

НАЦІОНАЛЬНИЙ УНІВЕРСИТЕТ «ЛЬВІВСЬКА ПОЛІТЕХНІКА»
ВІДОКРЕМЛЕНИЙ СТРУКТУРНИЙ ПІДРОЗДІЛ
«ФАХОВИЙ КОЛЕДЖ ІНФОРМАЦІЙНИХ ТЕХНОЛОГІЙ
НАЦІОНАЛЬНОГО УНІВЕРСИТЕТУ «ЛЬВІВСЬКА ПОЛІТЕХНІКА»

КУРСОВИЙ ПРОЄКТ

Фаховий молодший бакалавр
(освітньо-професійний ступінь)

на тему: «Проектування систем керування та збору інформації на базі
мікроконтролерів сімейства PIC16»

Виконав студент IV курсу, групи **КІ-42**

Спеціальності **123 Комп'ютерна інженерія**

Варіант № 40

Чернишов Кирило Олександрович

(прізвище, ім'я по батькові)

Керівник

Роман СТОЛЯРЧУК

(підпис)

(ім'я прізвище)

Курсовий проєкт перевірений
і допущений до захисту:

«__» _____ 2025р.

Курсовий проєкт захищений «__» _____ 2025 р.

з оцінкою « _____ »

Львів 2025

**ВІДОКРЕМЛЕНИЙ СТРУКТУРНИЙ ПІДРОЗДІЛ
«ФАХОВИЙ КОЛЕДЖ ІНФОРМАЦІЙНИХ ТЕХНОЛОГІЙ
НАЦІОНАЛЬНОГО УНІВЕРСИТЕТУ «ЛЬВІВСЬКА ПОЛІТЕХНІКА»**

ЗАТВЕРДЖЕНО

на засіданні циклової комісії

«Фундаментальної підготовки»

Протокол № 1 від 28 серпня 2025 р.

Голова комісії

_____ Ольга ЛАБАЗ

ЗАВДАННЯ

на курсовий проєкт варіант № 40

Чернишову Кирилу Олександровичу

(прізвище, ім'я та по батькові)

з навчальної дисципліни: **МІКРОКОНТРОЛЕРИ ТА ЇХ ПРОГРАМУВАННЯ**

Студент групи: **KI-42**

1. Тема проєкту: Система моніторингу температури.

2. Дата видачі завдання: "23" вересня 2025 р

3. Термін здачі курсового проєкту: "16" грудня 2025 р.

4. Вихідні дані до проєкту:

4.1 Пристрої введення інформації : LM35, DS1307

4.2 Функції пристрою введення інформації: . читання часу, дати, температури

4.4 Використовувані мікроконтролери: PIC16F877, PIC16F689 .

4.5 Протокол обміну інформацією: SPI.

4.6 Пристрій виведення інформації: .PKI, 1 рядок

4.7 Функції пристрою виведення інформації : індикація : $T1 = +/- 21C; T2 = +/- 25C$

5 Перелік обов'язкових демонстраційних креслень:

5.1 Узагальнена структурна схема МКС

5.2 Блок-схема алгоритму роботи МКС

5.3 Скриншот роботи проєкту у програмі PROTEUS

6. Склад розрахунково – пояснювальної записки (перелік питань до розробки):

ВСТУП

- 1 Розробка структурної схеми МКС
 - 1.1 Аналіз завдання та синтез структурної схеми МКС
 - 1.2 Обґрунтування вибору мікроконтролерів
 - 1.3 Призначення, принцип роботи, особливості периферійних пристроїв
- 2 Розробка електричної принципової схеми МКС
 - 2.1 Розробка схеми електричної принципової мікроконтролера, периферійних пристроїв та їх інтерфейсу.
 - 2.2 Створення проекту з використанням програми PROTEUS
- 3 Розробка програмного забезпечення
 - 3.1 Розробка алгоритму функціонування МКС
 - 3.2 Написання керуючої програми для мікроконтролера :
 - робота з програмою MPLAB, створення проекту;
 - конфігурування МК з використанням директиви __CONFIG;
 - ініціалізація мікроконтролерів;
 - ініціалізація периферійних модулів;
 - пояснення до написання основної частини програми.
 - 3.3 Використання програми PROTEUS для аналізу роботи проекту.

ВИСНОВКИ

ПЕРЕЛІК ПОСИЛАНЬ

Календарний план

Назва етапів	Термін виконання	Примітка
Вступ		
1 Розробка структурної схеми		
2 Розробка електричної принципової схеми		
3 Розробка програмного забезпечення		
Висновки		
Перелік посилань		

Студент

_____ (підпис)

Керівник проекту

_____ (підпис)

Кирило ЧЕРНИШОВ

_____ (імя та прізвище)

Роман СТОЛЯРЧУК

_____ (імя та прізвище)

РЕФЕРАТ

Текстова частина курсового проєкту: 32 сторінки, 1 таблиці, 13 рисунків, 8 посилань.

Об'єкт проектування - "Система моніторингу температури".

Мета проєкту - довершити свої навички у розробці структурної схеми, електричної принципової схеми МКС, створення проєкту з використанням програми PROTEUS, розробці алгоритму функціонування МКС, а також написання керуючої програми для мікроконтролерів. Закріпити теоретичні знання з основних понять дисципліни "Мікроконтролери та їх П".

Результат роботи над курсовим проєктом показав, що завдяки використанню двох простих мікроконтролерів та модуля реального часу можна створити повноцінну систему моніторингу температури.

МК, DS1307, SPI, CCI-4 p., (CA), SPLAN, MPLAB IDE.

ПЕРЕЛІК УМОВНИХ ПОЗНАЧЕНЬ

АЦП	Аналого-цифрові перетворювачі
ВІС	Великі інтегральні схеми
ГЗ	Глобальна задача
ГТІ	Генератор тактових імпульсів
ЗЗ	Загальна задача
ЗМП	Задачі малої розмірності
МК	Мікроконтролер
МКП	Мікроконтролерні пристрої
МКС	Мікроконтролерна схема
НЗ	Найпростіші задачі
ОМК	Однокристальні мікроконтролери
ПЗП	Постійний запам'ятовуючий пристрій
ЦАП	Цифроаналогові перетворювачі
ЦПП	Центральний процесорний пристрій
ШІМ	Широтно-імпульсна модуляція
I2C	Inter-Integrated Circuit (Міжінтегральна схема)
SPI	Serial Peripheral Interface (Послідовний периферійний інтерфейс)

ЗМІСТ

ВСТУП.....	7
1 РОЗРОБКА СТРУКТУРНОЇ СХЕМИ МКС.....	8
1.1 Аналіз завдання та синтез структурної схеми МКС.....	8
1.2 Вибір мікроконтролерів та аналіз їх параметрів.....	9
1.3 Опис периферійних пристроїв та робота з ними.....	10
1.3.1 Давач температури DS1307.....	10
1.3.2 Рідкокристалічні алфавітно-символьні індикатори	11
1.3.3 Особливості протоколу SPI	13
1.3.4 Робота з протоколом I2C	13
1.3.5 Давач температури LM35.....	14
2 ПРОЕКТУВАННЯ ЕЛЕКТРИЧНОЇ ПРИНЦИПОВОЇ СХЕМИ.....	15
2.1 Розробка схеми електричної принципової мікроконтролера.....	15
2.2 Створення проекту в середовищі PROTEUS VSM.....	19
3 РОЗРОБКА ПРОГРАМНОГО ЗАБЕЗПЕЧЕННЯ МКС.....	21
3.1 Розробка алгоритму функціонування МКС.....	21
3.2 Конфігурування пристрою з використанням директиви __CONFIG..	25
3.3 Написання керуючої програми для мікроконтролера.....	25
ВИСНОВКИ.....	31
ПЕРЕЛІК ПОСИЛАНЬ.....	32

ВСТУП

Типовими представниками RISC-процесорів є PIC-контролери (Peripheral Interface Controller – контролери периферійних інтерфейсів) виробництва фірми MicroChip.

PIC-контролери застосовуються у системах високошвидкісного керування автомобільними й електричними двигунами, приладах побутової електроніки, телефонних приставках з АВН, системах охорони із сповіщенням по телефонній лінії, міні-АТС. Окремі вимірювальні інформаційні системи відрізняються розрядністю ПЗП(постійно запам'ятовуючого пристрою): від 12 до 14 біт для серії PIC16Cxx, 16 біт – для серії PIC17Cxx.

Завдяки скороченій кількості команд (від 33 до 35) усі команди займають у пам'яті одне слово. Час виконання кожної команди, крім команд розгалуження, становить чотири такти – один цикл (200 нс на частоті 20 МГц). Оперативний запам'ятовувальний пристрій виконано за схемою з довільною вибіркою з можливістю безпосередньої адресації у коді команди до будь-якої комірки.

Стек реалізовано апаратно з глибиною 2, 8 або 16 комірок. Майже в усіх PIC-контролерах є система переривань, джерелом яких може бути таймер, а також зміна станів сигналів на деяких входах. У PIC-контролерах передбачений біт захисту ПЗП, що запобігає нелегальному копіюванню.

