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: «**Проєктування систем керування та збору інформації на базі мікроконтролерів сімейства PIC16»**

Виконав студент IV курсу, групи ***КІ-41***
Спеціальності *123 Комп'ютерна інженерія*
Варіант № 25
Червінь Іван Ярославович
(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 **№ 25**

Червіню Івану Ярославовичу

(прізвище, ім'я та по батькові)

з навчальної дисципліни: МІКРОКОНТРОЛЕРИ ТА ЇХ ПРОГРАМУВАННЯ

Студент групи: **КІ-41**

1. Тема проекту: Система моніторингу температури.

2. Дата видачі завдання: ”22” вересня 2025 р

3. Термін здачі курсового проєкту: ”15” грудня 2025 р.

4. Вихідні дані до проєкту:

4.1 Пристрої введення інформації : T74, DS1307

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

4.4 Використовувані мікроконтролери: PIC16F886, PIC16F877 .4

4.5 Протокол обміну інформацією: SPI.

4.6 Пристрій виведення інформації: термінал ПК

4.7 Функції пристрою виведення інформації: індикація: 25.12.21; 09:45:15 t=21C

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 Обгрунтування 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 _____ **Іван ЧЕРВІНЬ**
(підпис) (імя та прізвище)

Керівник проекту _____ **Роман СТОЛЯРЧУК**
(підпис) (імя та прізвище)

РЕФЕРАТ

Текстова частина курсового проєкту: 36 сторінок, 16 рисунків, 1 таблиця та 6 посилань.

Об’єкт проєктування - “Система моніторингу температури”.

Мета проєкту - довершити свої навики у розробці структурної схеми, електричної принципової схеми МКС, створення проекту 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, Т74, DS1307, SPI, термінал ПК, 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

Типовими представниками RISC-процесорів є PIC-контролери (Periferial 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 кілослів 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 двонапрямлений 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 проектуються, 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 рішенню 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 PIC16F886, PIC16F877.

Робота приладу забезпечується послідовним цифровим термодавачем T74 та модулем реального часу DS1307.

Вивід інформації здійснюватиметься через термінал ПК, що показуватиме індикацію.

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 контролю температури зображено графічний вивід інформації, що буде мати наступний вигляд: 25.12.21; 09:45:15 t=21C.

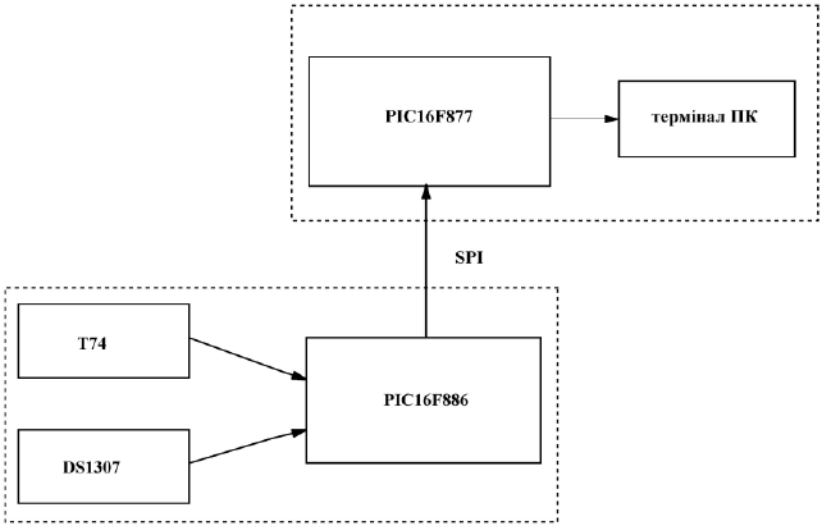

Рисунок 1.1 - Структурна схема Системи моніторингу температури

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 - звузити 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 мікро- контролер 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 PIC16F886 та PIC16F877 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 Послідовний цифровий термодавач T74

TC74 є серійно доступним цифровим датчиком температури, який особливо підходить для недорогих і малих додатків. Дані про температуру перетворюються з вбудованого термочутливого елемента і стають доступними у вигляді 8-бітового цифрового слова. Зв'язок з TC74 здійснюється через 2-провідний послідовний порт, сумісний з SMBus/I²C. Цю шину також можна використовувати для моніторингу кількох точок/багато зон. Біт SHDN в регістрі CONFIG можна використовувати для активації режиму очікування з низьким енергоспоживанням. Температурна роздільна здатність становить 1°C. Швидкість перетворення становить номінальну 8 вибірок/сек. Під час нормальної роботи струм спокою становить 200 мкА (типовий). Під час роботи в режимі очікування струм спокою становить 5 мкА (типовий). Невеликі розміри, низька вартість встановлення та простота використання роблять TC74 ідеальним вибором для впровадження терморегулювання в різноманітних системах.

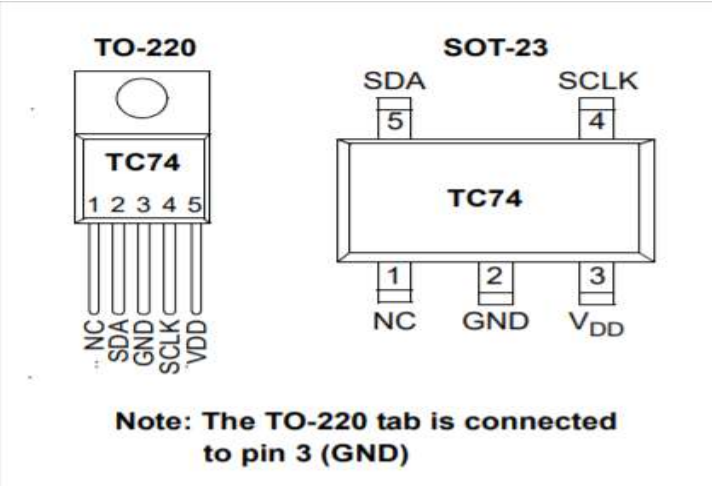

Рисунок 1.2 – Структурна схема T74

Технічні характеристики:

- Точність - найвища (найнижча): ±2°C (±3°C)
- Особливості: Режим очікування
- Робоча температура: -40°C ~ 125°C
- Тип виходу: I²C/SMBus
- Роздільна здатність: 7 б
- Температура сприйняття - місцева: -40°C ~ 125°C
- Тип датчика: цифровий, локальний
- Умови випробування: 25°C ~ 85°C (0°C ~ 125°C)
- Напруга - живлення: 2,7 В ~ 5,5 В

1.3.2 Модуль RTCC на основі мікросхеми 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.

