'w
        Lcd "W"
        Stroc_LCD()
    Case &h3e 'e
        Lcd "E"
        Stroc_LCD()
    Case &h3f 'r
        Lcd "R"
        Stroc_LCD()
    Case &h40 't
        Lcd "T"
        Stroc_LCD()
    Case &h41 'y
        Lcd "Y"
        Stroc_LCD()
    Case &h42 'u
        Lcd "U"
        Stroc_LCD()
    Case &h43 'i''
        Lcd "I"
        Stroc_LCD()
    Case &h44 'o''
        Lcd "O"
        Stroc_LCD()
    Case &h45 'p''
        Lcd "P"

```

```

        Stroc_LCD()
Case &h46 '['
        Lcd Chr(0)
        Stroc_LCD()
Case &h47 ']'
        Lcd Chr(1)
        Stroc_LCD()
Case &h48 '\'
        Lcd "\"
        Stroc_LCD()
Case &h49 'a'
        Lcd "A"
        Stroc_LCD()
Case &h4a 's'
        Lcd "S"
        Stroc_LCD()
Case &h4b 'd'
        Lcd "D"
        Stroc_LCD()
Case &h4c 'f'
        Lcd "F"
        Stroc_LCD()
Case &h4d 'g'
        Lcd "G"
        Stroc_LCD()
Case &h4e 'h'
        Lcd "H"
        Stroc_LCD()
Case &h4f 'j'
        Lcd "J"
        Stroc_LCD()
Case &h50 'k'
        Lcd "K"
        Stroc_LCD()
Case &h51 'l'
        Lcd "L"
        Stroc_LCD()
Case &h52 ';'
        Lcd ":"
        Stroc_LCD()
Case &h53 ' '
        Lcd Chr(&h22)
        Stroc_LCD()
Case &h54 'z'
        Lcd "Z"
        Stroc_LCD()
Case &h55 'x'
        Lcd "X"
        Stroc_LCD()
Case &h56 'c'
        Lcd "C"
        Stroc_LCD()

```

```

Case &h57 'v''
    Lcd "V"
    Stroc_LCD()
Case &h58 'b''
    Lcd "B"
    Stroc_LCD()
Case &h59 'n''''
    Lcd "N"
    Stroc_LCD()
Case &h5a 'm''''
    Lcd "M"
    Stroc_LCD()
Case &h5b ',''''
    Lcd "<"
    Stroc_LCD()
Case &h5c '.''''
    Lcd ">"
    Stroc_LCD()
Case &h5d '/''''
    Lcd "?"
    Stroc_LCD()
Case &h5e '" ""''''
    Lcd " "
    Stroc_LCD()
Case &h5f '^''''
    Lcd "~"
    Stroc_LCD()
Case &h60 '1''''
    Lcd Chr(&h21)
    Stroc_LCD()
Case &h61 '2''''
    Lcd "@"
    Stroc_LCD()
Case &h62 '3''''''
    Lcd "#"
    Stroc_LCD()
Case &h63 '4''''''
    Lcd "$"
    Stroc_LCD()
Case &h64 '5''''
    Lcd "%"
    Stroc_LCD()
Case &h65 '6''''''
    Lcd "^"
    Stroc_LCD()
Case &h66 '7''''
    Lcd "&"
    Stroc_LCD()
Case &h67 '8''''
    Lcd "*"
    Stroc_LCD()
Case &h68 '9''''

```

```

        Lcd "("
        Stroc_LCD()
Case &h69 '0''''''
        Lcd ")"
        Stroc_LCD()
Case &h6a '-''''''
        Lcd "_"
        Stroc_LCD()
Case &h6b '='''''''
        Lcd "+"
        Stroc_LCD()
Case &hfa 'F9
        ppc=&hf7
        No_shift()
Case &hfb 'F10
        ppc=&hf8
        No_shift()
Case &hfc 'F11
        ppc=&hf9
        No_shift()
End Select
End Sub
'=====
'===== PS/2 keybord =====
Sub Rus_no_shift()
gf:
        Select Case ppc
        '
        Case &h08 'esc
        '
                Key_1()
        Case &h77 'й
                Lcd Chr(&hb9)
                Stroc_LCD()
        Case &h78 'и
                Lcd Chr(&he5)
                Stroc_LCD()
        Case &h79 'y
                ppc=&h06
                No_shift()
        Case &h7a 'к
                Lcd Chr(&hba)
                Stroc_LCD()
        Case &h7b 'е
                ppc=&h03
                No_shift()
        Case &h7c 'н
                Lcd Chr(&hbd)
                Stroc_LCD()
        Case &h7d 'р
                Lcd Chr(&hb4)
                Stroc_LCD()
        Case &h7e 'м
                Lcd Chr(&hc1)

```