Великі інтегральні схеми PIC16Cxx мають вбудовані ПЗП ємністю від 0,5 до 4 кілобайт і ОЗП(оперативно-запам'ятовуючий пристрій) ємністю 32-256 байт. Основна частина контролерів має однократно програмований ПЗП, однак деякі контролери містять ПЗП з ультрафіолетовим стиранням, а PIC 16C84 містить пам'ять програм і пам'ять даних на базі ПЗП з електричним стиранням.

Крім того, контролери мають від одного до трьох таймерів, вбудовану систему скидання, watchdog таймер, внутрішній тактовий генератор, який може запускатися як від кварцового резонатора, так і від RC-ланцюга у широкому діапазоні частот – 0-25 МГц. Кількість розрядів портів – 12-33. Кожний розряд порту можна запрограмувати на введення або на виведення.

Контролер PIC16C64 додатково має вихід з ШІМ, за допомогою якого можна реалізувати ЦАП з розрядністю до 16 розрядів, а також послідовний двонапрявлений синхронний порт з інтерфейсами SPI, I2C, SCI/UART. PIC 16C71 і PIC 16C74 мають внутрішній 8-розрядний АЦП із пристроєм вибирання/зберігання і вхідним аналоговим мультиплексором.

1 РОЗРОБКА СТРУКТУРНОЇ СХЕМИ МКС

1.1 Аналіз завдання та синтез структурної схеми МКС

При проектуванні мікроконтролерних пристроїв (МКП) або систем (МКС) можна використовувати блоково-ієрархічний підхід, при якому уявлення про МКП або МКС, що проектується, розчленовуються на ієрархічні рівні. На вищому рівні використовується найменш деталізоване уявлення, що відображає лише тільки загальні риси і особливості системи, що проектується. На наступних рівнях ступінь подробиць розгляду зростає, при цьому система розглядається не загалом, а окремими блоками. Такий підхід дозволяє на кожному рівні формулювати і вирішувати задачі допустимої складності, що піддаються усвідомленню й розумінню людиною та рішення за допомогою доступних засобів. Переваги такого підходу полягають в тому, що складна задача великої розмірності розбивається на групи задач малої розмірності, які послідовно вирішуються, причому всередині груп різні задачі можуть вирішуватися паралельно.

Структурна схема – це схема, яка визначає основні функціональні частини виробу, їх взаємозв'язки та призначення. Під функціональною частиною розуміють складову частину схеми: елемент, пристрій, функціональну групу, функціональну ланку.

Структурна схема призначена для відображення загальної структури пристрою, тобто його основних блоків, вузлів, частин та головних зв'язків між ними. Із структурної схеми повинно бути зрозуміло, навіщо потрібний даний пристрій і як він працює в основних режимах роботи, як взаємодіють його частини. Позначення елементів структурної схеми можуть обиратись довільно, хоча загальноприйнятих правил виконання схем слід дотримуватись.

Об'єктом дослідження даного курсового проекту є Система моніторингу температури, що працює на онові 2 мікроконтролерах PIC16F877, PIC16F689.

Робота приладу забезпечується двійково-десятковим годинником-календарем з низьким споживанням струму DS1307 та датчиком температури LM35.

Вивід інформації здійснюватиметься через однорядковий рідкокристалічний індикатор, що показуватиме індикацію.

Усі прилади даного проекту з'єднуються за протоколом SPI (синхронної послідовної передачі даних).

Структурна схема МКС для Системи моніторингу температури визначає основні функціональні частини виробу та зазначена на рисунку 1.1.

Перший блок даної схеми відповідає за зчитування інформації. В подальшому, наші дані після зчитування пройдуть через два мікроконтролера по протоколу SPI.

У другому блоці схеми Системи моніторингу температури зображено графічний вивід інформації, що буде мати наступний вигляд: $T1 = +/- 21C$; $T2 = +/- 25C$.

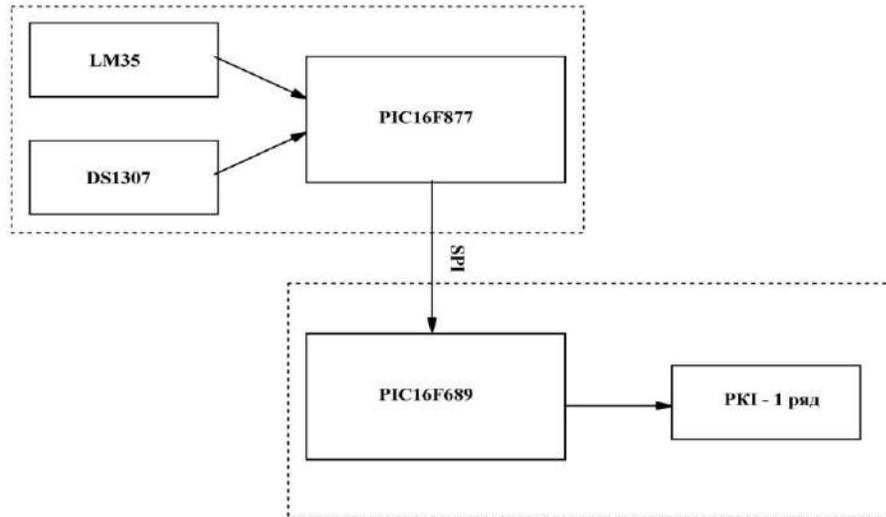


Рисунок 1.1 - Структурна схема системи секундоміра-таймера

1.2 Вибір мікроконтролерів та аналіз їх параметрів

В останні роки при розробці систем керування все більше уваги приділяється мікроконтролерній техніці. Це пов'язано з її бурхливим розвитком і широким асортиментом пропонованої продукції. Використання мікроконтролерів дозволяє конструювати пристрої з невеликими габаритами, що є відносно дешевими, простими і надійними, сумісними з персональним комп'ютером через стандартні інтерфейси.

Основна мета при виборі мікроконтролера - обрати мікроконтролер з мінімальною ціною, але в той же час він повинен відповідати вимогам продуктивності, надійності, умовам застосування, тощо.

Наступний крок при виборі мікроконтролера – пошук моделей, які задовольняють системним вимогам. Він включає підбір літератури, технічних описів і технічних комерційних журналів, а також демонстраційні консультації. Остання стадія вибору складається з кількох етапів, мета яких - звужити список прийнятних мікроконтролерів до одного. Ці етапи включають в себе аналіз ціни, доступності, засобів розробки, підтримки виробника, стабільності та наявності інших виробників. Проведення системного аналізу проекту дозволяє визначити вимоги до мікроконтролера:

- розрядність обчислювального ядра;
- набір вбудованих периферійних пристроїв (таймери, АЦП тощо...);
- апаратна організація обробки даних (структура машинного циклу, співвідношення тактів ГТІ і машинних циклів);

- можливість роботи по перериванню, за зовнішніми сигналами готовності або по командам людини;
- кількість керованих портів введення/виводу, характер передачі - байт або біт, програмне налаштування напрямку передачі;
- тип пристроїв введення/виводу, якими повинен керувати обирають мікроконтролер в проектованій системі (термінали, вимикачі, реле, клавіші, давачі, цифрові пристрої візуальної індикації, аналого-цифрові й цифро-аналогові перетворювачі, модулятори тощо);
- підтримувані способи завантаження програм в мікроконтролер, - можливість внутрішньосистемного програмування (ISP), використання при цьому стандартизованих інтерфейсів (SPI, I2C);
- кількість і тип напруги живлення;
- малогабаритні та естетичні обмеження;
- умови навколишнього середовища, необхідні для експлуатації.

Мікроконтролери PIC16F877 та PIC16F689 мають режими енергозбереження і можливість захисту коду програми.

У мікроконтролери вбудований сторожовий таймер WDT, який може бути вимкнений тільки в бітах конфігурації мікроконтролера. Для підвищення надійності сторожовий таймер WDT має власний RC генератор.

Додаткових два таймера виконують затримку старту роботи мікроконтролера. Перший, таймер запуску генератора (OST), утримує мікроконтролер в стані скидання, поки не стабілізується частота тактового генератора. Другий, таймер включення живлення (PWRT), спрацьовує після включення живлення і утримує мікроконтролер в стані скидання протягом 72мс (типове значення), поки не стабілізується напруга живлення. У більшості додатків ці функції мікроконтролера дозволяють виключити зовнішні схеми скидання.

Режим SLEEP призначений для забезпечення наднизького енергоспоживання. Мікроконтролер може вийти з режиму SLEEP по сигналу зовнішнього скидання, по переповненню сторожового таймера або при виникненні переривань.

Вибір режиму роботи тактового генератора дозволяє використовувати мікроконтролери в різних додатках. Режим тактового генератора RC дозволяє зменшити вартість пристрою, а режим LP знизити енергоспоживання. Біти конфігурації мікроконтролера використовуються для вказівки режиму його роботи.

1.3 Опис периферійних пристроїв та робота з ними

1.3.1 Модуль реального часу DS1307

Модуль реального часу з послідовним інтерфейсом DS1307 –це двійково-десятковий годинник-календар з низьким споживанням струму. Даний модуль

відраховує секунди, хвилини, години, день, дату, місяць і рік. Остання дата місяця автоматично коректується для місяців з кількістю днів менше 31, включаючи корекцію для високосного року. Модуль працює як в 24-годинному, так і в 12-годинному режимах з індикацією AM/PM. DS1307 має інтегровану схему контролю живлення, яка відслідковує проблеми живлення і у випадку їх детектування автоматично переключає модуль на живлення від іншого джерела живлення. На рисунку 1.2 представлена типова схема підключення модуля DS1307 до мікроконтролера. Розміщення виводів наведено на рисунку 1.3.

