ВІДЗИВ НАУКОВОГО КЕРІВНИКА НА ВИПУСКНУ КВАЛІФІКАЦІЙНУ РОБОТУ

Студента Андрієвича Владислава Миколайовича
представлену на здобуття освітнього ступеня бакалавр
зі спеціальності 123 «Комп'ютерна інженерія»

1. Робота виконана на тему „Проект підсистеми єдиного часу для мережі розумного будинку на базі платформи Arduino".

2. Актуальність роботи зумовлена розвитком технології «розумного будинку» та потребою в системах єдиного часу для синхронізації пристроїв мережі.

3. Основний зміст пояснювальної записки роботи викладено на 55 сторінках тексту, записка містить 58 рисунків, 7 сторінок додатків. Список використаних літературних джерел налічує 23 найменування.

4. Пояснювальна записка складається з реферату, переліку умовних скорочень, вступу, трьох розділів, висновків, списку використаної літератури, додатків.

В першому розділі розглядається теоретичний матеріал стосовно розробки системи єдиного часу: які задачі вимагають використання системи єдиного часу; аналізуються існуючі системи; описуються способи синхронізації часу та структурна схема; розглядаються програмні та апаратні засоби для реалізації системи.

В другому розділі проводиться розробка системи єдиного часу: визначаються вузли структурної схеми та джерело отримання корисної інформації про дату та час; розробляється алгоритм роботи системи єдиного часу та створюється функціональна схема; наводиться процес підключення елементів системи; розробляється програмний код мікроконтролера.

В третьому розділі описується, як працювати з системою єдиного часу, які налаштування потрібно проводити та як перевіряти працездатність системи; проводиться демонстрація роботи розробленої системи.

5. Висновки містять сумарні результати проведеної роботи.

6. Список використаних літературних джерел має достатній для даного виду робіт обсяг, є правильно оформленим та повністю опрацьованим у ході виконання роботи.

7. Додатки містять лістинг програмного коду мікроконтролера

8. Рисунки оформлені досить якісно, є інформативними та наведені у відповідних місцях роботи.

9. Якість оформлення роботи є високою.

10. Виявлені на етапі попереднього захисту недоліки оформлення пояснювальної записки та презентації усунені. Рекомендації стосовно покращення проекту враховані.

11. Ставлення студента до виконання випускної кваліфікаційної роботи – відповідальне.

12. Дипломна робота студента Андрієвича В.М. повністю відповідає поставленому завданню. Всі рішення по обґрунтуванню та розробці системи єдиного часу для мережі розумного будинку на базі платформи Arduino достатньо пророблені та обґрунтовані. Рівень наукової та технічної підготовки студента Андрієвича В.М. достатньо високий. Під час виконання завдання студент Андрієвич В.М. показав вміння самостійно розробляти алгоритми та програми, проводити налаштування, аналізувати отримані результати, робити правильні висновки.

Випускна кваліфікаційна робота студента Андрієвича В.М. написана на високому рівні та може бути допущена до захисту в Екзаменаційній комісії.

Пропонована оцінка – «відмінно».

Науковий керівник
старший викладач кафедри КІ та КБ — В.П. Сіденко
1 червня 2020 року

РЕЦЕНЗІЯ

на кваліфікаційну роботу
студента 4 курсу групи КІ-2 спеціальності 123 «Комп'ютерна інженерія»
факультету інформаційно-комп'ютерних технологій
Державного університету «Житомирська політехніка»
Андрієвича Владислава Миколайовича
на тему: „Проект підсистеми єдиного часу для мережі розумного будинку на базі платформи Arduino"

Кваліфікаційна робота студента Андрієвича В.М. „Проект підсистеми єдиного часу для мережі розумного будинку на базі платформи Arduino" присвячена актуальній тематиці – розробці системи єдиного часу для синхронізації пристроїв в мережі.

Робота містить апаратно реалізований проект підсистеми єдиного часу на базі платформи Arduino та пояснювальну записку наступного складу: реферат, перелік умовних скорочень, вступ, три розділи основного тексту, висновки, список використаної літератури, додатки.

Розділ 1 присвячений роботі з теоретичним матеріалом стосовно розробки системи єдиного часу: розглядаються задачі, які вимагають використання системи єдиного часу; аналізуються існуючі системи; описуються способи синхронізації часу та структурна схема; розглядаються програмні та апаратні засоби для реалізації системи.

Розділ 2 присвячений розробці системи єдиного часу: визначаються вузли структурної схеми та джерело отримання корисної інформації про дату та час; розробляється алгоритм роботи системи єдиного часу та створюється функціональна схема; наводиться процес підключення елементів системи; розробляється програмний код мікроконтролера.