```

        Stroc_LCD()
Case &h7f 'и
    Lcd Chr(&he6)
    Stroc_LCD()
Case &h80 'з
    Lcd Chr(&hb7)
    Stroc_LCD()
Case &h81 'х
    ppc=&h1a
    No_shift()
Case &h82 'ъ
    If lang=2 Then
        ppc=&hed
        GoTo gf
    Else
        Lcd Chr(&hc2)
        Stroc_LCD()
    End If
Case &h83 '\
    ppc=&h0d
    No_shift()
Case &h84 'ф
    Lcd Chr(&he4)
    Stroc_LCD()
Case &h85 'ы
    If lang=2 Then
        ppc=&h08
        No_shift()
    Else
        Lcd Chr(&hc3)
        Stroc_LCD()
    End If
Case &h86 'в
    Lcd Chr(&hb3)
    Stroc_LCD()
Case &h87 'а
    ppc=&h0e
    No_shift()
Case &h88 'п
    Lcd Chr(&hbe)
    Stroc_LCD()
Case &h89 'р
    ppc=&h0a
    No_shift()
Case &h8a 'о
    ppc=&h09
    No_shift()
Case &h8b 'л
    Lcd Chr(&hbb)
    Stroc_LCD()
Case &h8c 'д
    Lcd Chr(&he3)

```

```

        Stroc_LCD()
Case &h8d 'ж
    Lcd Chr(&hb6)
    Stroc_LCD()
Case &h8e 'э
    If lang=2 Then
        ppc=&hee
        GoTo gf
    Else
        Lcd Chr(&hc5)
        Stroc_LCD()

    End If
Case &h8f 'я
    Lcd Chr(&hc7)
    Stroc_LCD()
Case &h90 'ч
    Lcd Chr(&hc0)
    Stroc_LCD()
Case &h91 'с
    ppc=&h1b
    No_shift()
Case &h92 'м
    Lcd Chr(&hbc)
    Stroc_LCD()
Case &h93 'и
    Lcd Chr(&hb8)
    Stroc_LCD()
Case &h94 'т
    Lcd Chr(&hbf)
    Stroc_LCD()
Case &h95 'ь
    Lcd Chr(&hc4)
    Stroc_LCD()
Case &h96 'ф
    Lcd Chr(&hb2)
    Stroc_LCD()
Case &h97 'ю
    Lcd Chr(&hc6)
    Stroc_LCD()
Case &h98 '
    ppc=&h21
    No_shift()
Case &h99 '" '
    ppc=&h23
    No_shift()
Case &h9a 'ë
    Lcd Chr(&hb5)
    Stroc_LCD()
Case &h9b '1
    ppc=&h25
    No_shift()
Case &h9c '2

```

```

        ppc=&h26
        No_shift()
Case &h9d '3
        ppc=&h27
        No_shift()
Case &h9e '4
        ppc=&h28
        No_shift()
Case &h9f '5
        ppc=&h29
        No_shift()
Case &ha0 '6
        ppc=&h2a
        No_shift()
Case &ha1 '7
        ppc=&h2b
        No_shift()
Case &ha2 '8
        ppc=&h2c
        No_shift()
Case &ha3 '9
        ppc=&h2d
        No_shift()
Case &ha4 '0
        ppc=&h2e
        No_shift()
Case &ha5 '-'
        ppc=&h2f
        No_shift()
Case &ha6 '='
        ppc=&h30
        No_shift()
Case &hed 'i
        Lcd Chr(5)
        Stroc_LCD()
Case &hee 'e
        Lcd Chr(3)
        Stroc_LCD()
End Select
End Sub
=====
Sub Rus_shift()
re:
        Select Case ppc
        '
        Case &h08 'esc
        '
                Key_1()
        Case &hb2 'Й
                Lcd Chr(&ha6)
                Stroc_LCD()
        Case &hb3 'И
                Lcd Chr(&he1)
                Stroc_LCD()

```

```

Case &hb4 'Y
    Lcd Chr(&ha9)
    Stroc_LCD()
Case &hb5 'K
    ppc=&h50
    Shift_on()
Case &hb6 'E
    ppc=&h3e
    Shift_on()
Case &hb7 'H
    ppc=&h4e
    Shift_on()
Case &hb8 'I
    Lcd Chr(&ha1)
    Stroc_LCD()
Case &hb9 'III
    Lcd Chr(&hac)
    Stroc_LCD()
Case &hba 'II
    Lcd Chr(&he2)
    Stroc_LCD()
Case &hbb '3
    Lcd Chr(&ha4)
    Stroc_LCD()
Case &hbc 'X
    ppc=&h55
    Shift_on()
Case &hbd 'b
    If lang=2 Then
        ppc=&hf0
        GoTo re
    Else
        Lcd Chr(&had)
        Stroc_LCD()
    End If
Case &hbe '\''''''''
    ppc=&h48
    Shift_on()
Case &hbf 'Φ
    Lcd Chr(&haa)
    Stroc_LCD()
Case &hc0 'I
    If lang=2 Then
        ppc=&h43
        Shift_on()
    Else
        Lcd Chr(&hae)
        Stroc_LCD()
    End If
Case &hc1 'B
    ppc=&h58
    Shift_on()

```