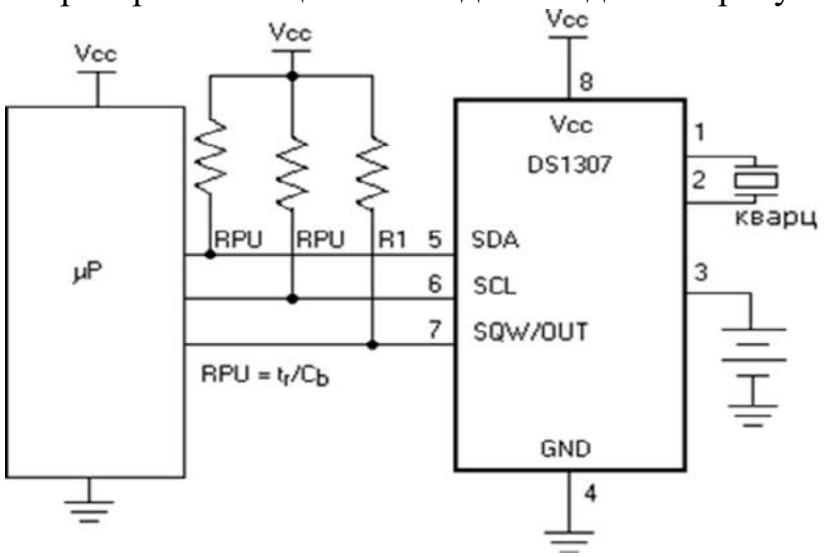

Рисунок 1.3 - Типова схема підключення модуля DS1307 до мікроконтролера

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 (настроєний 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, налаштовуючи 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 1.4.

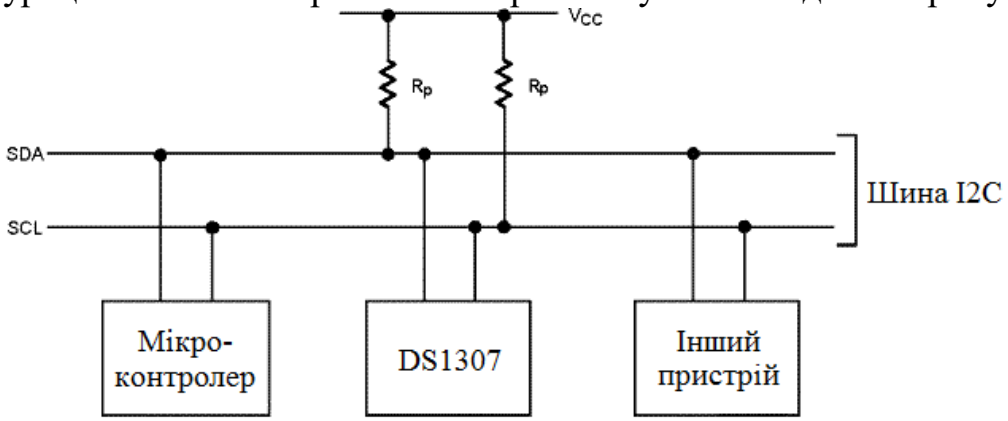

Рисунок 1.4 - Типова конфігурація двопровідної шини (I2C)

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.

Нижче описано процес передачі даних по двопровідній шині. Основні принципи передачі даних по шині I2C:

- передача даних може бути ініційована тільки тоді, коли шина вільна;
- під час передачі дані на лінії SDA можуть змінюватись тільки тоді, коли на лінії SCL низький рівень, інакше - зміна даних інтерпретуватиметься, як сигнал керування.

Таким чином, можливі наступні стани шини:

- «Шина не зайнята» –на лініях SDA і SCL зберігається високий рівень;
- «Початок передачі даних» (умова START*)* –зміна стану лінії SDA звисокого рівня на низький, у той час як на лінії SCL високий рівень;
- «Завершення передачі даних» (умова STOP) –зміна стану лінії SDA з низького рівня на високий, в той час як на лінії SCL високий рівень;
- «Коректні дані» – стан лінії SDA являє собою коректні дані, якщо після умови START стан лінії SDA не міняється на протязі високого рівня тактового сигналу. Дані на лінії повинні мінятися впродовж періоду низького рівня тактового сигналу. На один біт даних приходиться один тактовий імпульс;
- «Підтвердження» – кожен адресований приймаючий пристрій зобов‘язаний генерувати підтвердження після прийому кожного байту. Ведучий пристрій зобов‘язаний генерувати додатковий тактовий імпульс, який призначений для біта підтвердження.

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 побайтово, і кожний байт приймач підтверджує дев‘ятим бітом (біт підтвердження - ACK). В специфікації двох провідного інтерфейсу визначені звичайний режим (с тактовою частотою 100 кГц) і швидкий режим (с тактовою частотою 400 кГц). DS1307 працює тільки в звичайному режимі (100 кГц).

- R/-W – біт читання/запис або біт напряму передачі

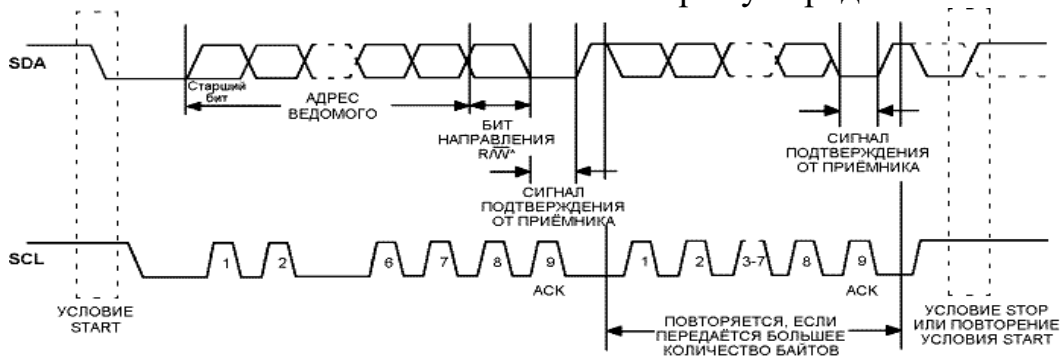

Рисунок 1.5 - Передача даних по двохнаправленій шині (I2C)

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.