Розділ 3 присвячений питанням роботи з системою, налаштуванням, які потрібно проводити та перевірці працездатності системи. Також, проводиться демонстрація роботи розробленої системи.

Зміст роботи відповідає поставленому завданню. У цілому кваліфікаційна робота виконана на високому рівні. Тема роботи розкрита повністю. Студентом продемонстровані ґрунтовні теоретичні знання та практичні фахові навички, що отримані як під час навчання, так і у процесі виконання роботи. Матеріал пояснювальної записки викладено логічно, науково і технічно грамотно. Рисунки, що наведений у роботі, оформлені якісно і наведені у відповідних місцях. Оформлення роботи у цілому відповідає вимогам стандартів. Проект підсистеми єдиного часу для мережі розумного будинку реалізовано на базі платформи Arduino. Його функціонал відповідає заявленому.

Позитивною стороною є те, що отримані результати мають практичну цінність – розроблений проект можна використовувати в мережах розумних будинків, для синхронізації пристроїв. Недоліками роботи є наступні: наявність деяких відхилень від вимого щодо оформлення пояснювальної записки (наприклад, відсутні посилання на літературні джерела у розділі 1). Зазначені недоліки у цілому не впливають на високу якість і актуальність роботи.

У цілому кваліфікаційна робота студента Андрієвича В.М. „Проект підсистеми єдиного часу для мережі розумного будинку на базі платформи Arduino" відповідає вимогам рівня вищої освіти «бакалавр» спеціальності 123 «Комп'ютерна інженерія» і рекомендується до захисту в ЕК.

Пропонована оцінка роботи – «відмінно».

Рецензент,
старший викладач кафедри БІ та Т — Д.С. Морозов
10 червня 2020 р.

Міністерство освіти і науки України
Державний університет «Житомирська політехніка»
Факультет інформаційно-комп'ютерних технологій
Кафедра комп'ютерної інженерії та кібербезпеки

ПОЯСНЮВАЛЬНА ЗАПИСКА

до випускної кваліфікаційної роботи бакалавра
на тему:

«Проект підсистеми єдиного часу для мережі розумного будинку на базі платформи Arduino»

Виконав студент 4-го курсу, групи КІ-2
спеціальності 123 «Комп'ютерна інженерія»
_____ Андрієвич В.М.
Керівник, старший викладач кафедри КІ та КБ
_____ Сіденко В.П.
Рецензент, старший викладач кафедри БІ та Т
_____ Морозов Д.С.

Житомир 2020

ДЕРЖАВНИЙ УНІВЕРСИТЕТ «ЖИТОМИРСЬКА ПОЛІТЕХНІКА»
ФАКУЛЬТЕТ ІНФОРМАЦІЙНО-КОМП'ЮТЕРНИХ ТЕХНОЛОГІЙ
КАФЕДРА КОМП'ЮТЕРНОЇ ІНЖЕНЕРІЇ ТА КІБЕРБЕЗПЕКИ
СПЕЦІАЛЬНІСТЬ 123 «КОМП'ЮТЕРНА ІНЖЕНЕРІЯ»

ЗАТВЕРДЖУЮ
Завідувач кафедри
комп'ютерної інженерії
та кібербезпеки
_____ Єфіменко А. А.
«06» квітня 2020 р.

ЗАВДАННЯ
на випускну кваліфікаційну роботу

Студента **Андрієвича Владислава Миколайовича**

Тема роботи: **Проект підсистеми єдиного часу для мережі розумного будинку на базі платформи Arduino**
затверджена Наказом по університету від **«06» квітня 2020 р. № 98с**
Термін здачі студентом закінченої роботи – **1 червня 2020 р.**
Вихідні дані для роботи:

Розробити підсистему єдиного часу для мережі розумного будинку на базі платформи Arduino.

Консультанти з випускної кваліфікаційної роботи із зазначенням розділів, що їх стосуються

Розділ	Консультант	Підпис, дата	
		Завдання видав	Завдання прийняв
1	Сіденко В.П. / Петросян Р.В.	06.04.20	06.04.20
2	Петросян Р.В.	20.04.20	20.04.20
3	Сіденко В.П. / Петросян Р.В.	11.05.20	11.05.20

Керівник _____ Сіденко В.П.