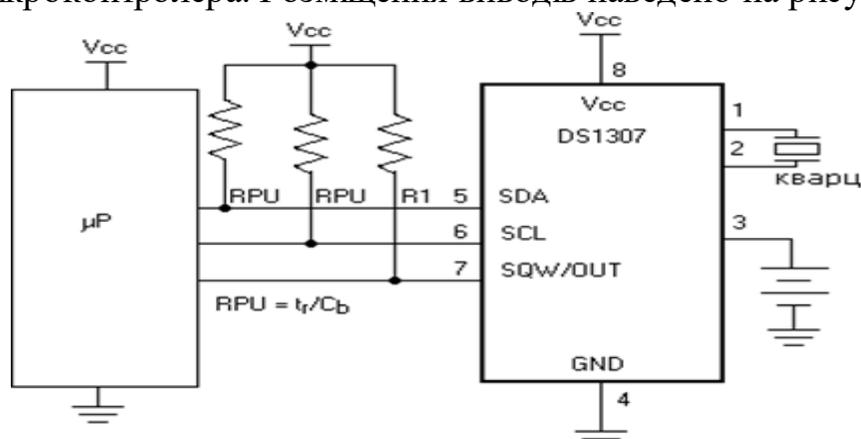


Рисунок 1.2 - Типова схема підключення модуля DS1307 до мікроконтролера

1.3.2 Рідкокристалічні алфавітно-символьні індикатори

РКІ із знаковинтезуючим контролером часто використовуються при проектуванні цифрових пристроїв.

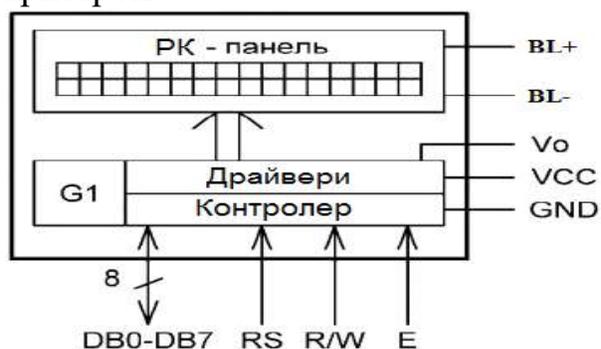


Рисунок 1.3 – Структурна схема РКІ

Основою РКІ є матриця з рідких кристалів, подаючи напругу на елементи якої можна "засвітити" крапку на екрані. Матриця складається із знакомісць (найчастіше 5*8 пікселів), згрупованих у кілька рядків. Цим керує вбудований контролер РКІ. Він керує пікселями РК – дисплею і інтерфейсною частиною індикатору.

Рідкокристалічні знакові цифрові табло належать до найбільш розповсюджених типів індикаторів. Ці індикатори бувають 1,2,4 стрічковими. У

стрічці 8, 12, 16, 20, 24, 40 символів, кожний з яких формується шрифтом формату 5*8 або 5*11 пікселів.

Завдання підключення зводиться до організації обміну даними між мікроконтролером та вбудованим контролером. Нижче показані приклад схеми підключення для восьмибітного інтерфейсу.



Інтерфейс має вісім інформаційних ліній: DB7..DB0 і три керуючих: RS, R/W, E. Лінія RS визначає до якого регістру контролера РКІ ми прагнемо звернутися, тобто яку інформацію ми передаємо - дані або команди. Лінія R/W визначає напрямок передачі даних - запис у РКІ або читання із РКІ. Лінія E включає (коли на лінії високий рівень) або виключає (коли на лінії низький рівень) передачу інформації, сформованої на інших інтерфейсних лініях. Основні параметри наведені у технічній літературі.

Ініціалізація РКІ полягає в передачі декількох команд у певній послідовності. Кількість команд ініціалізації може відрізнитися в різних контролерів. Під час ініціалізації краще не аналізувати прапорець BUSY, а чекати визначений час перед посилкою наступної команди, тому що прапорець встановлюється не відразу, а після певної команди (див. даташит). Ініціалізація РКІ виконується аналогічно як для паралельного інтерфейсу підключення так і для послідовного (підключення РКІ по I2C чи SPI).

Різниця між 8-бітним та 4-бітним інтерфейсом полягає лише в тому, що при 4-бітній передачі даних необхідно подавати сигнали RS, R/W як для молодшої так і для старшої тетради. Команди ініціалізації роботи РКІ залишаються однаковими.

У РКІ є 2 види пам'яті: DDRAM, CGRAM (CGROM). DDRAM - display data RAM (пам'ять дисплея) - те, що записане в цій пам'яті, безпосередньо відображається на дисплеї. Ця пам'ять має наступний адресний простір і відповідне відображення на дисплеї.

Тобто, те, що записане в DDRAM за адресою, наприклад, 42h, буде відображатися в третій позиції на другому рядку дисплея.

CGRAM (CGROM) - character generator RAM (ROM) - пам'ять

знакогенератора. Пам'ять знакогенератора розділена на CGRAM - доступна для запису/читання, сюди можна залити 8 своїх власних символів і CGROM - доступна тільки для читання.

1.3.3 Особливості протоколу SPI

Послідовний периферійний інтерфейс (Serial Peripheral Interface - SPI) забезпечує високошвидкісний синхронний обмін даними між мікроконтролером і периферійними пристроями, а також між декількома мікроконтролерами.

В SPI режимі можливі одночасний синхронний прийом і передача 8-розрядних даних.

Режим ведучого SPI. Ведучий може ініціалізувати передачу даних у будь-який момент, оскільки він генерує тактовий сигнал, і визначає, коли ведений повинен передати дані відповідно до використовуваного протоколу.

У режимі ведучого дані передаються/прийняті після їхнього запису/читання з регістра SSPBUF. Якщо в SPI режимі потрібно тільки приймати дані, вивід SDO може бути заблокований (настроений як вхід). Дані з виводу SDI послідовно зсуваються в регістр SSPSR із встановленою швидкістю. Кожний прийнятий байт завантажується в регістр SSPBUF (як нормально отриманий байт) з формуванням переривань і впливом на відповідні біти статусу. Ця функція може бути корисна при реалізації "монітора шини".

Будь-яка небажана функція послідовного порта може бути виключена, налаштувавши відповідні біти регістрів напрямку даних TRIS. Наприклад, якщо в режимі ведучого SPI виконується тільки передача даних, то виводи SDI й -SS можуть використовуватися як цифрові виходи, скинувши відповідні біти TRIS в '0'.

1.3.4 Робота з протоколом I2C

DS1307 підтримує обмін даними по протоколу I2C по двопровідній двонаправленій шині. Пристрій, який передає дані на шину, є передавачем, а пристрій, приймаючий дані, - відповідно приймачем. Ведучий пристрій генерує синхроімпульси (serial clock - SCL), керує доступом до шини і генерує умови START і STOP. DS1307 працює на шині як ведений пристрій. Типова конфігурація шини з використанням протоколу I2C наведена на рисунку 1.5.

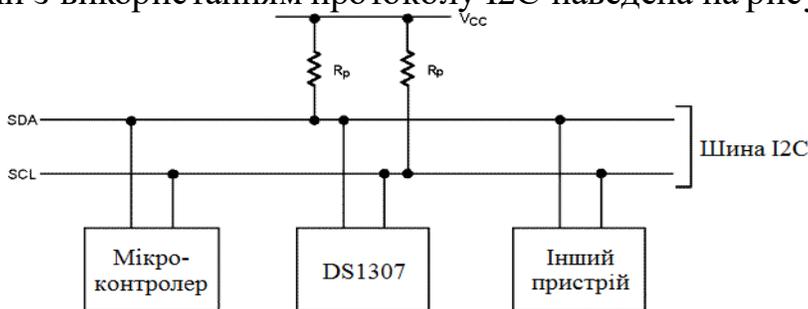


Рисунок 1.5 - Типова конфігурація двопровідної шини (I2C)

DS1307 на послідовній шині працює як ведений пристрій. Доступ до нього здійснюється встановленням умови START і передачею пристрою ідентифікаційного коду, після якого передається адреса регістру. До наступного за обраним регістром доступ здійснюється послідовно, поки не буде виконана умова STOP.

Кожна передача даних ініціюється умовою START і завершується умовою STOP. Кількість байтів даних, що передаються між умовами START і STOP, не обмежена і визначається ведучим пристроєм. Інформація передається по-байтово, і кожний байт приймач підтверджує дев'ятим бітом (біт підтвердження - ACK). В специфікації двох провідного інтерфейсу визначені звичайний режим і швидкий режим (с тактовою частотою 400 кГц). DS1307 працює тільки в звичайному режимі (100 кГц).

Для підтвердження прийому пристрій має підтягнути до низького рівня лінію SDA під час тактового імпульсу таким чином, щоб на лінії SDA залишався стабільний низький рівень.