В залежності від стану біта R/-W можливі два типи передачі даних:

- передача даних від ведучого передавача до веденого приймача. (Перший байт, який передається ведучим, - це адреса веденого

пристрою, з яким здійснюватиметься обмін інформацією. Далі передаються безпосередньо байти даних. Ведений повертає біт підтвердження після кожного прийнятого байту. Дані передаються, починаючи з старшого біта (most significant bit - MSB);

- дані передаються від веденого передавача до ведучого приймача. (Перший байт (адрес веденого) передається ведучим. Ведений повертає біт підтвердження, а за ним певну кількість байтів даних. Ведучий повертає біт підтвердження після всіх прийнятих байтів, крім останнього. В кінці останнього прийнятого байту повертається "не підтверджено" (NACK). Ведучий пристрій генерує всі послідовні синхроімпульси, а також умови START і STOP. Передача завершується умовою STOP або повтором умови START. Так як повтор умови START також є початком наступної послідовної передачі, то в такому випадку шина не вивільняється. Дані передаються, починаючи з старшого біту).

DS1307 може працювати в наступних двох режимах:

- режим веденого приймача(режим запису DS1307);
- режим ведучого передавача (режим читання DS1307).

У режимі веденого приймача послідовні дані приймаються по SDA і синхронізуються по SCL. Після кожного прийнятого байту передається біт підтвердженя. Умови START і STOP розпізнаються як початок і кінець послідовної передачі. Розпізнавання адреси виконується апаратно після прийому адреси веденого ібіта напряму передачі (див. рисунок 1.6).

Рисунок 1.6 - Режим веденого приймача(режим запису DS1307).

У режимі ведучого передавача (режим читання DS1307) перший байт приймається і опрацьовується так само, як і в режимі веденого приймача. Але в даному режимі біт напряму буде вказувати, що напрям передачі протилежний. DS1307 передає послідовні дані по лінії SDA поки на лінію SCL надходять послідовні імпульси. Умови START і STOP розпізнаються в початку і в кінці послідовної передачі (див. рисунок 1.7).

Рисунок 1.7 -Режим ведучого передавача (режим читання DS1307)

Байт адреси–перший байт, який приймається після умови START, генерується ведучим пристроєм. Байт адреси формується з 7-бітної адреси DS1307, яка відповідає послідовності 1101000, і наступного біта напряму передачі (R/-W), який для читання рівний 1. Після прийому і декодування байта адреси, пристрій видає підтвердження на лінію SDA.

1.3.5 Віртуальний термінал

Віртуальний Термінал VSM дає можливість використовувати клавіатуру й екран персонального комп'ютера, щоб посилати й приймати по послідовному асинхронному інтерфейсу RS232 дані, що надходять до/від імітованої мікропроцесорної системи. Це особливо корисно при налагодженні, де можна відобразити відлагоджувальні повідомлення, що сформовані програмним забезпеченням, яке розробляється.

Віртуальний термінал має наступні характеристики:

- повний дуплекс - послідовні вхідні дані відображаються, як символи ASCII; - простий двопровідний інтерфейс даних: RXD для отриманих даних і TXD для передачі даних;
- простий двопровідний апаратний інтерфейс встановлення зв'язку: RTS (ready-to-send) - готовий до передачі, і CTS (clear-to-send) - очищений для передачі;
- швидкість від 300 до 57 600 бод;
- 7 або 8 інформаційних розрядів;
- парність, непарність;
- 0, 1 або 2 стопових біта;
- процедура встановлення зв'язку програмного забезпечення XON/XOFF на додаток до апаратної процедури встановлення зв'язку;
- нормальна або негативна полярність сигналів і для RX/TX, і для RTS/CTS.

Щоб підключити термінал до моделі необхідно:

- вибрати значок мультиметра на правій панелі, вибрати Віртуальний Термінал (VIRTUAL TERMINAL) і розмістити його на схемі.
- підключити виводи RX і TX до передавальної й приймальніої ліній тестованої системи. RX - вхід, TX – вихід;
- якщо цільова система використовує апаратну процедуру встановлення зв'язку, то необхідно підключити виводи RTS і CTS до відповідних до шин керування потоком. відредагувати налаштування віртуального термінала: швидкість у бодах, довжину слова, парність, керування потоком даних і полярності сигналів;
- запустити моделювання звичайним способом. Термінал відобразить вхідні дані, як тільки одержить їх; щоб посилати символи в систему, необхідно

впевнитися, що фокус вказівника у вікні термінала. Друкуйте необхідний текст на клавіатурі персонального комп'ютера.

- як тільки почнеться моделювання, подальші функціональні можливості будуть доступні в контекстному меню, яке викликається правою кнопкою миші на вікні термінала. Це меню дає можливість призупинити дисплей, і копіювати й вставляти текст у буфер обміну й з нього.

Особливості використання віртуального термінала:

- віртуальний термінал підтримує керуючі ASCII коди CR (0x0D) (повернення каретки), BS (0x08) (повернення) і BEL (0x07) (звуковий сигнал). Усі інші коди ігноруються!!! - віртуальний термінал - це просто цифрова модель, і також не вимагає ніяких специфічних рівнів напруги на своїх виводах. Він буде нормально працювати як безпосередньо з мікропроцесором або іншим універсальним асинхронним приемопередатчиком, так і з обладнанням RS232 драйвером типу MAX232, який містить логічні інвертори.

- рівень активного стану на виводах RX і TX за замовчуванням високий. У такий спосіб стан холостого ходу - високий рівень, стартовий біт - низький рівень, а стоповий біт - високий рівень. Інформаційні розряди з'являються як високий рівень при '1' і низький рівень при '0'. Це сумісно із вбудованими модулями UART у багатьох мікроконтролерах. В іншому випадку (наприклад, щоб підключити термінал на вихід RS232 драйвера), необхідно встановити негативну полярність RX/TX.