Календарний план

№ з/п	Назва етапів випускної кваліфікаційної роботи	Термін виконання етапів роботи	Примітка
1	Постановка завдання	6 квітня 2020 – 7 квітня 2020	Виконано
2	Пошук задач, які вимагають використання системи точного (єдиного) часу	8 квітня 2020 – 10 квітня 2020	Виконано
3	Пошук та аналіз існуючих системи точного (єдиного) часу	11 квітня 2020 – 15 квітня 2020	Виконано
4	Аналіз систем синхронізації часу (радіостанції, мережева синхронізація часу та синхронізація часу від орбітальних супутників)	16 квітня 2020 – 22 квітня 2020	Виконано
5	Аналіз структурної схеми системи єдиного часу	23 квітня 2020	Виконано
6	Пошук та аналіз програмних та апаратних засобів для реалізації системи	24 квітня 2020 – 29 квітня 2020	Виконано
7	Визначення вузлів структурної схеми системи	30 квітня 2020 – 1 травня 2020	Виконано
8	Розробка алгоритму роботи системи	2 травня 2020 – 7 травня 2020	Виконано
9	З'єднання (підключення) елементів системи єдиного часу	8 травня 2020	Виконано
10	Розробка програмного коду мікроконтролера	9 травня 2020 – 20 травня 2020	Виконано
11	Налаштування комп'ютера для роботи з розробленою системою єдиного часу	21 травня 2020– 22 травня 2020	Виконано
12	Написання пояснювальної записки	23 травня 2020– 29 травня 2020	Виконано
13	Підготовка презентації та виступу	30 травня 2020– 31 травня 2020	Виконано
14	Отримання відгуку керівника та рецензії на кваліфікаційну роботу	4 червня 2020 – 8 червня 2020	Виконано
15	Захист	13 червня 2020	Виконано

Студент _____ Андрієвич В.М.

Керівник _____ Сіденко В.П.

РЕФЕРАТ

Дипломна робота складається з переліку умовних скорочень, вступу, трьох розділів, висновків, списку використаних джерел, додатків і має 55 сторінок основного тексту, 58 рисунків, 7 сторінок додатків. Список використаних джерел містить 23 найменування і займає 3 сторінки. Загальний обсяг роботи – 65 сторінки.

Дипломна робота присвячена розробці підсистеми єдиного часу для мережі розумного будинку на базі платформи Arduino.

В першому розділі розглядається теоретичний матеріал стосовно розробки системи єдиного часу: які задачі вимагають використання системи єдиного часу; аналізуються існуючі системи; описуються способи синхронізації часу та структурна схема; розглядаються програмні та апаратні засоби для реалізації системи.

В другому розділі проводиться розробка системи єдиного часу: визначаються вузли структурної схеми та джерело отримання корисної інформації про дату та час; розробляється алгоритм роботи системи єдиного часу та створюється функціональна схема; наводиться процес підключення елементів системи; розробляється програмний код мікроконтролера.

В третьому розділі описується, як працювати з системою єдиного часу, які налаштування потрібно проводити та як перевіряти працездатність системи; проводиться демонстрація роботи розробленої системи.

У висновках підбито підсумки про проведену роботу.

У списку використаних джерел наведено список посилань, за якими знаходиться інформація, яка використовувалася під час аналізу теоретичної частини дипломної роботи.

Додатки до дипломної роботи містять розроблений програмний код для мікроконтролера системи єдиного часу.

ABSTRACT

Bachelor's graduate work consists of a list of abbreviations, introduction, three sections, conclusions, a list of sources, appendices and has 55 pages of main text, 58 figures, 7 pages of appendices. The list of sources contains 23 names and occupies 3 pages. The total volume of work is 65 pages.

Bachelor's graduate work is devoted to the development of a single time subsystem for a smart home network based on the Arduino platform.

The first section discusses the theoretical material on the development of a single time system: what tasks require the use of a single time system; existing systems are analyzed; methods of time synchronization and block diagram are described; software and hardware for system implementation are considered.

The second section develops a single time system: the nodes of the block diagram and the source of useful information about the date and time are determined; the algorithm of work of single time system is developed and the functional scheme is created; the process of connecting system elements is given; the program code of the microcontroller is developed.

The third section describes how to work with the system of a single time, what settings need to be made and how to check the system; a demonstration of the developed system is performed.

The conclusions summarize the work done.

The list of sources contains a list of links that contain information that was used in the analysis of the theoretical part of the bachelor's graduate work.

Appendices for the bachelor's graduate work contain the developed program code for the microcontroller of the single time system.