```

Case &hc2 'A
    ppc=&h49
    Shift_on()
Case &hc3 'П
    Lcd Chr(&ha8)
    Stroc_LCD()
Case &hc4 'P
    ppc=&h45
    Shift_on()
Case &hc5 'O
    ppc=&h44
    Shift_on()
Case &hc6 'И
    Lcd Chr(&ha7)
    Stroc_LCD()
Case &hc7 'Д
    Lcd Chr(&he0)
    Stroc_LCD()
Case &hc8 'Ж
    Lcd Chr(&ha3)
    Stroc_LCD()
Case &hc9 'Э
    If lang=2 Then
        ppc=&hf1
        GoTo re
    Else
        Lcd Chr(&haf)
        Stroc_LCD()
    End If
Case &hca 'Я
    Lcd Chr(&hb1)
    Stroc_LCD()
Case &hcb 'Ч
    Lcd Chr(&hab)
    Stroc_LCD()
Case &hcc 'C
    ppc=&h56
    Shift_on()
Case &hcd 'M
    ppc=&h5a
    Shift_on()
Case &hce 'И
    Lcd Chr(&ha5)
    Stroc_LCD()
Case &hcf 'T
    ppc=&h40
    Shift_on()
Case &hd0 'b
    Lcd Chr(&h62)
    Stroc_LCD()
Case &hd1 'B
    Lcd Chr(&ha0)

```

```

        Stroc_LCD()
Case &hd2 'IO
        Lcd Chr(&hb0)
        Stroc_LCD()
Case &hd3 '.'
        Lcd ","
        Stroc_LCD()
Case &hd4 "" """"
        ppc=&h5e
        Shift_on()
Case &hd5 'Ë
        Lcd Chr(&ha2)
        Stroc_LCD()
Case &hd6 '1'''
        ppc=&h60
        Shift_on()
Case &hd7 '2'''
        ppc=&h53
        Shift_on()
Case &hd8 '3''''
        Lcd Chr(&hcc)
        Stroc_LCD()
Case &hd9 '4''''
        Lcd Chr(&h3b)
        Stroc_LCD()
Case &hda '5''''
        ppc=&h64
        Shift_on()
Case &hdb '6''''
        Lcd ":"
        Stroc_LCD()
Case &hdc '7''''
        ppc=&h5d
        Shift_on()
Case &hdd '8''''
        ppc=&h67
        Shift_on()
Case &hde '9''''
        ppc=&h68
        Shift_on()
Case &hdf '0''''
        ppc=&h69
        Shift_on()
Case &he0 '-''''
        ppc=&h6a
        Shift_on()
Case &he1 '= ''''''''
        ppc=&h6b
        Shift_on()
Case &hf0 'İ
        Lcd Chr(4)
        Stroc_LCD()

```

```

        Case &hf1 '€
            Lcd Chr(2)
            Stroc_LCD()
        End Select
    End Sub

```

```

'=====

```

```

Sub Select_lang()
sl:
    Select Case lang
    Case 0
        lang_set=0
        lang_set_sh=&h3b
        Locate 4,19
        Lcd "En"
        Decr lcd_ms
        Stroc_LCD()
    Case 1
        lang_set=&h76
        lang_set_sh=&hb1
        Locate 4,19
        Lcd "Ru"
        Decr lcd_ms
        Stroc_LCD()
    Case 2
        lang_set=&h76
        lang_set_sh=&hb1
        Locate 4,19
        Lcd "Ua"
        Decr lcd_ms
        Stroc_LCD()
    Case 3
        lang=0
        GoTo sl
    End Select
    '    incr lang
End Sub

```

```

'=====

```

```

Sub PC_read()
Select Case keydata
'    Case &h08 'esc
'        Key_1()
    Case &h15 'q
        ppc=&h01
    Case &h1d 'w
        ppc=&h02
    Case &h24 'e
        ppc=&h03
    Case &h2d 'r

```

```

        ppc=&h04
Case &h2c 't
        ppc=&h05
Case &h35 'y
        ppc=&h06
Case &h3c 'u
        ppc=&h07
Case &h43 'i
        ppc=&h08
Case &h44 'o
        ppc=&h09
Case &h4d 'p
        ppc=&h0a
Case &h54 '['
        ppc=&h0b
Case &h5b ']'
        ppc=&h0c
Case &h5c '\'
        ppc=&h0d
Case &h1c 'a
        ppc=&h0e
Case &h1b 's
        ppc=&h0f
Case &h23 'd
        ppc=&h10
Case &h2b 'f
        ppc=&h11
Case &h34 'g
        ppc=&h12
Case &h33 'h
        ppc=&h13
Case &h3b 'j
        ppc=&h14
Case &h42 'k
        ppc=&h15
Case &h4b 'l
        ppc=&h16
Case &h4c ';
        ppc=&h17
Case &h52 '"
        ppc=&h18
Case &h1a 'z
        ppc=&h19
Case &h22 'x
        ppc=&h1a
Case &h21 'c
        ppc=&h1b
Case &h2a 'v
        ppc=&h1c
Case &h32 'b
        ppc=&h1d
Case &h31 'n

```

```

        ppc=&h1e
Case &h3a 'm
        ppc=&h1f
Case &h41 ',
        ppc=&h20
Case &h49 '.
        ppc=&h21
Case &h4a '/
        ppc=&h22
Case &h29 '" "
        ppc=&h23
Case &h0e ``
        ppc=&h24
Case &h16 '1
        ppc=&h25
Case &h1e '2
        ppc=&h26
Case &h26 '3
        ppc=&h27
Case &h25 '4
        ppc=&h28
Case &h2e '5
        ppc=&h29
Case &h36 '6
        ppc=&h2a
Case &h3d '7
        ppc=&h2b
Case &h3e '8
        ppc=&h2c
Case &h46 '9
        ppc=&h2d
Case &h45 '0
        ppc=&h2e
Case &h4e '-'
        ppc=&h2f
Case &h55 '='
        ppc=&h30
Case &h66 'bs
        ppc=&h31
Case &h5a 'enter
        ppc=&h32
Case &h61 'left
        ppc=&h33
Case &h6a 'riht
        ppc=&h34
Case &h60 'Down
        ppc=&h35
Case &h63 'Up
        ppc=&h36
Case &h1f 'F4
        ppc=&h3a
Case &h27 'f5

```