У режимі веденого приймача послідовні дані приймаються по SDA і синхронізуються по SCL. Після кожного прийнятого байту передається біт підтвердження. Умови START і STOP розпізнаються як початок і кінець послідовної передачі. Розпізнавання адреси виконується апаратно після прийому адреси веденого і біта напряму передачі.

1.3.5 Датчик температури LM35

Серія LM35 - це точні аналогові датчики температури з інтегральним контуром, вихідна напруга яких лінійно пропорційна температурі Цельсія. Таким чином, LM35 має перевагу перед лінійними датчиками температури, каліброваними в Кельвінах, оскільки користувачеві не потрібно віднімати велику постійну напругу від його виходу для отримання зручного масштабування. LM35 не вимагає жодного зовнішнього калібрування або обрізки для забезпечення типової точності $\pm 1/4$ °C при кімнатній температурі та $\pm 3/4$ °C протягом повного діапазону температур від -50 до + 150 °C.

Розпіновка LM35 - це досить просто. Перший штифт призначений для необхідного джерела живлення датчика, яке переходить від 4 до 30 В, хоча воно може відрізнятись залежно від виробника, отже, краще поглянути на таблицю даних датчика що ви придбали. Потім, у центрі, у нас є штифт для виходу, тобто той, який буде подавати ту чи іншу напругу в залежності від температури. І третій штифт шліфується.

2 ПРОЕКТУВАННЯ ЕЛЕКТРИЧНОЇ ПРИНЦИПОВОЇ СХЕМИ

2.1 Розробка схеми електричної принципової мікроконтролера

У процесі проектування будь-якого пристрою, формується набір технічної документації по цьому пристрою. В технічній документації описується параметри, робота пристрою, необхідні для виготовлення компоненти та методи і засоби налагодження функціональних можливостей проектованого пристрою.

Одним із найважливіших розділів технічної документації являється подання електричної принципової схеми проектованого пристрою.

Електрична принципова схема – це графічне зображення будови пристрою на фізичному рівні. Дана схема відображає кількість та тип використовуваних у пристрої компонентів (радіотехнічних елементів, мікросхем, інтерфейсних роз'ємів, тощо...) та їх взаємодію один з одним. Підключення компонентів схеми один до одного відображається лініями. Згідно цієї схеми відбувається розведення доріжок на платі у відповідності до ліній підключення, зображених на схемі, та монтаж і пайка елементів проектованого пристрою. Усі елементи, зображені на електричній принциповій схемі, підписуються згідно вимог ДСТУ.

Жоден електричний пристрій не може працювати без джерела напруги живлення. Пристрої, що побудовані на основі мікроконтролера, це цифрові пристрої. Діапазон напруги живлення для таких пристроїв складає від 1.5 до 5.5В, а струм - постійний. Для підключення цифрових пристроїв до мережі 220В використовуються трансформаторні, імпульсні, конденсаторні блоки живлення тощо... До складу найпростішої схеми входять: понижуючий трансформатор, діодний міст, електролітичні конденсатори та мікросхема стабілізації (див. рисунок 2.1).

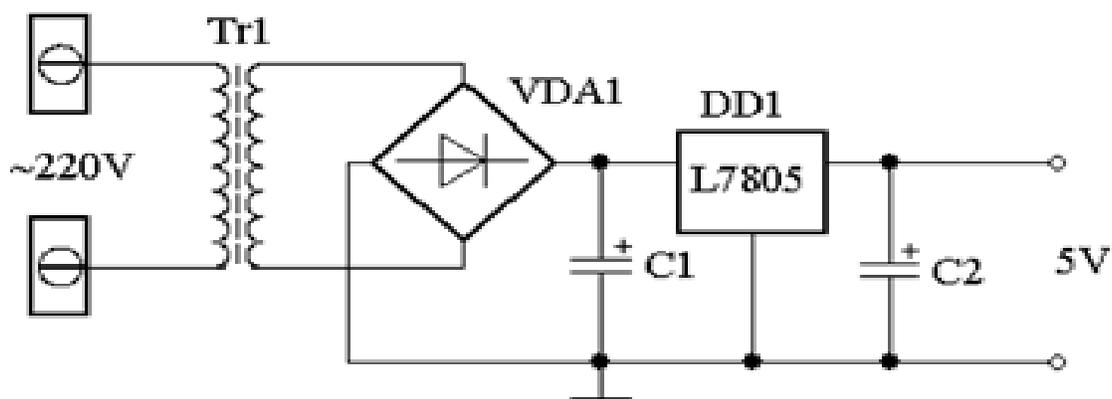


Рисунок 2.1- Схема стабілізації живлення для цифрових пристроїв на основі мікроконтролерів

Ключовим елементом в даній схемі є стабілізатор L7805. Існує два види стабілізатора 7805: з струмом навантаження до 1А і малопотужний 78L05 з струмом навантаження до 0,1А. Крім них є проміжний варіант 78M05 з струмом навантаження до 0,5А.

Ємність С1 на вході стабілізатора необхідна для згладження високочастотних завад при подачі вхідної напруги. Ємність С2 на виході стабілізатора задає стабільність напруги при різкій зміні струму навантаження, а також суттєво знижує амплітуду пульсацій. Для його нормальної роботи напруга на вході повинна бути у діапазоні 10-20 В.

Для формування напруги такого рівня використовується однофазний трансформатор та діодний міст. Для того, щоб сформувати необхідні умови для роботи МКС необхідно передбачити джерело імпульсів синхронізації.

В мікроконтролерах сімейства PIC16 передбачено декілька типів генераторів. Для PIC16 користувач може запрограмувати два конфігураційних біти (FOSC1 і FOSCO) для вибору одного із чотирьох режимів: RC, LP, XT, HS., де:

RC – генератор на RC-ланці;

XT – стандартний кварцовий генератор;

HS – високочастотний кварцовий генератор;

LP – низькочастотний малої потужності генератор.

В режимах XT, LP і HS до виводів OSC1/CLKIN і OSC2/CLKOUT підключається кварцовий або керамічний резонатор.

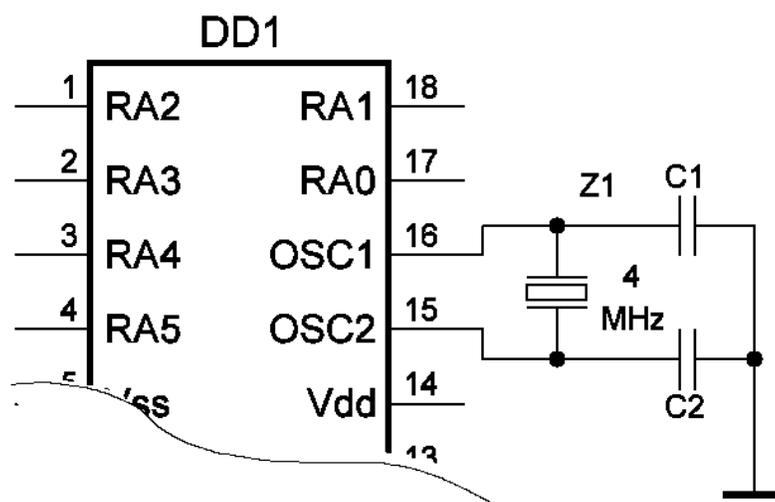


Рисунок 2.2 - Схема тактування мікроконтролера

Для підключення резонатора використовується типова схема виробника. Ємність конденсаторів повинна бути в інтервалі від 15 до 22 пФ, один вивід яких підключається до резонатора, а інший до землі.

У пристроїв, побудованих на основі мікроконтролерів передбачений механізм апаратного ресетування. У мікроконтролерів PIC16 вивід, що призначений для ручного перезапуску називається /MCLR. Активний сигнал для скидання – це сигнал логічного 0, що подається на вхід /MCLR при натисканні кнопки S1.

Для зміни програмного забезпечення МК використовується роз'єм ICSP. Даний роз'єм, як і одноіменний протокол обміну, дозволяє змінювати робочі параметри мікроконтролера без вилучення його із пристрою. При прошиванні мікроконтролера на ніжку /MCLR подається сигнал рівнем 15В. Даний сигнал позначено на роз'ємі як V_{pp} . Задіюється механізму доступу до пам'яті мікроконтролера. Дані передаються по лінії Data на вивід PGD мікроконтролера, при цьому з лінії Clock роз'єму ICSP поступає сигнал тактування. Даний сигнал приймається виводом PGC мікроконтролера, та відповідає за коректний запис програми у мікроконтролер. Сигнали Vss і Vdd забезпечують стандартну напругу живлення для мікроконтролера рівнем +5В.

На електричній принциповій схемі показано як взаємодіють різні вузли між собою та які компоненти використовуються для побудови проектного пристрою. Згідно із цією схемою у подальшому формуватиметься принцип роботи усієї системи та алгоритм функціонування мікроконтролера, як основного вузла керування.

Пристрої введення/виведення – це різного роду периферійні модулі, які підключаються до функціонально завершеного пристрою для керування роботою цього пристрою та отримання інформації про виконувани ним функції. Наприклад, для ПК такими пристроями є клавіатура, мишка (пристрої введення) та монітор і принтер (пристрої виведення).