ЗМІСТ

ПЕРЕЛІК УМОВНИХ СКОРОЧЕНЬ 7
ВСТУП 8
РОЗДІЛ 1. АНАЛІЗ ІСНУЮЧИХ РІШЕНЬ СИСТЕМ ЄДИНОГО ЧАСУ 9
1.1. Перелік задач, які вимагають використання системи точного (єдиного) часу 9
1.2. Існуючі системи точного (єдиного) часу 10
1.2.1. Система єдиного часу «Тахіон» 10
1.2.2. Сервер часу TimeServer PLC2013 11
1.2.3. Сервер точного часу Sinhron-M500 12
1.3. Системи синхронізації часу 13
1.3.1. Радіостанції 13
1.3.2. Мережева синхронізація часу та синхронізація часу від орбітальних супутників 15
1.4. Структурна схема системи 19
1.5 Програмні та апаратні засоби для реалізації системи 20
1.5.1. Програмні засоби 20
1.5.2. Апаратні засоби 23
Висновки з першого розділу 28
РОЗДІЛ 2. ВИБІР ЕЛЕМЕНТНОЇ БАЗИ, РОЗРОБКА СИСТЕМИ ЄДИНОГО ЧАСУ 29
2.1. Визначення вузлів структурної схеми системи 29
2.2. Розробка алгоритму роботи системи єдиного час 30
2.3. З’єднання елементів системи єдиного часу 32
2.4. Розробка програмного коду мікроконтролера 36
Висновки з другого розділу 40
РОЗДІЛ 3. ОПИС РОБОТИ З СИСТЕМОЮ ЄДИНОГО ЧАСУ ТА ДЕМОНСТРАЦІЯ РОБОТИ СИСТЕМИ 42
3.1 Опис роботи з системою єдиного часу 42
3.2 Демонстрація роботи системи єдиного часу 47
Висновки з третього розділу 52
ВИСНОВКИ 53

СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ 54
ДОДАТКИ 57

ПЕРЕЛІК УМОВНИХ СКОРОЧЕНЬ

ARP – Address Resolution Protocol

COM (-port) – Communication port

DHCP – Dynamic Host Configuration Protocol

GCC – GNU Compiler Collection

GNSS – Global Navigation Satellite System

GPS – Global Positioning System

GLONASS (ГЛОНАСС) – Глобальна навігаційна супутникова система

IDE – Integrated development environment

IP – Internet Protocol

MAC – Media Access Control

MPIDE – Multi-Platform Integrated Development Environment

NTP – Network Time Protocol

PPS (1PPS) – Pulse per second

PTP – Precision Time Protocol

SNTP – Simple Network Time Protocol

TCP – Transmission Control Protocol

UDP – User Datagram Protocol

USB – Universal Serial Bus

UTC – Coordinated Universal Time

ШІМ – Широтно-імпульсна модуляція

ВСТУП

Час – одиниця виміру, якою користуються всі. Тому, майже будь-які розробки (в рамках розумного) заслуговують на свій коефіцієнт актуальності і можуть бути використані в різних сферах.

В період, коли інформаційно-комп'ютерні технології досягли досить високого рівня і продовжують розвиватися високими темпами, охопили всі сфери людського життя, системи точного (єдиного) часу стали необхідними в тих сферах, де необхідно мати прив'язку до точного (єдиного) часу (комунікації, безпека та оборона, наука, промисловість тощо).

Наприклад, єдиний час в промислових мережах, необхідний для узгодження роботи пристроїв і додатків, які здійснюють обробку даних в режимі реального часу; в системах моніторингу та управління з метою протоколювання подій та реагування на них тощо.

Стосовно розумних будинків, то з розвитком технологій вони набирають свою популярність серед людей, через свою новизну, цікавість та зручність. Система єдиного часу в розумному будинку може використовуватись в різних направленнях, наприклад, в протоколюванні подій в системі охорони периметру чи в керуванні системами енергоживлення будинку (включення або виключення світла) тощо.

Тому, тема розробки системи єдиного часу для мережі розумного будинку є досить актуальною і перспективною.

Метою роботи є розробити пристрій, який буде приймати запити на отримання інформації про дату та час, формувати відповідь та передавати її запитувачу.

Об'єктом та предметом дослідження є система точного (єдиного) часу.

РОЗДІЛ 1. АНАЛІЗ ІСНУЮЧИХ РІШЕНЬ СИСТЕМ ЄДИНОГО ЧАСУ

1.1. Перелік задач, які вимагають використання системи точного (єдиного) часу

Системи точного (єдиного) часу використовуються в різних сферах життя людей, де необхідно мати прив’язку до часу:

- промисловість
- комунікації
- наука
- безпека та оборона
- наземна та космічна навігація тощо.

Синхронізація часу в промислових мережах необхідна для узгодження роботи пристроїв і додатків, які здійснюють обробку даних в режимі реального часу. Крім цього, синхронізація потрібна в системах моніторингу та управління з метою протоколювання подій, що виникають, і своєчасного реагування на них. Однією з найбільш чутливих до точності часу систем є система автоматизації на енергетичних підстанціях. Сучасна система автоматизації підстанцій описана в стандарті IEC 61850, в якому зазначено, що точність синхронізації часу на шині процесів повинна бути не гірше 1 мікросекунди.

В науці прив’язка до точного часу дуже важлива при проведенні різних досліджень та експериментів, комунікації між різними науковими об’єктами.

Синхронізація часу в сфері комунікацій необхідна для коректної роботи всіх елементів, які забезпечують функціонування комунікаційної системи, крім того наявність впровадженої системи точного часу необхідна для зручності звичайних користувачі даної системи.