```
        ppc=&h39
Case &h2f 'f6
        ppc=&hfd
Case &h37 'f7
        ppc=&hfe
Case &h3f 'f8
        ppc=&hff
Case &h47 'F9
        ppc=&hf7
Case &h5e 'F10
        ppc=&hf8
Case &h56 'F11
        ppc=&hf9
End Select
```

```
End Sub
```

Додаток В

Лістинг програми мікроконтроллера DD5

```

'//////////
'Program to RunLine module
'This program is the propert of
'Danilov Igor Gennady, developed in an
'assistance with pridpriyatiem "Elensi"
'and registered under the trade mark of
'LedArtis tm.
'//////////
$Device= m32          ' used device
$Stack = 120          ' stack depth
$Clock = 12            ' adjust for used crystal
$I2C SDA=PORTC.4, SCL=PORTC.5
$1Wire=PORTD.3
$Baud = 9600
$Def Clock=PORTC.1
$Def LE=PORTC.0

Declare Sub ReadPage(Padr As Word)
Declare Sub WritePage(Padr As Word)
Declare Sub FillBuf()
Declare Sub OUTB(d As Byte)
Declare Sub WRITEB()
Declare Sub WRITE()
Declare Sub Nullleft()
Declare Sub GetTime()
Declare Sub GetTemp()
Declare Sub OutDate()
Declare Sub OutTime()
Declare Sub OutTemp()
Declare Sub RTC_Init()
Declare Interrupt Urxc()

Const DS1307=&hd0
'Const Ctrl=&hA3      '\
Const sE=&hA3         'E end
'Const sP=&h3E         'P pause
Const sD=&hf9         'D date
Const sT=&hf8         'T time
Const sG=&hf7         'G temperature
Const sPlus=&hF5      '+'
Const sMin=&hF4        '-'
Const sBeg=&hAE      '~ begin transmit
Const sBD=&hA1        '{ begit date/time setup
Const sED=&h75        '} end date/time setup

Dim MODULES As Byte, i As Byte, k As Byte, j As Byte,ccc As Byte

```

```

Dim ADDR As Word, DAT As Byte, DAT1 As Byte, Udata As Byte
Dim BUFF(32) As Byte
Dim BTime(15) As Byte
Dim STROKE As Byte, EFF As Byte
Dim BUFZ(14) As Byte, count As Byte
Dim m As Byte, n As Byte
Dim Table1(3374) As Flash Byte
Dim day As Byte, month As Byte, year As Byte, dof As Byte
Dim hour As Byte, min As Byte, secnd As Byte, Minus As Byte
Dim ihour As Byte, imin As Byte, isec As Byte, iday As Byte, imonth As Byte, iyear As
Byte, idof As Byte
Dim FlagCo As Byte, FlagEnd As Byte, tmp As Byte, tmp1 As Byte, cnt As Byte, cnt1 As
Byte

Dim temp As Byte, cunt As Byte, TR As Byte, TN As Byte
Dim T As Word
Dim gg(8) As Byte, o As Byte, r As Byte

PORTA=0
PORTB=0
PORTC=0
DDRB=&hFF
DDRC=&hFF
DDRA=&hFF

FlagCo=0:FlagEnd=0
gg(0)=&hbf
gg(1)=&h93
gg(2)=&h7d
gg(3)=&h23
gg(4)=&h7C
gg(5)=&h0E
gg(6)=&h92
gg(7)=&hA3
o=0
'--- Init buffer -----
For i=0 To 31
    BUFF(i)=0
Next
For i=0 To 14
    BTime(i)=0
Next
'--- end init buffer -----
RTC_Init()
'----- Init stroke ----
PORTA=0
For i=1 To 32
    Set Clock
    WaitUs 30
    Reset Clock
Next
Set LE
Reset LE

```

‘----- End Init stroke -----

ADDR=0

Enable Interrupts

Enable Urxc

‘-----

MainLoop:

Do

 ReadPage(ADDR)

 If DAT=sE Then ‘E

 GoTo ENDS ‘End of Stroke

 End If

 If DAT=sD Then ‘D

 GoTo DATE ‘Date

 End If

 If DAT=sT Then ‘T

 GoTo TIME ‘Time

 End If

 If DAT=sG Then ‘G

 GoTo TEMPER ‘Temperature

 End If

 OUTB(DAT) ‘14 bytes in buffer

 WRITEB() ‘write to DRV

 GoTo CONTINUE

ENDS:

 Nullleft()

 ADDR=0

 ‘o=0

 GoTo MainLoop

PAUSE:

 Wait 1

 GoTo CONTINUE

DATE:

 Nullleft()

 GetTime()

 OutDate()

 Wait 5

 PORTA=0

 For i=1 To 31

 Set Clock

 WaitUs 30

 Reset Clock

 Next

 Set LE

 WaitUs 40

 Reset LE

 GoTo CONTINUE

TIME:

 Nullleft()