Для мікроконтролерних систем номенклатура пристроїв введення чи виведення дещо інша. Пристроями вводу крім клавіатури (клавіатурної матриці) можуть буди різні датчики – тепла, вологості, загазованості, ПЧ-датчики, тощо... В свою чергу пристрої виводу також не обмежуються монітором. Для МКС це, – як правило, – знаковимвольні або графічні РКІ, ССІ, акустичні елементи (біпери, динаміки), навідь звичайні світлодіоди.

Найпростіші із цих пристроїв введення/виведення інтегруються в пристрій та підключаються безпосередньо до портів мікроконтролера, проте є і такі, для підключення яких використовуються інтерфейсні роз'єми.

Інтерфейс – у мікроконтролерній техніці – це набір провідників, які використовуються для підключення зовнішнього пристрою вводу/виводу та забезпечують реалізацію протоколу передачі даних. Як правило, такі групи провідників об'єднуються в межах одного роз'єму. Наприклад, для зовнішніх пристроїв, які працюють по USART/RS-232 протоколу, група провідників, яка забезпечує зв'язок із мікроконтролером основного пристрою по цьому протоколу об'єднується у роз'єм типу D-SUB9.

Наведена електрична принципова схема працює наступним чином. Після подачі живлення мікроконтролер покроково виконує програму, що закладена у його пам'ять програм. Першим етапом є ініціалізація портів мікроконтролера та його вбудованих модулів. Активується інтерфейс зв'язку та виконується обмін даними.

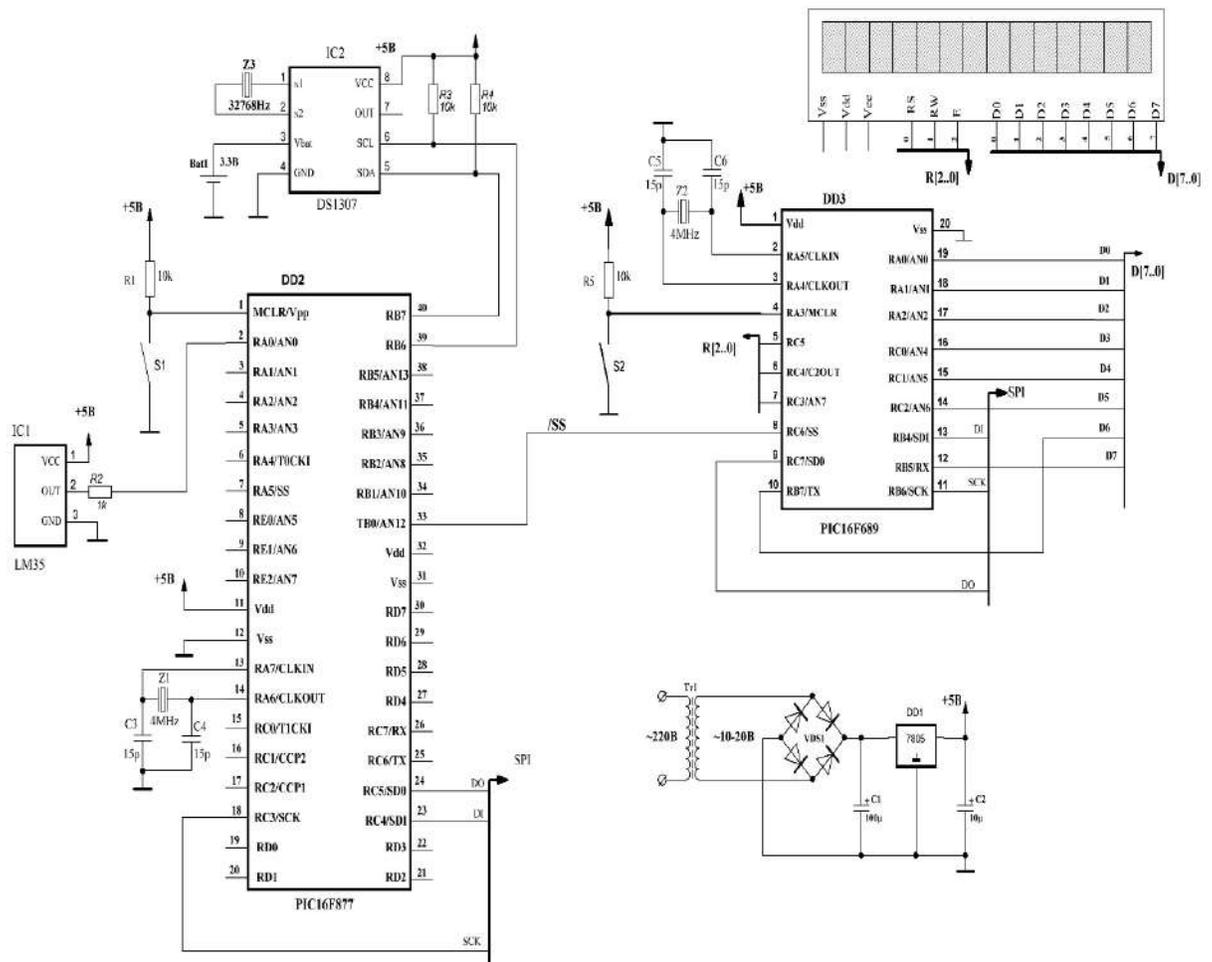


Рисунок 2.3 - Електрична принципова схема роботи Системи моніторингу температури

Таблиця 2.1 – Перелік елементів електричної принципової схеми

№ П/П	Назва елемента	Позначення	Номінал
1	Трансформатор	TR1	220В
2	Діодний міст	VDS1	D13A4
3	Регулятор напруги	DD1	7805
4	Кварцеві резонатори	Z1,Z2	4МГц
5	Кварцевий резонатор	Z3	32768Гц
6	Конденсатор	C1	100мкФ

7	Конденсатор	C2	10мкФ
8	Конденсатори	C3,C4,C5,C6	15пФ
9	Резистори	R1,R3, R4, R5	10кОм
10	Резистори	R2	1кОм
11	Мікроконтролери	DD2,DD3	PIC16F877, PIC16F689
12	Датчик температури	IC1	LM35
13	1-рядковий РКІ	LCD	LM016L
14	Кнопка-перемикач	S1,S2	SK1-53
15	Модуль реального часу	IC2	DS1307
16	Батарея	Bat1	3.3В

2.2 Створення проекту в середовищі PROTEUS VSM

Більшість радіоаматорів стикалися із ситуацією, коли за браком досвіду чи присутність помилок у схемі або ж за іншими обставинами, псували радіоелектронні компоненти.

З'явилася величезна кількість програм симуляторів, що замінюють реальні радіодеталі й прилади, віртуальними моделями. Симулятори дозволяють, без збирання реального пристрою, налагодити роботу схеми, знайти помилки, отримані на стадії проектування, зняти необхідні характеристики й багато чого іншого. Однією з таких програм є PROTEUS VSM.

Proteus VSM складається із двох самостійних програм ISIS та ARES. ARES це трасировальник друкованих плат з можливістю створення своїх бібліотек. Запуск програми відбувається з меню Пуск – Програми - Proteus 7 Professional - ISIS 7 Professional. При запуску програми з'являється основне вікно програми ISIS 7 Professional.

Найбільший простір відведений під вікно редагування EDIT WINDOW. Саме в ньому відбуваються всі основні процеси створення, редагування й налагодження схеми пристрою. Ліворуч угорі маленьке вікно попереднього перегляду Overview Window з його допомогою можна переміщуватися по вікну редагування (клацаючи лівою кнопкою миші по вікну попереднього перегляду, можна переміщувати вікно редагування за схемою, якщо звичайно схема не вміщається у вікно). Переміщувати вікно редагування за схемою можна ще утримуючи натиснутої кнопку SHIFT та рухаючи курсором миші по вікну редагування. Наближувати й віддаляти схему у вікні можна кнопками F6 й F7 або ж колесом миші, F5 центрує схему у вікні, а натискання F8 підганяє розмір схеми під-вікно редагування.