В безпеці та обороні при виявленні, протоколюванні, плануванні, проведенні різних операцій та інших видів дій точний час є необхідним, оскільки це сфера діяльності, яка пов’язана з безпекою людей та держави та нормальним їх функціонуванням.

1.2. Існуючі системи точного (єдиного) часу

На даний момент існує велика кількість систем точного часу, які відрізняються між собою ціною, технічними характеристиками, цільовими об’єктами та іншим. Нижче представлено декілька систем точного часу з технічними характеристиками.

1.2.1. Система єдиного часу «Тахіон»

Система єдиного часу «Тахіон» (рис 1.1) призначена для синхронізації технічних засобів управління і диспетчеризації в середовищі універсального координованого часу UTC.



Рис 1.1. «Тахіон» (стієчний варіант виконання)

Використовується для:

- формування і зберігання шкали точного часу
- видачі інформації споживачам про стан шкали часу UTC
- індикації поточного часу.

Технічні характеристики:

- можливість передачі інформації про час по різних інтерфейсах

- похибка часу при зникненні сигналів синхронізації GNSS – не більше 4,5 мікросекунди на добу

- різні варіанти конструктивного виконання – стієчний (див. рис 1.1), настінний, настільний

- в пристроях індикації забезпечується підтримка автономного режиму роботи (без синхронізації) з абсолютною похибкою ходу годинника в цьому режимі без додаткової корекції не більше 4 мілісекунди на добу.

1.2.2. Сервер часу TimeServer PLC2013

Система точного часу TimeServer PLC2013 (рис 1.2) призначена для синхронізації пристроїв по протоколу SNTP.

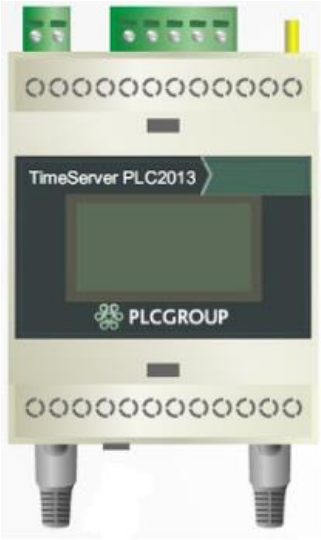


Рис. 1.2. TimeServer PLC2013

Пристрій розрахований на застосування на об'єктах енергетики, промислових підприємствах, а також в інших організаціях, що здійснюють самостійну прив'язку до системи точного часу.

Технічні характеристики:

- синхронізація часу по протоколу SNTPv4

- доступ по WEB-інтерфейсу для конфігурації пристрою

- підтримка мережевих протоколів: IPv4, ARP, TCP, HTTP, SNTPv4

- два незалежних інтерфейса Ethernet-10/100 (RJ45)

- синхронізація часу по двом портам Ethernet

- використовується вбудований приймач ГЛОНАСС/GPS і зовнішній NTP-сервер в якості опорних джерел синхросигналу

- внутрішній прецизійний осцилятор дозволяє забезпечити надійну автономну роботу сервера в умовах перешкод або втрати сигналу від супутникової навігаційної системи або втрати Інтернет-з'єднання.

1.2.3. Сервер точного часу Sinhron-M500

Сервер точного часу Sinhron-M500 (рис. 1.3) – система для синхронізації часу великого підприємства, побудована на платформі x86-64 і має можливість резервування основних вузлів для забезпечення високої відмовостійкості системи.



Рис. 1.3. Sinhron-M500

Комплекс призначений для формування сигналів еталонного часу і частоти з синхронізацією від супутникових навігаційних GPS/GLONASS систем, а також для видачі високостабільної опорної частоти 10 МГц.

Технічні характеристики:

- автоматична конфігурація DHCP

- підтримка протоколів NTP, SNTP, SNMP, PTP, RS-232

- можливість віддаленого доступу (Telnet, SSH/SFTP, SNMP v1, v2, v3, HTTP/HTTPS)

- 2 повністю незалежних антенних канали

- операційна система TimeOS

- зручний веб-інтерфейс зі статистикою та графіками.

1.3. Системи синхронізації часу

1.3.1. Радіостанції

Є багато способів для отримання інформації про точний час. Один з таких способів – отримання інформації шляхом прийняття корисного сигналу, який випромінюють передавачі на радіостанціях. Нижче описано декілька радіостанцій, які транслюють корисний сигнал, з їх характеристиками.