```

    GetTime()
    OutTime()
    Wait 5
    PORTA=0
    For i=1 To 31
        Set Clock
        WaitUs 30
        Reset Clock
    Next
    Set LE
    WaitUs 40
    Reset LE
    GoTo CONTINUE
TEMPER:
    Nullleft()
    GetTemp()
    OutTemp()
    Wait 5
    PORTA=0
    For i=1 To 31
        Set Clock
        WaitUs 30
        Reset Clock
    Next
    Set LE
    WaitUs 40
    Reset LE
    GoTo CONTINUE
ARRUP:
    OUTB(&h8E)
    WRITEB()
    GoTo CONTINUE
ARRDN:
    OUTB(&h8F)
    WRITEB()
    GoTo CONTINUE
CONTINUE:
    ADDR=ADDR+1

Loop
'///// Fill Buffer ///
Sub FillBuf()
Do
    ReadPage(ADDR)
    BUFF(ADDR)=DAT
    ADDR=ADDR+1
Loop While DAT <> &hAA
    ADDR=0
End Sub
'//////// Read byte from 24C64 ///
Sub ReadPage(Padr As Word)
Local tmp As Byte

```

```

I2Cstart &ha0      ' 24C64 address
I2Cwrite Msb(Padr) ' write adr MSB
I2Cwrite Padr      ' write adr LSB
    DAT=I2Cread, Ack
tmp=I2Cread, Nack
I2Cstop
End Sub

'//////// Write byte to 24C64 //////////
Sub WritePage(Padr As Word)
Disable Interrupts
I2Cstart &ha0      ' 24C64 address
I2Cwrite Msb(Padr) ' write adr MSB
I2Cwrite Padr      ' write adr LSB
    I2Cwrite Udata
I2Cstop
WaitMs 6
End Sub

'//////// Fill Output Znak Buffer
Sub OUTB(d As Byte)
Local ii As Byte
    For ii=0 To 13
        BUFZ(ii)=Table1(d*14+ii)
    Next
End Sub

'////////Sdvig 0 do konca stroki /
Sub Nullleft()
    For k=0 To 31
        For i=0 To 30
            m=31-i:n=m:Decr n
            BUFF(m)=BUFF(n)
        Next
        PORTA=0
        For i=0 To 31
            Set Clock
            Nop
            Reset Clock
        Next
        Set LE
        Nop
        Reset LE
        WaitMs 2
        BUFF(0)=0
        For i=0 To 31
            m=31-i
            PORTA=BUFF(m)
            Set Clock
            Nop
            Reset Clock
        Next
        Set LE
    Next

```

```

        Nop
        Reset LE
        WaitMs 17
    Next
End Sub
'///// Out Buffer to DRV ///
Sub WRITEB()
Local bd As Byte
    count=BUFZ(0)
    For j=1 To count
        For i=0 To 30
            BUFF(31-i)=BUFF(30-i)
        Next
        BUFF(0)=BUFZ(j)
        PORTA=0
        For i=0 To 31
            Set Clock
            Nop
            Reset Clock
        Next
        Set LE
        Nop
        Reset LE

        For i=0 To 31
            PORTA=BUFF(31-i)
            Set Clock
            Nop
            Reset Clock
        Next
        WaitMs 2
        Set LE
        Nop
        Reset LE
        WaitMs 17
    Next
End Sub
'/////////////////
Sub WRITE()
Local bb As Byte
    count=BUFZ(0)
    For j=1 To count
        bb=BUFZ(j)
        PORTA=bb
        Nop
        Nop
        Set Clock
        WaitUs 30
        Reset Clock
        ' waitms 100
    Next
End Sub

```

```

'//////////
Sub RTC_Init()
Disable Interrupts
Nop
Nop
    I2Cstart DS1307
    I2Cwrite 0
    isec=I2Cread,Nack
    I2Cstop

    isec=isec And &h7F
    I2Cstart DS1307
    I2Cwrite 0
    I2Cwrite isec
    I2Cstop

    I2Cstart DS1307
    I2Cwrite 7
    I2Cwrite 144'147
    I2Cstop
Enable Interrupts
End Sub
'///// Get Date-Time ////
Sub GetTime()
Disable Interrupts
Nop
Nop
    I2Cstart DS1307
    I2Cwrite 0
    secnd=I2Cread, Ack
    min=I2Cread, Ack
    hour=I2Cread, Ack
    dof=I2Cread, Ack
    day=I2Cread, Ack
    month=I2Cread, Ack
    year=I2Cread, Nack
    I2Cstop
Enable Interrupts
End Sub
'///// Get temperature ////
Sub GetTemp()
Disable Interrupts
Nop
Nop
    1Wreset
    1Wwrite &hcc; &h44
    Wait 1
    1Wreset
    1Wwrite &hcc; &hbbe
    1Wread T,9
    Minus=0 ' Plus temperature

```