- за замовчуванням, термінал не відображає символи, що друкуються (echo (Echo Typed Characters) їх можна включити у контекстному меню.

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 електричні 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).

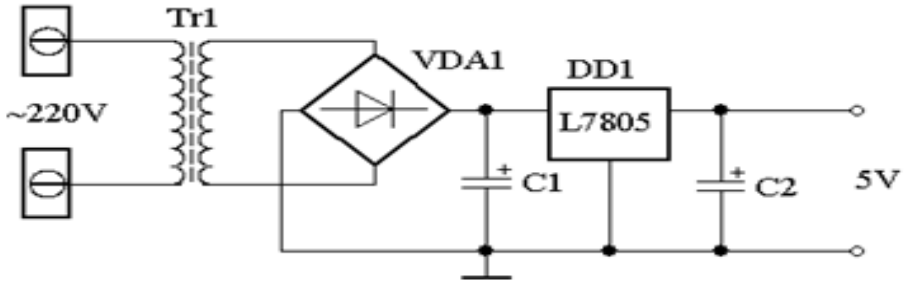

Рисунок 2.1- Схема стабілізації живлення для цифрових пристроїв на основі мікроконтролерів

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.

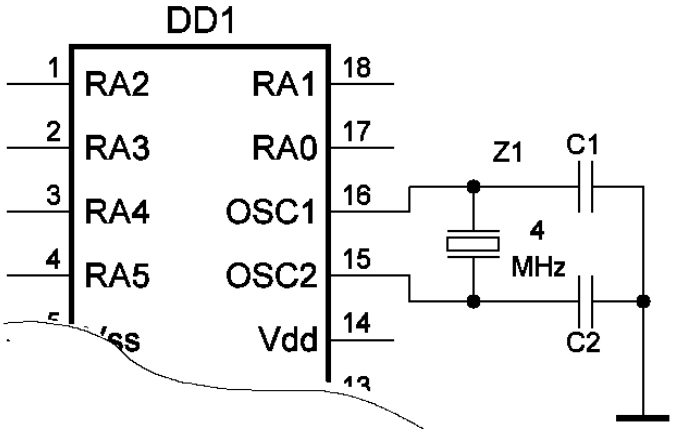

Рисунок 2.2 - Схема тактування мікроконтролера

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 Vpp. Задіюється механізму доступу до пам‘яті мікроконтролера. Дані передаються по лінії Data на вивід PGD мікроконтролера, при цьому з лінії Clock роз‘єму ISCP 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 проектованого 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 виконувані 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, ІЧ-датчики, тощо… В свою чергу пристрої виводу також не обмежуються монітором. Для МКС це, – як правило, – знакосимвольні 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‘єднюються 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.