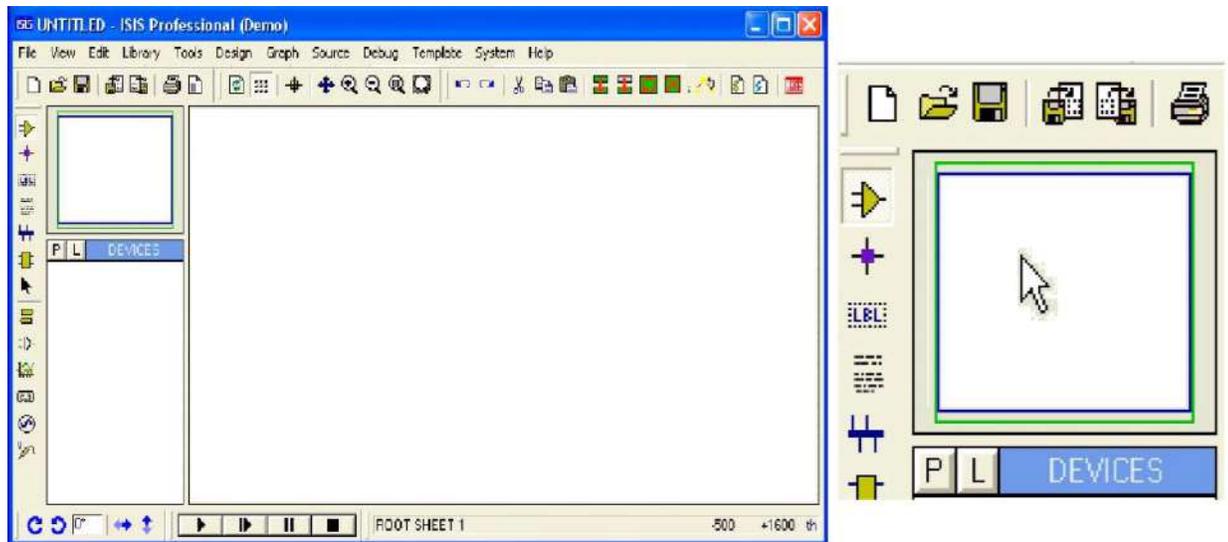


Рисунок 2.4 - Основне вікно програми ISIS 7 Professional і вікно попереднього перегляду.

Під вікном попереднього перегляду знаходиться Object Selector - список обраних у цей момент компонентів, символів й інших елементів. Виділений у списку об'єкт відображається у вікні попереднього перегляду.

Всі можливі функції та інструменти Proteus VSM доступні через головне меню, через піктограми які знаходяться під меню і у лівому куті основного вікна та через гарячі клавіші, які можуть перепризначуватися користувачем. Внизу основного вікна розташовані: кнопки обертання та розвороту об'єкта навколо своєї осі, панель керування інтерактивною симуляцією - виглядає як магнітофонна і функції такі ж: ПУСК, ПОКРОКОВИЙ РЕЖИМ, ПАУЗА, СТОП.

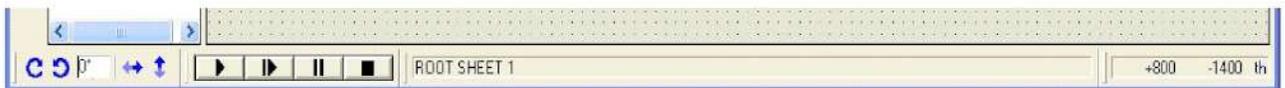


Рисунок 2.5 - Панель керування інтерактивною симуляцією.

Середовище PROTEUS має величезну бібліотеку електронних компонентів. Всі елементи перебувають у бібліотеці компонентів. Щоб до неї потрапити потрібно перейти в режим COMPONENT (компоненти), натиснувши відповідну піктограму P (Pick devices) або двічі клацнувши лівою кнопкою в полі вибору компонентів Object Selector.

Компоненти можна вибирати по категоріях, та у списку. Зображення компонента з'явиться у вікні попереднього перегляду. Щоб розмістити компонент на робочому полі потрібно навести курсор миші на потрібне місце і клацнути лівою кнопкою миші. Щоб з'єднати виводи компонентів провідником

потрібно підвести курсор миші до виводу компонента, після появи на виводі компонента хрестика потрібно клацнути лівою кнопкою миші і підвести курсор миші до виводу іншого компонента.

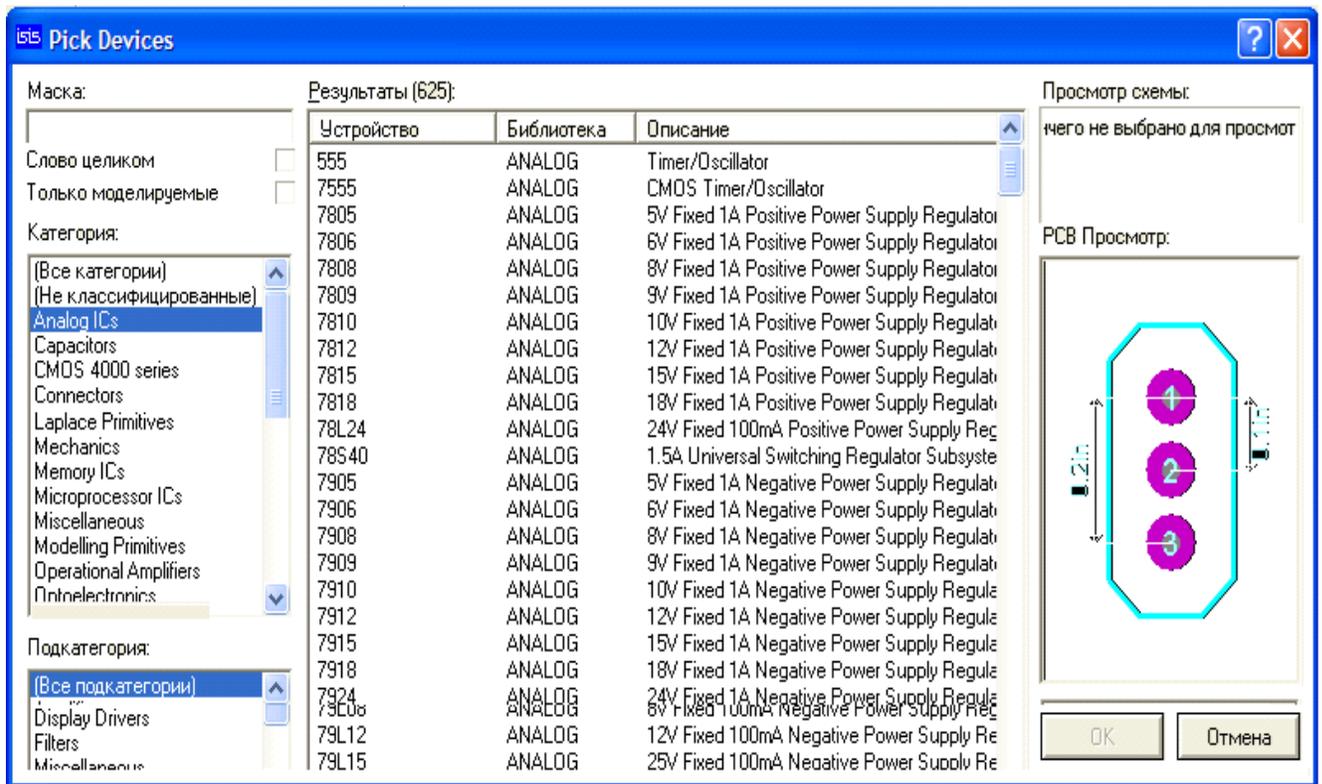


Рисунок 2.6 - Бібліотека електронних компонентів.

Для того, щоб переглянути роботу схеми, необхідний вихідний HEX-файл. Середовище PROTEUS підтримує багато засобів розробки, серед них і HI-TECH Cі компілятор і CROWHILL PIC BASIC і BASIC STAMP. Попередньо HEX-файл повинен бути створений з використанням середовищем програмування мікроконтролерів від фірми MICROCHIP, а саме MPLAB.

3 РОЗРОБКА ПРОГРАМНОГО ЗАБЕЗПЕЧЕННЯ МКС

3.1 Розробка алгоритму функціонування МКС

Алгоритм — це послідовність, система, набір систематизованих правил виконання обчислювального процесу, що обов'язково приводить до розв'язання певного класу задач після скінченного числа операцій. При написанні комп'ютерних програм алгоритм описує логічну послідовність операцій. Для візуального зображення алгоритмів використовують блок-схеми.

Блок-схема – схематичне зображення послідовності дій, спрямованих на досягнення спільної мети та їх особливостей. Розділ програми, у якому відбувається занесення значення змінних подається у вигляді паралелограма, в якому перераховані імена цих змінних. Розділ програми, у якому відбувається виконання певної операції подається у формі прямокутника, а розділ перевірки умови – у формі ромба. Крім того можуть використовуватись розділи, які повторюють певний набір команд для іншого аналогічного випадку. Такі розділи називають блоком ідентичних операцій та зображують у вигляді шестикутника. Ще один розділ, що є поширеним у програмах – це розділ виконання підпрограми. Зображується він у формі прямокутника із двома вертикальними смугами по краях фігури та записом у ньому назви підпрограми.

Кожен алгоритм є списком добре визначених інструкцій для розв'язання задачі. Починаючи з початкового стану, інструкції алгоритму описують процес обчислення, які відбуваються через послідовність станів, які, зрештою, завершуються кінцевим станом.

Принципи побудови програмного забезпечення точно такі ж, як і апаратної частини: створення кінцевого продукту з готових блоків - підпрограм. Якщо всю роботу програми розмістити у векторах переривань, то основний додаток ніяк не відчує виконання фрагментів сусідніх підпрограм. Головне визначити можливості апаратної частини.

Програмне забезпечення мікроконтролера необхідно будувати з максимальним використанням периферії. При цьому ядро не виконуватиме безлічі рутинних операцій.

Після включення живлення сервісна програма починає проводити ініціалізацію. На початку програми прикріплюється файл бібліотеки мікроконтролера, після чого оголошуються біти портів і регістри пам'яті, які використовуються в програмі. Мікроконтролер має вільні для використання комірки пам'яті і, щоб при кожному зверненні до певної комірки не описувати її адресу, та полегшити формування програми - присвоюється їй назва.