```

If T>2000 Then
    T=(65535-T)+1
    Minus=1 ' Minus temperature
End If
    T=T/16          'real temperature
    TN=T
Enable Interrupts
End Sub
'///// Out Date to stroke /////
Sub OutDate()
    tmp=day And &hF0
    Swap(tmp)
    If tmp=&h00 Then
        tmp=&h2E
    Else
        tmp=tmp+&h24
    End If
    OUTB(tmp)
    WRITE()
    tmp=day And &h0F
    If tmp=&h00 Then
        tmp=&h2E
    Else
        tmp=tmp+&h24
    End If
    OUTB(tmp)
    WRITE()

    tmp=&h21 ' .
    OUTB(tmp)
    WRITE()
    tmp=month And &hF0
    Swap(tmp)
    If tmp=&h00 Then
        tmp=&h2E
    Else
        tmp=tmp+&h24
    End If
    OUTB(tmp)
    WRITE()
    tmp=month And &h0F
    If tmp=&h00 Then
        tmp=&h2E
    Else
        tmp=tmp+&h24
    End If
    OUTB(tmp)
    WRITE()
    PORTA=0
    For cnt=0 To 1      '*To center
        Set Clock
        WaitUs 30
    
```

```

        Reset Clock
    Next
    Set LE
    WaitUs 40
    Reset LE
End Sub
'///// Out Time to stroke /////
Sub OutTime()
    tmp1=hour And &hF0
    Swap(tmp1)
    If tmp1=&h00 Then
        tmp1=&h2E
    Else
        tmp1=tmp1+&h24
    End If
    OUTB(tmp1)
    WRITE()
    tmp1=hour And &h0F
    If tmp1=&h00 Then
        tmp1=&h2E
    Else
        tmp1=tmp1+&h24
    End If
    OUTB(tmp1)
    WRITE()

    tmp1=&h52 ' :
    OUTB(tmp1)
    WRITE()
    tmp1=min And &hF0
    Swap(tmp1)
    If tmp1=&h00 Then
        tmp1=&h2E
    Else
        tmp1=tmp1+&h24
    End If
    OUTB(tmp1)
    WRITE()
    tmp1=min And &h0F
    If tmp1=&h00 Then
        tmp1=&h2E
    Else
        tmp1=tmp1+&h24
    End If
    OUTB(tmp1)
    WRITE()
    PORTA=0
    For cnt1=0 To 1 '*To center
        Set Clock
        WaitUs 30
        Reset Clock
    Next

```

```

        Set LE
        WaitUs 40
        Reset LE
End Sub
'/////Out temperature to stroke
Sub OutTemp()
Local f As Byte, g As Byte
    TR=Bcd(TN)
If TN<>0 Then
    If Minus=0 Then
        temp=&h6B
    Else
        temp=&h2f
    End If
    OUTB(temp)
    WRITE()
End If
    PORTA=0
    Set Clock
    WaitUs 30
    Reset Clock
If TN<>0 Then
    temp=TR And &hF0
    Swap(temp)
    If temp=&h00 Then
        g=&h2E
    Else
        g=temp+&h24
    End If
    OUTB(g)
    WRITE()
    temp=TR And &h0F
    If temp=&h00 Then
        f=&h2E
    Else
        f=temp +&h24
    End If
    OUTB(f)
    WRITE()
End If
    temp=&h34 ' Spase min
    OUTB(temp)
    WRITE()
    temp=&h56 ' C
    OUTB(temp)
    WRITE()
    PORTA=0
    For cunt=0 To 1 ' *To center
        Set Clock
        WaitUs 30
        Reset Clock
    Next

```

```

        Set LE
        WaitUs 40
        Reset LE
    End Sub
    '///// Interrupt by UART Receive ///
    Interrupt Urxc(), Save All '
    Disable Interrupts

        InputBin Udata
        Select Case Udata
            Case sBeg
                GoTo Poehali
            Case sBD
                GoTo SetTime
            Case Else
                GoTo GoHome
        End Select
    '-----
    Poehali:
        ADDR=0
        Udata=0

    InputLoop:
        InputBin Udata
        If Udata=sE Then
            FlagEnd=1
        Else
            FlagEnd=0
        End If
        GoTo WR

    WR:
        WritePage(ADDR)
        If FlagEnd=1 Then
            FlagCo=0:FlagEnd=0
            ADDR=0
            PORTA=0
            For k=0 To 31
                Set Clock
                WaitUs 30
                Reset Clock
            Next
            Set LE
            WaitUs 40
            Reset LE
            GoTo GoHome
        Else
            ADDR=ADDR+1
            GoTo InputLoop
        End If
        GoTo GoHome
    '-----
    SetTime:

```

```

For ccc=0 To 13
    InputBin Udata
    BTime(ccc)=Udata
Next
day=BTime(7)
ccc=Swap(day)
ccc=ccc And &hF0
day=BTime(8) Or ccc

month=BTime(9)
ccc=Swap(month)
ccc=ccc And &hF0
month=BTime(10) Or ccc

year=BTime(11)
ccc=Swap(year)
ccc=ccc And &hF0
year=BTime(12) Or ccc

dof=BTime(6)

hour=BTime(4)
ccc=Swap(hour)
ccc=ccc And &hF0
hour=BTime(5) Or ccc

min=BTime(2)
ccc=Swap(min)
ccc=ccc And &hF0
min=BTime(3) Or ccc

secnd=BTime(0)
ccc=Swap(secnd)
ccc=ccc And &hF0
secnd=BTime(1) Or ccc

I2Cstart DS1307
I2Cwrite 0
I2Cwrite secnd
I2Cwrite min
I2Cwrite hour
I2Cwrite dof
I2Cwrite day
I2Cwrite month
I2Cwrite year
I2Cstop
GoTo GoHome

```

GoHome:

```

    Enable Interrupts
    Enable Urxc

```

End Interrupt

Table1= &h00,&h00,&h00,&h00,&h00,&h00,&h00,	
&h00,&h00,&h00,&h00,&h00,&h00,&h00, ‘0	00
&h07,&h1E,&h3F,&h21,&h23,&h3F,&h1F,	
&h00,&h00,&h00,&h00,&h00,&h00,&h00, ‘q	01
&h0A,&h38,&h3F,&h07,&h1E,&h38,&h1E,	
&h07,&h3F,&h38,&h00,&h00,&h00,&h00,	‘w 02
&h07,&h1E,&h3F,&h29,&h29,&h3B,&h1A,	
&h00,&h00,&h00,&h00,&h00,&h00,&h00, ‘e	03
&h07,&h3F,&h3F,&h10,&h20,&h30,&h10,	
&h00,&h00,&h00,&h00,&h00,&h00,&h00, ‘r	04
&h05,&h20,&h3F,&h3F,&h20,&h00,&h00,	
&h00,&h00,&h00,&h00,&h00,&h00,&h00, ‘t	05
&h08,&h21,&h39,&h1F,&h06,&h1C,&h38,	
&h20,&h00,&h00,&h00,&h00,&h00,&h00, ‘y	06
&h07,&h3E,&h3F,&h01,&h01,&h3F,&h3E,	
&h00,&h00,&h00,&h00,&h00,&h00,&h00, ‘u	07
&h05,&h00,&hBF,&hBF,&h00,&h00,&h00,	
&h00,&h00,&h00,&h00,&h00,&h00,&h00, ‘i	08
&h07,&h1E,&h3F,&h21,&h21,&h3F,&h1E,	
&h00,&h00,&h00,&h00,&h00,&h00,&h00, ‘o	09
&h07,&h3F,&h3F,&h24,&h24,&h3C,&h18,	
&h00,&h00,&h00,&h00,&h00,&h00,&h00, ‘p	0A
&h03,&hFF,&h81,&h00,&h00,&h00,&h00,	
&h00,&h00,&h00,&h00,&h00,&h00,&h00, ‘[	0B
&h03,&h81,&hFF,&h00,&h00,&h00,&h00,	
&h00,&h00,&h00,&h00,&h00,&h00,&h00, ‘]	0C
&h06,&h01,&h06,&h18,&h60,&h80,&h00,	
&h00,&h00,&h00,&h00,&h00,&h00,&h00, ‘\	0D
&h07,&h06,&h2F,&h29,&h29,&h3F,&h1F,	
&h00,&h00,&h00,&h00,&h00,&h00,&h00, ‘a	0E
&h07,&h1A,&h3B,&h2D,&h2D,&h37,&h16,	
&h00,&h00,&h00,&h00,&h00,&h00,&h00, ‘s	0F
&h07,&h0E,&h1F,&h11,&h11,&hFF,&hFF,	
&h00,&h00,&h00,&h00,&h00,&h00,&h00, ‘d	10
&h07,&h7F,&hFF,&h90,&h90,&hC0,&h40,	
&h00,&h00,&h00,&h00,&h00,&h00,&h00, ‘f	11
&h07,&h18,&h3D,&h25,&h25,&h3F,&h1E,	
&h00,&h00,&h00,&h00,&h00,&h00,&h00, ‘g	12
&h07,&h3F,&h3F,&h08,&h08,&h3F,&h3F,	
&h00,&h00,&h00,&h00,&h00,&h00,&h00, ‘h	13
&h06,&h02,&h03,&h01,&h3F,&h3E,&h00,	
&h00,&h00,&h00,&h00,&h00,&h00,&h00, ‘j	14
&h07,&h3F,&h3F,&h08,&h1C,&h37,&h23,	
&h00,&h00,&h00,&h00,&h00,&h00,&h00, ‘k	15
&h06,&h3F,&h3F,&h01,&h01,&h01,&h00,	
&h00,&h00,&h00,&h00,&h00,&h00,&h00, ‘l	16
&h03,&h4C,&hCC,&h00,&h00,&h00,&h00,	
&h00,&h00,&h00,&h00,&h00,&h00,&h00, ‘;	17
&h05,&h00,&hF0,&h00,&hF0,&h00,&h00,	

&h08,&h70,&hF8,&h88,&h88,&h88,&hF8, &h70,&h00,&h00,&h00,&h00,&h00,&h00, ‘degree	‘	32
&h07,&h7E,&hFF,&h81,&h83,&hFF,&h7F, &h00,&h00,&h00,&h00,&h00,&h00,&h00, ‘Q	3C	
&h0C,&hC0,&hFC,&h3F,&h07,&h7C,&hE0, &h7C,&h07,&h3F,&hFC,&hC0,&h00,&h00, &h07,&hFF,&hFF,&h91,&h91,&h91,&h81, &h00,&h00,&h00,&h00,&h00,&h00,&h00, ‘E	‘W	3D
&h08,&hFF,&hFF,&h88,&h8C,&hFE,&h73, &h01,&h00,&h00,&h00,&h00,&h00,&h00, ‘R	3E	