WWVB – це радіостанція часового сигналу поблизу Форт-Коллінс, штат Колорадо, яка управляється Національним інститутом стандартів і технологій. Більшість радіокерованих годинників у Північній Америці використовують передачі WWVB, щоб встановити правильний час. ERP-сигнал потужністю 70 кВт, що передається від WWVB, є безперервною несучою хвилею з частотою 60 кГц. Ця частота походить з декількох атомних годинників, розташованих на місці передавача, що дає невизначеність частоти менше 1 частини в 1012. Сигнал, який передається, несе в собі інформацію про рік, день року, годину, хвилину та деякі інші значення.

Time from NPL (MSF) (з англ. – Час від NPL (MSF)) – це радіосигнал, який транслюється з радіостанції Anthorn поблизу міста Анторн, Камбрія, яка служить національним орієнтиром Великобританії у часі. Сигнал часу походить від трьох атомних годинників, встановлених на місці передавача, і базується на стандартах часу, що підтримуються Національною фізичною лабораторією Великобританії у Теддінгтоні. Сигнал транслюється на високоточній частоті 60 кГц і може прийматися по всій території Великобританії, а також на більшій частині Північної

та Західної Європи. Носійна частота сигналу підтримується від 60 кГц до 2 частин в 1012, керуючись цезійними атомними годинниками на радіостанції.

RWM – радіопозивний групи короткохвильових передавачів еталонного сигналу часу «Москва». Сигнал узгоджений з еталоном часу, який знаходиться у Всеросійському науково-дослідницькому інституті фізико-технічних і радіотехнічних вимірювань (Менделєєве). Потужності передавачів: на частоті 4996 кГц – 5 кВт, на частотах 9996 і 14996 кГц – по 8 кВт.

JJY – це позивний низькочастотної радіостанції, яка транслює сигнал перевірки часу, розташованої в Японії. Станція транслюється з двох місць: одна на горі Отакадойя, а інша на горі Хагане. JJY управляється Національним інститутом інформаційно-комунікаційних технологій. Обидві станції транслюють ідентичний широтно-імпульсний тимчасовий код і передають 24 години на добу. Японський стандартний час задається атомним годинником в Токіо. Ця інформація відправляється на передавальні станції і використовується для установки точного часу на кожній станції. Формат тимчасового коду дуже схожий на формат WWVB в Сполучених Штатах. Аналогічно WWVB або MSF, сигнал JJY використовується для синхронізації споживчих радіокерованих годинників, що продаються по всій Японії.

DCF77 – позивний довгохвильового передавача точного часу і частоти, що забезпечує функціонування, серед іншого, годинників з автоматичною синхронізацією, а також систем телеметрії в Європі. Передавач DCF77 знаходиться в Німеччині і працює на частоті 77,5 кГц з потужністю 50 кВт. Інформаційний сигнал передається кожну хвилину з 0 по 58 секунду. Абревіатура складається з індексів D – Deutschland, C – передавач в довгохвильовому діапазоні, F – Франкфурт, число 77 позначає несучу частоту в 77,5 кГц. Сигнал знаходиться під керівництвом Фізико-технічної федеральної установи в Брауншвейзі. Посилався з 1959 року як сигнал еталонної частоти, з 1973 року став містити дані про дату і час. Точний час, що передається за допомогою передавача DCF77, є офіційним в ФРН і формується на підставі отриманих даних від трьох атомних годинників, похибка становить менше однієї секунди в один мільйон років. Прийом сигналу DCF77, в

залежності від часу доби і пори року, може бути здійснений на максимальній відстані до 1900 км вдень та до 2100 км вночі. Передається амплітудно-маніпульований сигнал, що містить дані в двійково-десятковій формі (по 1 біту даних в кожній секунді сигналу). Пакет даних, що посилається протягом хвилини, містить поточний час (в UTC+1) та дату (в тому числі день тижня).

1.3.2. Мережева синхронізація часу та синхронізація часу від орбітальних супутників

Існує два методи синхронізації часу:

- односторонній
- двосторонній.

При односторонньому методі синхронізації часу годинник А (А –годинник, який виступає в якості джерела точного часу для кінцевих пристроїв) відправляє інформацію про час годинникам Б (Б – кінцеві пристрої, на яких необхідно здійснити синхронізацію часу). Годинники Б використовують отримані дані для синхронізації часу, враховуючи затримку на передачу інформації. Даний метод може бути використаний тільки в тих системах, де мережа передачі даних і шлях від годинника А до годинника Б не змінюються, в іншому випадку при синхронізації часу буде невірно врахована затримка.

При двосторонньому методі синхронізації часу використовується двостороннє з'єднання між годинниками А та Б. Це необхідно для того, щоб динамічно визначати затримку при передачі інформації. Годинники А та Б обмінюються повідомленнями з мітками часу, після чого розраховується затримка, яка враховується при синхронізації часу на пристроях.

При синхронізації часу розрізняють два типи мережевих інфраструктур:

- виділена мережа
- конвергентна мережа.