У поняття ініціалізації входить:

- присвоєння початкових значень всім змінним програми;
- налаштування всіх внутрішніх систем мікроконтролера;

- відновлення режимів роботи системи, встановлених у попередньому сеансі роботи шляхом зчитування відповідних параметрів з EEPROM-пам'яті;
- виконання підпрограм, призначених для налаштування всіх периферійних модулів у вихідний стан;
- виконання підпрограми запуску системних лічильників-таймерів і механізму переривань мікроконтролера.

Для використання у програмі констант і змінних необхідно їх попередньо описати та оголосити. Константа - це просто деяке число, що часто використовується в програмі, і має певний зміст.

Крім констант, які, відповідно до своєї назви, протягом усього часу роботи програми ніколи не міняються, у програмі існують змінні. Під змінною в програмі розуміють ім'я регістра, або адресу комірки пам'яті, де в процесі виконання програми буде тимчасово зберігатися заздалегідь невідома величина, або проміжний результат обчислень.

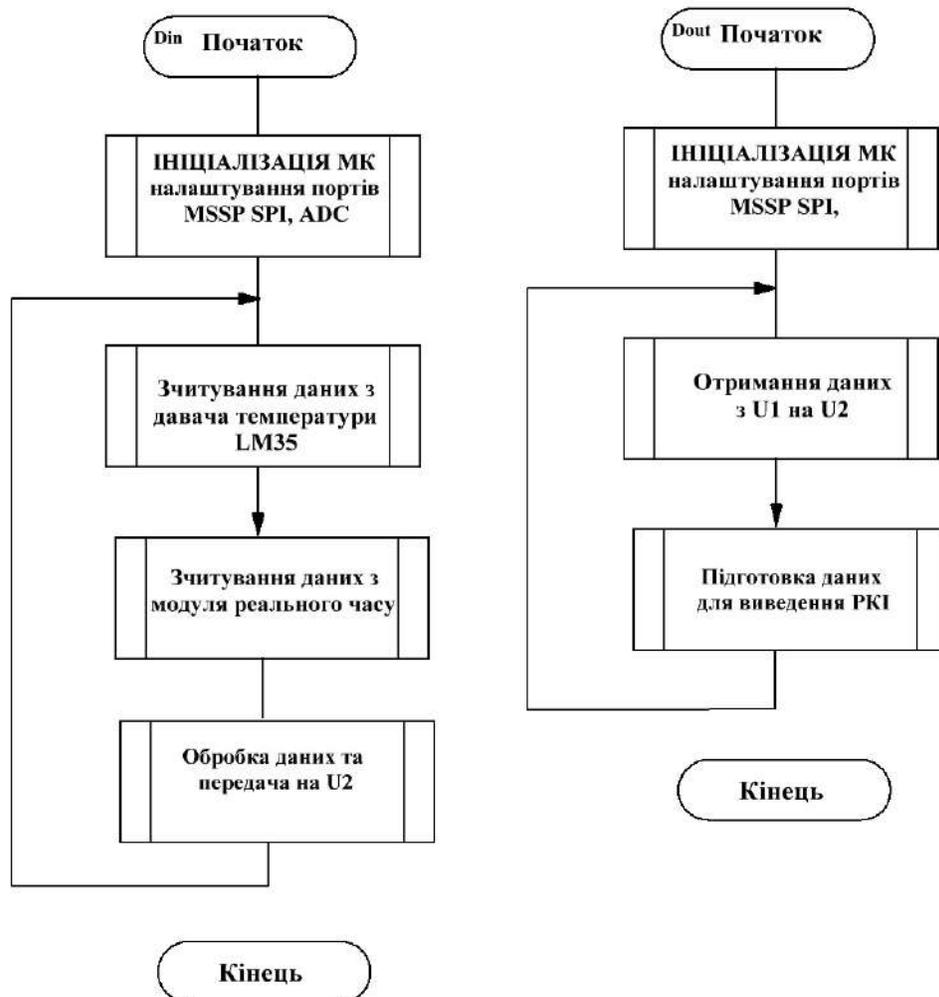


Рисунок 3.1 - Блок-схеми роботи Система моніторингу температури

3.2 Конфігурування пристрою з використанням директиви __CONFIG

Сімейство мікроконтролерів PIC16 має набір спеціальних функцій, призначених для розширення можливостей системи, мінімізації вартості, виключення навісних компонентів, забезпечення мінімального енергоспоживання і захисту коду від зчитування. В PIC16 реалізовані наступні спеціальні функції:

- вибір типу генератора;
- скидання;
- схема скидання по включенню живлення ;
- таймер скидання (DRT);
- сторожовий таймер (WDT)
- режим пониженого енергоспоживання (SLEEP);
- захист коду від зчитування;
- біти ідентифікації.

В регістрі конфігурації описуються біти конфігурації, в які входять:

- біт захисту
- біт дозволу роботи таймера включення живлення
- біт дозволу роботи сторожового таймера
- вибір типу генератора

Біти конфігурації задаються за допомогою директиви __CONFIG.

Фрагмент заголовку при написання програми для мікроконтролера PIC16F877 з використанням мови C:

```
__CONFIG(FOSC_XT&WDTE_OFF&PWRTE_OFF&CP_OFF&BOREN_OFF&LVP_OFF&CPD_OFF & WRT_OFF &DEBUG_OFF);
```

Фрагмент заголовку при написання програми для мікроконтролера PIC16F689 з використанням мови C:

```
#include <pic.h>
__CONFIG(FOSC_INTRCIO&WDTE_OFF&PWRTE_OFF&CP_OFF&BOREN_OFF&LVP_OFF&CPD_OFF&WRT_OFF&DEBUG_OFF &MCLRE_OFF);
```

3.3 Написання керуючої програми для мікроконтролера

Після включення живлення сервісна програма починає проводити ініціалізацію даних. На початку програми прикріплюється файл бібліотеки мікроконтролера, після чого оголошуються біти портів і регістри пам'яті, які використовуються в програмі. Даний тип мікроконтролера має вільні для використання комірки пам'яті і, щоб при кожному зверненні до певної комірки

не описувати її адресу, та полегшити формування програми - присвоюється її назва.

У поняття ініціалізації входить:

- присвоєння початкових значень всім змінним програми;
- налаштування всіх внутрішніх систем мікроконтролера;
- відновлення режимів роботи системи, встановлених у попередньому сеансі роботи шляхом зчитування відповідних параметрів із флеш-пам'яті;
- виконання підпрограм, призначених для установки всіх периферійних пристроїв схеми позиціонера у вихідний стан (очищення індикаторів, підтвердження команди зупинки двигуна тощо...);
- виконання підпрограми запуску системних лічильників-таймерів і механізму переривань мікроконтролера.

Крім констант, які, відповідно до своєї назви, протягом усього часу роботи програми ніколи не міняються, у програмі існують змінні. Під змінною в програмі розуміється ім'я регістра, або адреса комірки пам'яті, де в процесі виконання програми буде тимчасово зберігатися заздалегідь невідома величина, або проміжний результат обчислень.

Після виконання модуля ініціалізації програма переходить до основного циклу. У цьому циклі програма перебуває увесь час, поки включене живлення.

Першою частиною нашого проекту було написання тексту програми для першого мікроконтролера, що виконує функцію одержання даних з пристроїв DS1307 та LM35, а також обробку та передачу даних на другий МК.

Наведений нижче модуль програми здійснює ініціалізацію портів мікроконтролера PIC16F877 та модуля SPI:

```
#include "МК_init.h"
//---- опис функцій -----
void МК_init(void)
{

    //--Налаштування портів--//
    //налаштування порта В на введення, окрім біта RB5, тому що він
    використовується як керуючий біт в протоколі SPI
    TRISB=0b11101111;//перші два виводи здійснюють читання даних з
    DS1307
    TRISA1=1; //вивід читання даних з LM35
    TRISC=0; //порт С встановлюємо в режим виведення, оскільки по цьому
    порту здійснюється зв'язок за даним протоколом
    //----- MSSP SPI -----// Налаштування модуля протоколу
    //--- SSPSTAT
    SMP=0;// опитування входу в середині періоду виведення даних
```

```

    SCKE=0;// вибір фронту тактового сигналу, дані передаються по задньому
фронту сигналу на вивід SCK
    //--- SSPCON1
    CKP=0;// дані передаються по задньому фронту сигналу на вивід SCK
    // Режим роботи модуля MSSP:
    // ведучий режим SPI, тактовий сигнал = Fosc/4
    SSPM3=0;
    SSPM2=0;
    SSPM1=0;
    SSPM0=0;
    SSPEN=1;
    GIE=0;
}

```

Після ініціалізації ми виконуємо написання керуючої програми у файлі main.c :

```

__CONFIG(FOSC_HS&WDTE_OFF&PWRTE_OFF&CP_OFF&BOREN_OFF&LVP_OFF&CPD_OFF&WRT_OFF&DEBUG_OFF);

```

```

#ifndef _XTAL_FREQ
#define _XTAL_FREQ 4000000
#endif

#include "MK_init.h" //підключення файлу ініціалізації МК
#include "i2c_p.h"//підключення файлу з функціями для роботи з DS1307

//---- ОСНОВНА ПРОГРАМА -----
void main(void)
{
    MK_init();//підключення файлу ініціалізації МК
    while(1)
    {
//Спочатку ми здійснюємо отримання даних про температуру з LM35
        RB4=temp;
        spi_trans(temp);//та передаємо це значення на другий МК

//Далі ми здійснюємо читання та передачу даних часу з DS1307 на другий МК:
        i2c_start_p();//початок читання даних
        i2c_trans_p(0b11010001);//0xD1
        sec=i2c_recieve_ack_p();//запис даних у змінну
        hvl=i2c_recieve_ack_p(); //запис даних у змінну
    }
}

```

```

god=i2c_recieve_nack_p(); //запис даних у змінну
i2c_stop_p();

spi_trans(sec);
spi_trans(hv1);
spi_trans(god);

    }
}

```

Другою частиною нашого проекту було написання тексту програми для другого мікроконтролера, що виконує функцію одержання даних з першого МК та виводить отримані дані на однорядковий РКІ.