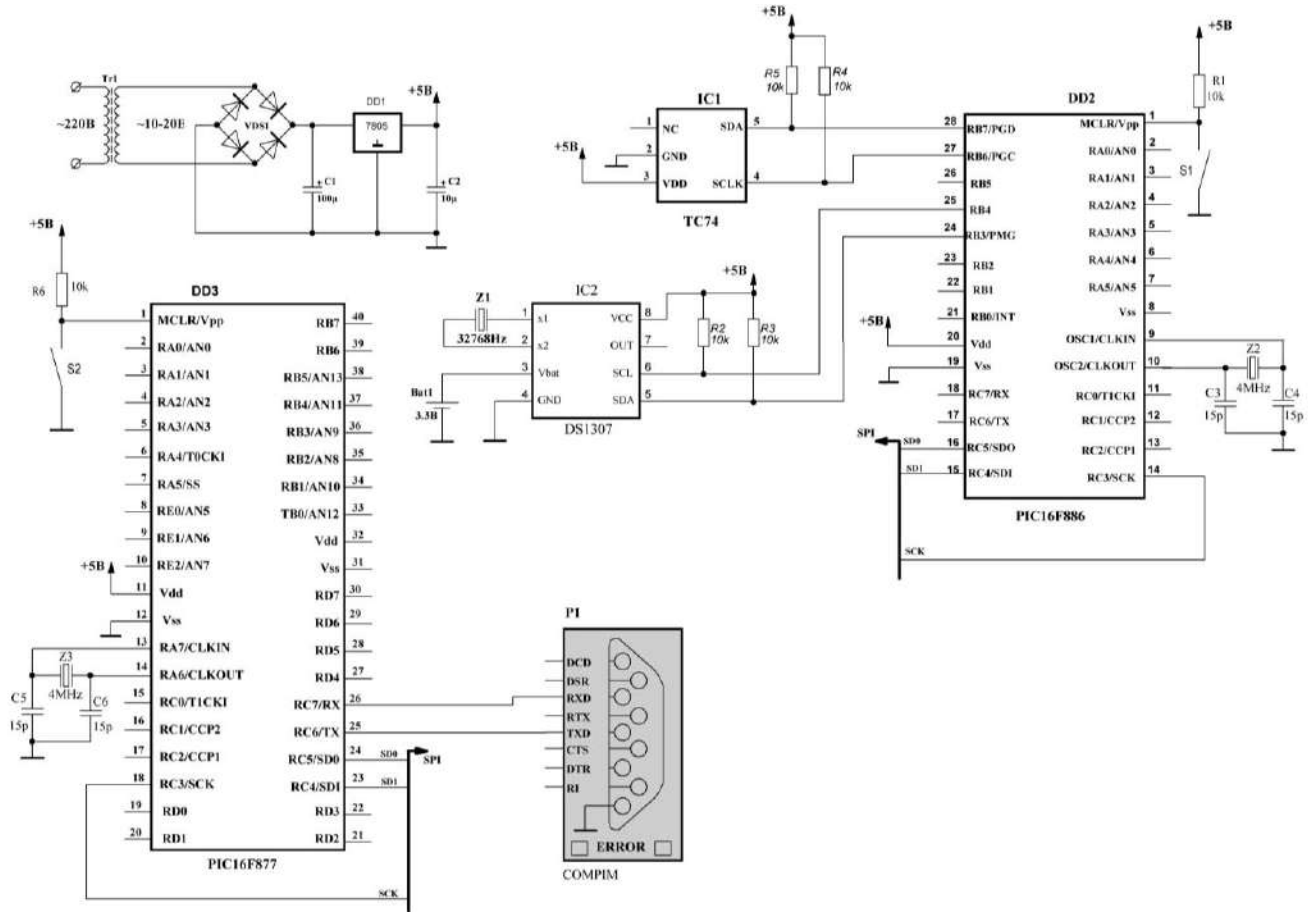

Рисунок 2.3 - Електрична принципова схема роботи Системи моніторингу температури

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, Z3	4МГц
5	Конденсатор	C1	100мкФ
6	Конденсатор	C2	10мкФ
7	Конденсатори	C3, C4, C5, C6	15рФ
8	Резистори	R1,R2, R3, R6	10кОм
9	Резистори	R4, R5	5кОм
10	Мікроконтролери	DD2,DD3	PIC16F886, PIC16F877
11	Цифровий термодавач	IC1	T74
12	Модуль реального часу	IC2	DS1307
12	Термінал ПК	P1	COMPIM
13	Кнопка-перемикач	S1,S2	SK1-53
14	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 трасирувальник 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.

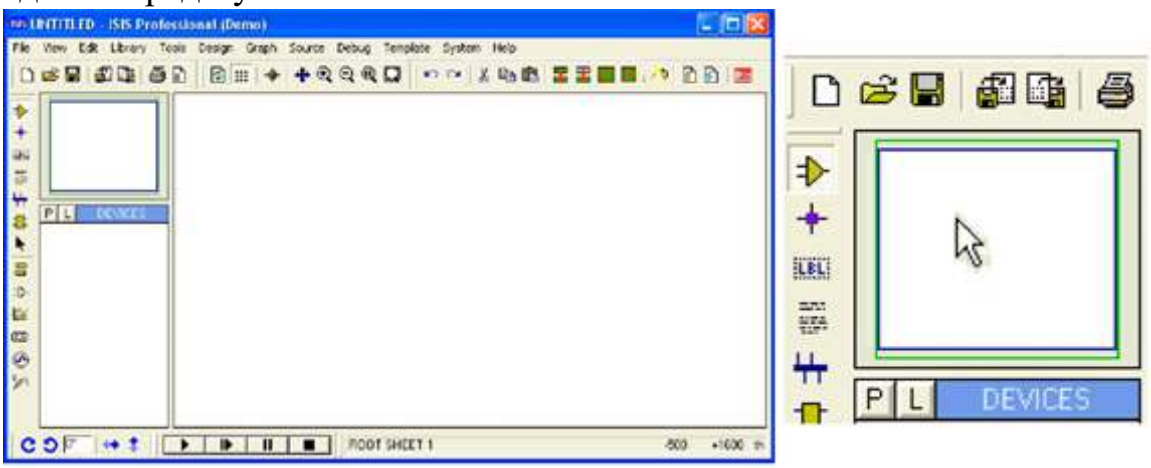

Рисунок 2.4 - Основне вікно програми ISIS 7 Professional і вікно попереднього перегляду.

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'явитися 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.

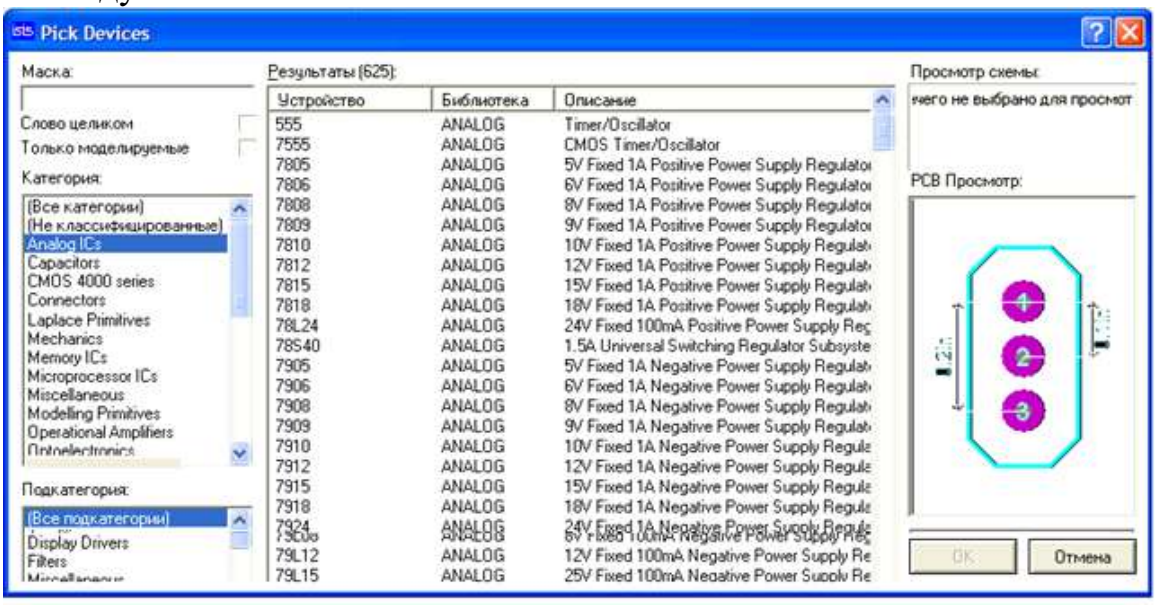

Рисунок 2.6 - Бібліотека електронних компонентів.

Для того, щоб переглянути роботу схеми, необхідний вихідний HEX-файл. Середовище PROTEUS підтримує багато засобів розробки, серед них і HI-TECH Сі 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.