У виділеній мережі для синхронізації часу використовують виділену лінію передачі даних. У такій мережі використовують методи синхронізації часу: 1PPS і IRIG-B.

У конвергентній мережі синхронізація часу відбувається по тим же каналам, по яким передається весь інший трафік мережі. Для подібних систем використовують протоколи синхронізації часу NTP, SNTP і PTP.

В окрему групу можна віднести системи синхронізації часу від орбітальних супутників – наприклад, GPS та ГЛОНАСС.

1PPS або PPS – це електричний сигнал, що має ширину менше однієї секунди і різко піднімається або різко падає, який точно повторюється один раз в секунду. Сигнали PPS виводяться радіомаяками, стандартами частоти, іншими типами прецизійних осциляторів та деякими GPS-приймачами. Точні годинники іноді виготовляють шляхом з'єднання генератора сигналів PPS з технологічним обладнанням, яке корегує PPS-сигнал під секунду UTC і переводить його на використовуваний дисплей. Атомні годинники зазвичай мають зовнішній вихід PPS, хоча внутрішньо вони можуть працювати при 9, 192, 631, 770 Гц. Сигнали PPS мають точність від 12 пікосекунд до декількох мікросекунд в секунду або від 2 наносекунд до декількох мілісекунд на день, в залежності від роздільної здатності та точності пристрою, що генерує сигнал. Сигнал PPS не містить мітки часу. Годинник А посилає 1 імпульс в секунду по окремій мережі. Годинник Б використовує цей імпульс тільки для синхронізації початку кожної секунди. Пристрої за допомогою PPS не можуть отримати інформацію по даті і часу, тому його найчастіше використовують спільно з протоколами синхронізації.

IRIG timecodes – стандартні формати для передачі інформації про час. Різні часові коди, визначені у стандарті, мають алфавітні позначення: A, B, D, E, G і H. Основна відмінність часових кодів – кількість імпульсів на одиницю часу, яка коливається між одним імпульсом в хвилину і 10000 імпульсів в секунду. Найбільш використовуваним є формат IRIG-B, який передає сто імпульсів в секунду, кодуючи інформацію в BCD. Це означає, що 100 біт інформації передається щосекунди. Цей кадр даних містить інформацію про день року (1–366), години,

хвилини та секунди. Хоча інформація передається лише один раз в секунду, пристрій може дуже точно синхронізувати свій час з пристроєм-передавачем, використовуючи фазово-замкнений цикл для синхронізації з носієм. Типові комерційні пристрої синхронізуються до 1 мікросекунди, використовуючи IRIG-B.

NTP – це найпопулярніший мережевий протокол для синхронізації часу, що використовується в глобальних та локальних мережах. Принцип роботи протоколу полягає у створенні ієрархії серверів часу і клієнтів. Усього рівнів ієрархії може бути 16 (починаючи з 0), годинники 16 рівня вважаються не синхронізованими. Кожен рівень містить джерела часу. Пристрої попереднього рівня завжди виступають в якості серверів часу для пристроїв нижчого рівня. Кожен пристрій в такій ієрархічній системі може отримувати дані про час з пристроїв свого рівня і попереднього. Отримані дані порівнюються за алгоритмом Марзулло і вибирається найбільш точний час. Алгоритм дозволяє зводити до мінімуму вплив даних від некоректно налаштованих NTP-серверів на загальну систему. Подібна розгорнута багаторівнева система також дозволяє оптимізувати трафік і навантаження на пристрої в мережі. Точність синхронізації по протоколу NTP в середньому становить 10 мс. Часто можна досягти точності близько 0,2 мс в локальних мережах.

SNTP – спрощена версія протоколу NTP. Застосовується в локальних мережах для некритичних до часу додатків. Формат повідомлень, якими між собою обмінюються пристрої в системах SNTP і NTP, ідентичний, тому протоколи сумісні один з одним. На відміну від NTP, у SNTP немає складних алгоритмів порівняння і вибору найкращого сервера часу, тому пристрій може бути синхронізовано тільки з одним сервером часу, і, якщо інформація на сервері некоректна, то кінцевий пристрій про це не дізнається.