	3F	
&h07,&h80,&h80,&hFF,&hFF,&h80,&h80, &h00,&h00,&h00,&h00,&h00,&h00,&h00, ‘T	40	
&h07,&hC0,&hF1,&h3F,&h1E,&hF8,&hE0, &h00,&h00,&h00,&h00,&h00,&h00,&h00, ‘Y	41	
&h07,&hFE,&hFF,&h01,&h01,&hFF,&hFE, &h00,&h00,&h00,&h00,&h00,&h00,&h00, ‘U	42	
&h05,&h00,&hFF,&hFF,&h00,&h00,&h00, &h00,&h00,&h00,&h00,&h00,&h00,&h00, ‘I	43	
&h07,&h7E,&hFF,&h81,&h81,&hFF,&h7E, &h00,&h00,&h00,&h00,&h00,&h00,&h00, ‘O	44	
&h07,&hFF,&hFF,&h84,&h84,&hFC,&h78, &h00,&h00,&h00,&h00,&h00,&h00,&h00, ‘P	45	
&h04,&h18,&h66,&h81,&h00,&h00,&h00, &h00,&h00,&h00,&h00,&h00,&h00,&h00, ‘{	46	
&h04,&h81,&h66,&h18,&h00,&h00,&h00, &h00,&h00,&h00,&h00,&h00,&h00,&h00, ‘}	47	
&h03,&hFF,&hFF,&h00,&h00,&h00,&h00, &h00,&h00,&h00,&h00,&h00,&h00,&h00, ‘	48	
&h07,&h3F,&h7F,&hC4,&hC4,&h7F,&h3F, &h00,&h00,&h00,&h00,&h00,&h00,&h00, ‘A	49	
&h07,&h62,&hF3,&h99,&h89,&hCF,&h46, &h00,&h00,&h00,&h00,&h00,&h00,&h00, ‘S	4A	
&h07,&hFF,&hFF,&h81,&h81,&hFF,&h7E, &h00,&h00,&h00,&h00,&h00,&h00,&h00, ‘D	4B	
&h07,&hFF,&hFF,&h90,&h90,&h90,&h80, &h00,&h00,&h00,&h00,&h00,&h00,&h00, ‘F	4C	
&h07,&h7E,&hFF,&h81,&h89,&hCF,&h4E, &h00,&h00,&h00,&h00,&h00,&h00,&h00, ‘G	4D	
&h07,&hFF,&hFF,&h10,&h10,&hFF,&hFF, &h00,&h00,&h00,&h00,&h00,&h00,&h00, ‘H	4E	
&h06,&h02,&h03,&h01,&hFF,&hFE,&h00, &h00,&h00,&h00,&h00,&h00,&h00,&h00, ‘J	4F	
&h07,&hFF,&hFF,&h10,&h7E,&hCF,&h81, &h00,&h00,&h00,&h00,&h00,&h00,&h00, ‘K	50	
&h07,&hFF,&hFF,&h01,&h01,&h01,&h01, &h00,&h00,&h00,&h00,&h00,&h00,&h00, ‘L	51	
&h05,&h00,&h33,&h33,&h00,&h00,&h00, &h00,&h00,&h00,&h00,&h00,&h00,&h00, ‘:	52	
&h05,&h00,&hF0,&h00,&hF0,&h00,&h00,		

&hFF,&h7E,&h00,&h00,&h00,&h00,&h00, &h05,&h00,&h02,&h03,&h00,&h00,&h00, &h00,&h00,&h00,&h00,&h00,&h00,&h00, ‘,	‘IO D2
&h07,&hFD,&hFD,&h94,&h95,&h95,&h94, &h00,&h00,&h00,&h00,&h00,&h00,&h00, ‘Ë	D3
&h0C,&hFF,&hFF,&h70,&h0E,&hFF,&hFF, &h00,&h68,&h98,&h98,&h68,&h00,&h00, ‘№	D5
	D8
&h03,&h4C,&hCC,&h00,&h00,&h00,&h00, &h00,&h00,&h00,&h00,&h00,&h00,&h00, ‘;	D9
&h05,&h00,&h33,&h33,&h00,&h00,&h00, &h00,&h00,&h00,&h00,&h00,&h00,&h00, ‘:	DB
&h05,&h40,&h3F,&h3F,&h40,&h00,&h00, &h00,&h00,&h00,&h00,&h00,&h00,&h00, ‘i	ED
&h07,&h1E,&h3F,&h29,&h21,&h33,&h12, &h00,&h00,&h00,&h00,&h00,&h00,&h00, ‘e	EE
&h00,&h00,&h00,&h00,&h00,&h00,&h00, &h00,&h00,&h00,&h00,&h00,&h00, ‘	EF
&h05,&h80,&h7E,&h7E,&h80,&h00,&h00, &h00,&h00,&h00,&h00,&h00,&h00,&h00, ‘Ï	F0
&h07,&h7E,&hFF,&h91,&h91,&hC3,&h42, &h00,&h00,&h00,&h00,&h00,&h00,&h00 ‘€	F1