Наведений нижче модуль програми здійснює ініціалізацію портів мікроконтролера PIC16F689 та модуля SPI:

```

#ifndef _XTAL_FREQ
#define _XTAL_FREQ 4000000
#endif

#include "МК_init.h"
//---- опис функцій -----
void МК_init(void)
{
    //--Налаштування портів--//
//виключаємо аналогово-цифровий перетворювач
    CHS3=1;
    CHS2=1;
    CHS1=1;
    CHS0=1;

    TRISA=0;//порт А налаштовуємо на виведення, для того щоб виводити
відповідні символи РКІ
    TRISC=0b11111000;//перші біти даного порта ми налаштовуємо на вивід,
оскільки це керуючі біти РКІ, а 7 на ввід, тому що це керуючий біт протоколу
    //Виводи на порту В для виведення даних на РКІ:
    TRISB5=1;
    TRISB7=1;
//Виводи одержання даних з першого МК:
    TRISB4=1;
    TRISB6=1;

```

```

//----- MSSP SPI -----
//--- SSPSTAT
SMP=0; // опитування входу в середині періоду виведення даних
SCKE=0; вибір фронту тактового сигналу, дані передаються по задньому
фронту сигналу на вивід SCK

//--- SSPCON1
SCKP=0; // дані передаються по задньому фронту сигналу на вивід SCK
// Режим роботи модуля MSSP:
// ведений режим SPI, тактовий сигнал з вивода SCK. Вивід -SS
SSPM3=0;
SSPM2=1; //slave
SSPM1=0;
SSPM0=0;
SSPEN=1;
    GIE=1;
}

```

Після ініціалізації ми здійснюємо написання керуючої програми у файлі main.c :

```

void main(void)
{

    МК_init(); //модуль ініціалізації МК
    lcd_init(); //модуль ініціалізації РКІ

    while(1)
    {
//виконуємо зчитування даних з першого МК, а саме час:
        sec=spi_receive();
        hvl =spi_receive();
        god =spi_receive();

//та температуру:
        temp=spi_receive();

//наступний фрагмент здійснює виведення отриманих даних на РКІ:
//час
        lcd_pos(0); //встановлення курсора в 0 позицію
        sprintf(MAS, "%x:%x:%x", sec, hvl, god); //виведення самих даних

```

```

lcd_string();
//температура
lcd_pos(8); //встановлення курсора в 8 позицію
sprintf(MAS, "%x", temp); //виведення температури
lcd_string();
    }
}

```

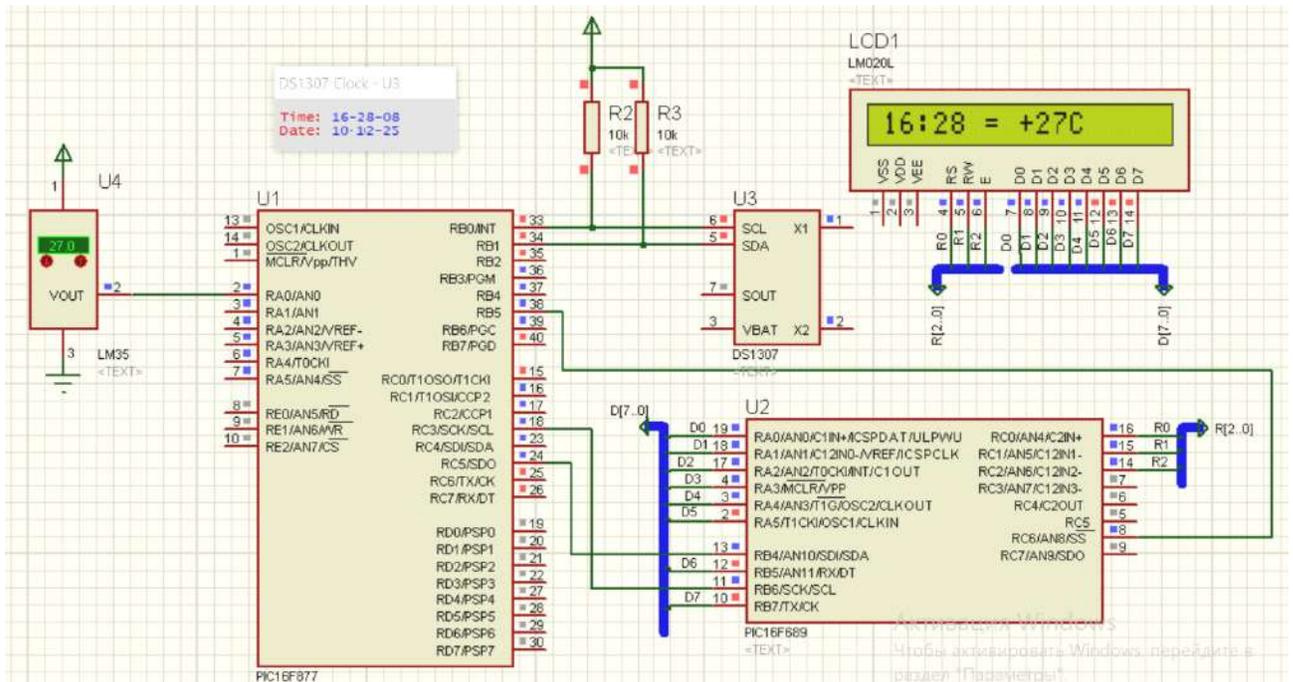


Рисунок 3.2 - Робоча схеми в PROTEUS

ВИСНОВКИ

В процесі курсового проектування розглянуто принципи проектування пристроїв і систем, що побудовані з використанням однокристальних мікроконтролерів фірми Microchip. Під час виконання проектування здійснено закріплення теоретичних знань та практичних навичок по реалізації етапів проектування. Узагальнено технічний матеріал для формування основної ідеї проектування. Здійснено аналіз вхідних даних, обґрунтування і синтез структурної схеми. Детально розглянуто складові апаратної частини їх призначення і особливості функціонування. Проаналізовано документацію, що описує роботу складових вузлів МКС. На основі цих даних описано та спроектовано принципову схему та алгоритм роботи.

Для побудови блок-схем, алгоритмів та принципових схем закріплено навички роботи з програмою sPlan. Для написання і владодження програмного забезпечення використано середовище розробки MPLAB 8.92. У цьому середовищі створено проект, написано текст програми з використанням мови C, і здійснено компіляцію у hex-файл. Середовище розробки може використовуватись разом з програматорами для запису готового вихідного коду прошивки у пам'ять програми вибраного мікроконтролера.

При написанні тексту програми сформовано значення регістра конфігурації за даними документації виробника, описано регістри спеціальних функцій та загального призначення.

Проведено ініціалізацію мікроконтролера, його вузлів та модулів.

Для забезпечення роботи периферійних пристроїв здійснено керування, прийом та передача даних з пристроїв введення та до пристроїв виведення згідно варіанту.

У робочій частині забезпечено функціонування МКС згідно варіанту.

Курсовий проект може бути основою розробки керуючої системи збору даних чи системи керування та контролю за виконанням завдань.

ПЕРЕЛІК ПОСИЛАНЬ

1. Войтович І. В., Сидоренко В. М. Мікропроцесорні системи та мікроконтролери. — Київ: КНТ, 2020.
2. Мазур О. М. Мікроконтролери AVR: програмування мовою С. — Львів: Львівська політехніка, 2018.
3. Погорілий Ю. С. Вбудовані системи на базі мікроконтролерів. — Київ: КПІ ім. І. Сікорського, 2021.
4. Barrett S. F., Pack D. J. Atmel AVR Microcontroller Primer: Programming and Interfacing. — Morgan & Claypool Publishers, 2019.
5. Han-Way Huang. The Atmel AVR Microcontroller: MEGA and XMEGA in Assembly and C. — Cengage Learning, 2017.
6. Jack Ganssle. The Art of Designing Embedded Systems. — Newnes, 2020.
7. Barnett R. H., Cox S., O’Cull L. Embedded C Programming and the Atmel AVR. — Cengage Learning, 2018.
8. Martin Bates. PIC Microcontrollers: An Introduction to Microelectronics. — *Newnes*, 2022.