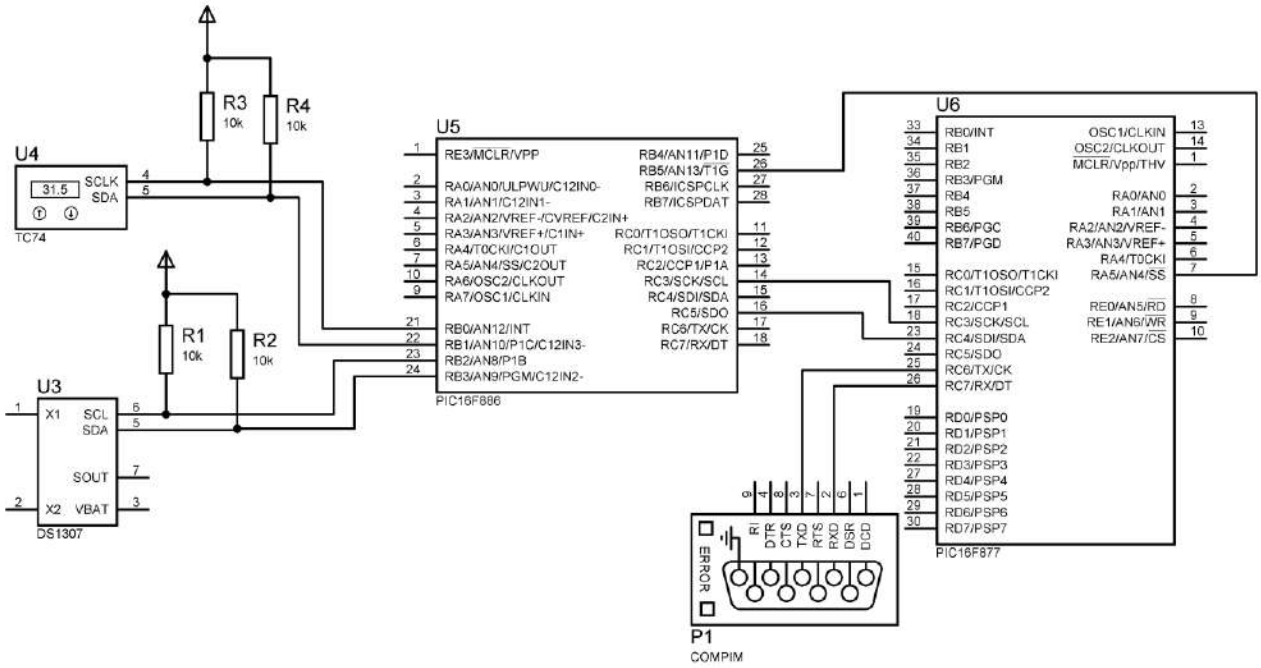

Рисунок 2.7 - Досліджувана схема

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.

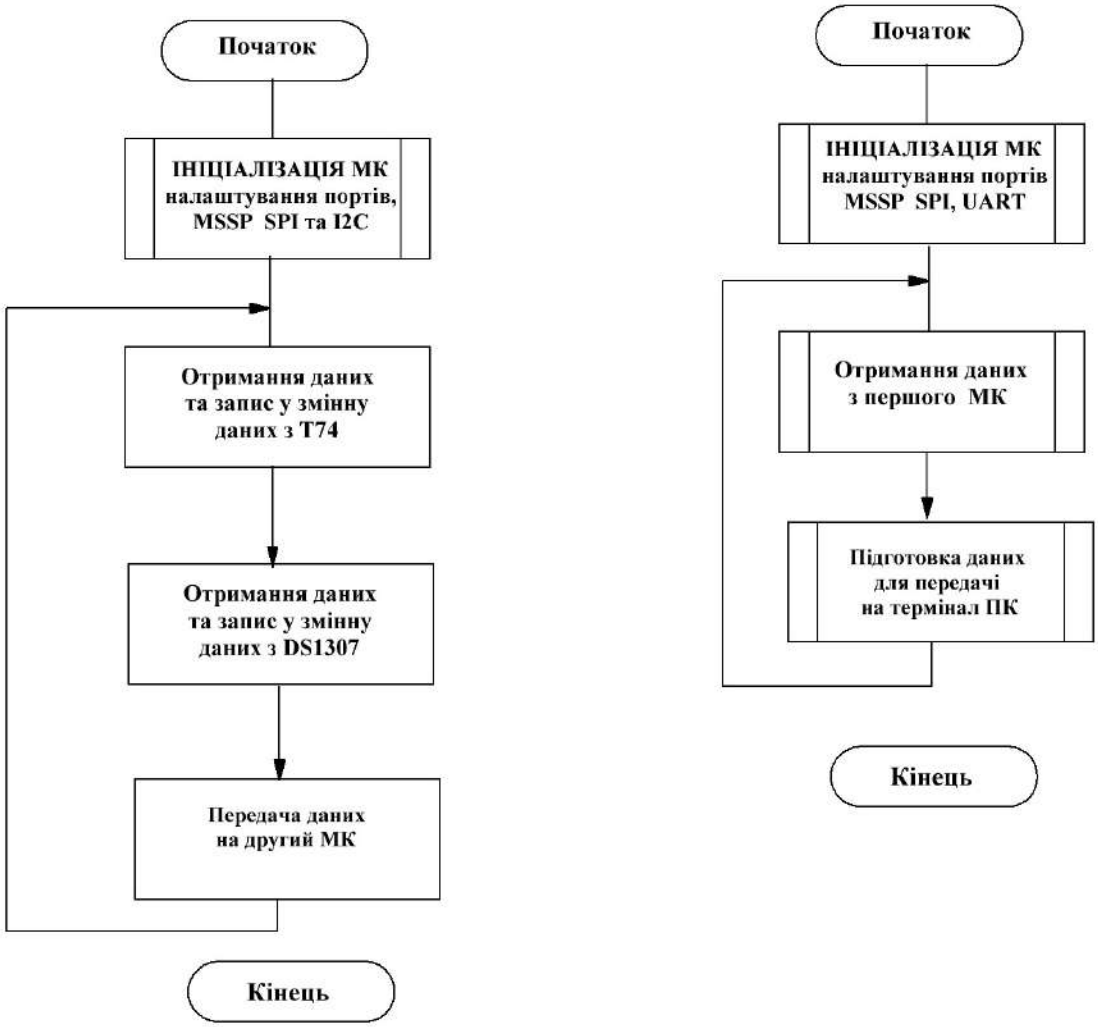

Рисунок 3.1 - Блок-схеми роботи Системи моніторингу температури

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 PIC16F886 з використанням мови С:

```
__CONFIG (FOSC_XT & WDTE_OFF & PWRTE_OFF& MCLRE_ON &
CP_OFF& CPD_OFF & BOREN_OFF & IESO_ON & FCMEN_OFF& LVP_ON&
DEBUG_OFF );
__CONFIG(BOR4V_BOR40V&WRT_OFF);
```

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

```
__CONFIG(FOSC_XT&WDTE_OFF&PWRTE_OFF&CP_OFF&BOREN_OFF&L
VP_OFF& CPD_OFF & WRT_OFF &DEBUG_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 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 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 T74 та DS1307, а також обробку та передачу даних на другий МК.

Наведений нижче модуль програми здійснює ініціалізацію портів мікроконтролера PIC16F886 модуля SPI та I2C:

```
#include "MK_init.h"
//---- опис функцій ----------------------
void MK_init(void)
{
//--Налаштування портів--//
TRISA=1; //порт, який не використовується
TRISC=0b11010111;/*4 та 6 - виводи передачі даних на другий МК за протоколом SPI*/
TRISB=0b00101111;/*1 та 2 - виводи прийому даних з T74, 3 та 4 - виводи прийому даних з DS1307, а 6 - керуючий вивід SPI*/

//----------- MSSP SPI -----------
//--- SSPSTAT
SMP=0;// опитування входу в середині періоду виведення даних
CKE=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;

//----------- MSSP I2C --------------------
                SSPADD=9;
                SMP=1;
                CKE=0;
                SSPCON2=0;

                SSPM3=1;
                SSPM2=0;
                SSPM1=0;
                SSPM0=0;
                SSPEN=1;
        }
```

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

```
        //---підключення описових файлів---//
        #include "MK_init.h"
        #include "i2c_p.h"
        #include "i2c_p2.h"
        //---оголошення глобальних змінних---//
//змінні часу
        unsigned char sec;
        unsigned char min;
        unsigned char hrn;
//змінні дати
        unsigned char day;
        unsigned char mis;
        unsigned char rik;
//змінні температури
        unsigned int T;

//---- оголошення функцій для передачі даних ----------------
        void spi_trans(unsigned char dat);
        //---- опис функцій ----------------------
        void spi_trans(unsigned char dat)
        {
        RB5=0;
        SSPBUF=dat;
        while(BF==0){}
        RB5=1;
        }
```

```
//---- ОСНОВНА ПРОГРАМА ------------------
void main(void)
{
MK_init();//функція ініціалізації МК


//---- РОБОЧИЙ ЦИКЛ ------------------
while(1)
{

//--------- читання часу -----------------
i2c_start_p();//запуск роботи з DS1307
i2c_trans_p(0b11010000);//режим запису у пам’ять пристрою
i2c_trans_p(0x00);//встановлення початку читання в нульову позицію
i2c_stop_p();//зупинка роботи DS1307

i2c_start_p();//повторний старт роботи з DS1307
i2c_trans_p(0b11010001);// режим читання даних з пам’яті пристрою
//запис зчитаних даних у змінні:
sec=i2c_recieve_ack_p();
min=i2c_recieve_ack_p();
hrn=i2c_recieve_nack_p();
i2c_stop_p(); //зупинка роботи з DS1307
//передача зчитаних даних на другий МК:
spi_trans(sec);
spi_trans(min);
spi_trans(hrn);

//--------- читання дати -----------------
i2c_start_p();//запуск роботи з DS1307
i2c_trans_p(0b11010000);//режим запису у пам’ять пристрою
i2c_trans_p(0x04);//встановлення початку читання з четвертої позиції
i2c_stop_p();//зупинка роботи з DS1307

i2c_start_p(); //повторний старт роботи з DS1307
i2c_trans_p(0b11010001);//// режим читання даних з пам’яті пристрою
//запис зчитаних даних у змінні:
day=i2c_recieve_ack_p();
mis=i2c_recieve_ack_p();
rik=i2c_recieve_nack_p();
i2c_stop_p(); //зупинка роботи з DS1307
//передача зчитаних даних на другий МК:
spi_trans(day);
spi_trans(mis);
spi_trans(rik);

//----------- читання температури ----------
```

```
i2c_start2_p();
i2c_trans2_p(0b10011010);   // 1001101* - A5 (tranceive)
i2c_trans2_p(0x01);         // CONFIG-reg
i2c_trans2_p(0b01000000);   // запуск конвертації ADC
i2c_stop2_p();
__delay_ms(200);

//------- читання результатів вимірювання -----------
i2c_start2_p();
i2c_trans2_p(0b10011010);   // 1001101* - A5
i2c_trans2_p(0x00);                 //TEMP-reg

i2c_start2_p();                     // повторний СТАРТ
i2c_trans2_p(0b10011011);   //зміна напряму 1001101+1 (receive)
T=i2c_recieve_nack2_p();    // читання інформаційного байту температури
i2c_stop2_p();

spi_trans(T);//передача зчитаного значення на другий МК
}
}
```

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

Наведений нижче модуль програми здійснює ініціалізацію портів мікроконтролера PIC16F877 та модулі SPI, UART:

```
#include "MK_init.h"

//---- опис функцій ----------------------
void MK_init(void)
{
//---------- Налаштування портів ---------------//
TRISA5=1; //порт керування SPI
TRISC=0b00111000;/*4 та 5 - виводи отримання даних по SPI від першого МК, а 7 та 8 - виводи передачі даних для виведення на термінал ПК*/
TRISB=0;//порт не використовується
TRISD=0; //порт не використовується
```

Наступний фрагмент коду здійснює ініціалізацію регістрів модуля SPI:

```
//---------- MSSP SPI -------------------
//--- SSPSTAT
SMP=0;// опитування входу в середині періоду виведення даних
CKE=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:
```

```
        // ведений режим SPI, тактовий сигнал = Fosc/4
                SSPM3=0;
                SSPM2=1;
                SSPM1=0;
                SSPM0=0;
                SSPEN=1;
                GIE=0;
        RS=0;
        RW=0;
        E=0;

        //------UART----------
        TXREG=0;//регістр прийому
        RCREG=0;//регістр прийому
        //-----SPBRG-----------
        SPBRG=25;
        //------TXSTA---------
        BRGH=1;
        SYNC=0;//асинхроний режим
        TX9=0;// робота 9-ти бітної посилки
        TXEN=1;//дозвіл включення передавача
        //------RCSTA--------------
        RX9=0;//дозвіл роботи 9-ти бітрної посилки
        CREN=1;//дозвіл включення передавача
        SPEN=1;//включення модуля UART
        GIE=0;
        }
```

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

```
//оголошуємо та описуємо функцію прийому даних від першого МК за даним протоколом:
        unsigned char spi_receive(void);
//---- опис функцій ----------------------
        unsigned char spi_receive(void)
        {
        while(BF==0){}
        return SSPBUF;
        }

//---- ОСНОВНА ПРОГРАМА -------------------
void main(void){
MK_init(); //функція ініціалізації МК


printf("\033[2J"); // очищаємо весь екран
printf("\033[0m"); // скидаємо усі атрибути
printf("\033[1;1f"); // переміщаємо курсор в 1 рядок 1 стовпця
printf("\033[7m");// встановлюємо режим "реверс"
```

```
printf("\033[33m");// встановлюємо колір тла

//Виводимо рамку-заготовку за допомогою printf
printf("---------------------\r\n");
printf("|     MONITORYNG    |\r\n");
printf("---------- ----------\r\n");
printf("|  Chas ||          |\r\n");
printf("---------- ----------\r\n");
printf("|  Data ||          |\r\n");
printf("---------------------\r\n");
printf("|  Temp ||          |\r\n");
printf("---------------------\r\n");

printf("\033[0m"); // після чого знову скидаємо усі атрибути
printf("\033[7m"); //та  встановлюємо режим "реверс"

//---- РОБОЧИЙ ЦИКЛ ------------------

while(1){

//За допомогою описаної вище функції отримуємо дані з першого МК
//дані часу
sec=spi_receive();
min=spi_receive();
hrn=spi_receive();

                    //дані дати
rik=spi_receive();
mis=spi_receive();
day=spi_receive();

//дані температури
T=spi_receive();

printf("\033[4;13f"); //встановлюємо курсор в позицію 4 рядка; 13 стовпця
printf("%d:%d:%d",hrn,min,sec); //та виводимо значення години

printf("\033[6;11f"); //встановлюємо курсор в позицію 6 рядка; 11 стовпця
printf("%d.%d.%d", day,mis,rik); //та виводимо значення дати

printf("\033[8;15f"); //встановлюємо курсор в позицію 8 рядка; 15 стовпця
printf("%d C",T); //та виводимо значення температури
}
}
```

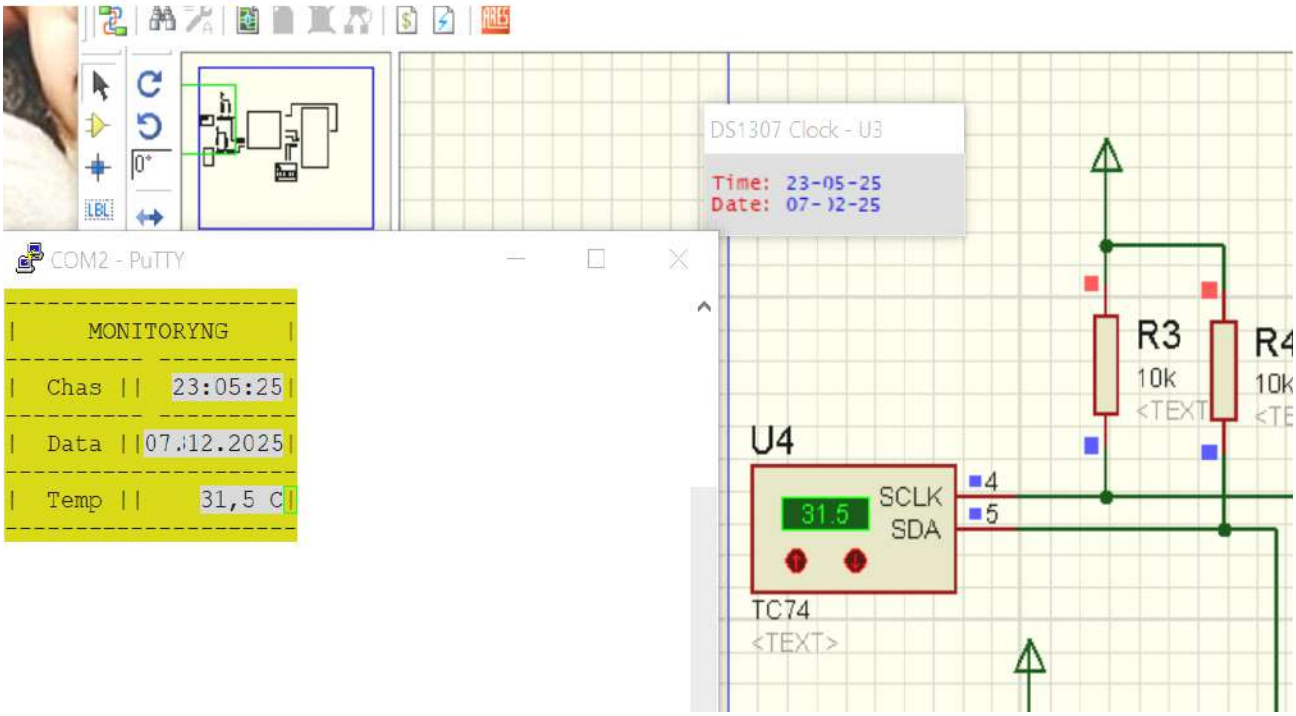

Рисунок 3.3 - Робоча схеми в PROTEUS в режимі включеного реле

ВИСНОВКИ

В процесі курсового проєктування розглянуто принципи проектування пристроїв і систем, що побудовані з використанням однокристальних мікроконтролерів фірми Microchip. Під час виконання проєктування 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 влагодження програмного забезпечення використано середовище розробки MPLAB 8.92. У цьому середовищі створено проект, написано текст програми з використанням мови С, 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 проєкт 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. Martin Bates. PIC Microcontrollers: An Introduction to Microelectronics. — Newnes, 2022.