PTP – протокол точного часу, який описується стандартом IEEE 1588 v2. Був розроблений для систем, яким не вистачає точності синхронізації, яку надають протоколи SNTP та NTP. PTP призначений для використання в локальних мережах і гарантує високу точність синхронізації. Даний протокол може бути реалізований на програмному або апаратному рівні пристроїв (найбільш точною є реалізація на

апаратному рівні). Точність досягається за рахунок обміну повідомленнями з часовими мітками між елементами системи PTP. Основними елементами в системі PTP є Головний годинник (годинник, за яким синхронізується час у системі), Провідний годинник (годинник, який виступає в якості джерела точного часу для кінцевих пристроїв) та Ведений годинник (кінцевий пристрій, на якому необхідно синхронізувати час). В стандарті IEEE 1588 v2 визначені алгоритми вибору Головного годинника, визначення затримок та їх компенсація, а також процес обміну повідомленнями. В системі PTP може бути декілька джерел точного часу, але активним може бути лише одне. Для вибору Головного годинника в системі PTP є спеціальний механізм BMCA (Best Master Clock Algorithm), який гарантує відмовостійкість системи. Як тільки призначений Головний годинник не може виконувати свою роль, то наступний годинник, що претендує на роль Головного годинника (по критеріям BMCA), автоматично стане джерелом точного часу для кінцевих пристроїв. Для цього, всі Провідні годинники системи постійно знаходяться в режимі прослуховування повідомлень, і, в разі необхідності, привласнюють собі роль Головного годинника. В системі PTP, за допомогою отриманих часових міток, які пересилаються повідомленнями, розраховується затримка. Затримка може розраховуватись двома методами: однокроковим та двокроковим. Різниця між ними в тому, що при двокроковому методі відправляється додаткове повідомлення з міткою часу. Також є два режими обчислення затримок в системі PTP: End-to-End та Peer-to-Peer. В одній системі краще використовувати однаковий режим роботи для більш коректної синхронізації пристроїв.

ГЛОНАСС – російська глобальна навігаційна супутникова система, яка призначена для оперативної доставки інформації про час та місцезнаходження користувачам наземного, космічного, повітряного і морського базування. Обмін інформацією здійснюється за допомогою 24 супутників, що переміщаються по 3-м орбітальним траєкторіях на висотах близько 19100 км. Дані супутники працюють асинхронно з обертанням Землі, завдяки цьому вдається досягти більшої стабільності трансльованого сигналу. ГЛОНАСС забезпечує доступ сигналу з

точністю 3-6 метрів. Доступ до цивільних сигналів системи ГЛОНАСС надається споживачам безкоштовно, без будь-яких обмежень та в будь-якій точці Землі.

GPS (Global Positioning System) – американська супутникова система навігації, яка транслює корисні сигнали про час і відстані та дозволяє визначати місця розташування об'єктів в межах глобальної координатної системи WGS 84. Система GPS розроблялася для використання Міністерством оборони США в своїх цілях, але зараз доступна для використання і цивільним користувачам. Система використовує 31 супутник (24 основних і 7 резервних), які працюють синхронно з обертанням Землі. Знаходяться дані супутники на висоті близько 20 тис. км і за добу встигають двічі облетіти Землю. GPS здатна забезпечити доступ до даних в будь-якому місці Землі (за винятком полярних широт) практично в будь-яких погодних умовах. GPS забезпечує доступ сигналу з точністю 2-4 м. Для забезпечення роботи інформаційного каналу достатньо придбати пристрій з GPS-приймачем. Синхронізація часу здійснюється під час визначення місцезнаходження пристрою, оснащеного GPS-приймачем. Для цього пристрій ловить сигнал із супутників, встановлених на навколоземній орбіті. Кожен з супутників має атомний годинник, за рахунок чого система GPS забезпечує високу точність. Мінусом даного методу синхронізації часу є необхідність в GPS-антені, сигнал від якої може бути нестабільним.

На сьогодні російська ГЛОНАСС і американська GPS є основними передавачами точного часу та діючими системами глобальної супутникової навігації.

1.4. Структурна схема системи

В залежності від того, звідки і як отримуються корисні сигнали про дату і час в системах єдиного часу, де обробляється отримана інформація та через що передається вихідна інформація елементи однієї системи можуть відрізнятися від елементів іншої.

На рис. 1.4 представлений один із варіантів структурної схеми системи єдиного часу. Наведена система (див. рис. 1.4) складається с трьох основних елементів з'єднаних між собою: мікроконтролер (для обробки отриманої від джерела інформації та керування всіма процесами в системі), годинник реального часу (для обліку інформації про дату та час) та мережева карта (мережевий модуль) (для можливості підключення системи єдиного часу до мережі).



Рис. 1.4. Структурна схема системи єдиного часу

Принцип роботи простий: мікроконтролер отримує дані про дату та час від джерела, оброблює ці дані та записує їх до годинника реального часу, який буде вести облік отриманих даних; коли система єдиного часу отримає запит на отримання дати та часу, мікроконтролер отримає необхідну інформацію від годинника реального часу, упакує дані у відповідний пакет та відправить його у мережу через мережевий модуль.

1.5 Програмні та апаратні засоби для реалізації системи

1.5.1. Програмні засоби

На даний момент, коли інформаційно-комп'ютерні технології розвиваються дуже швидко, існує велика кількість програмних продуктів для написання програмного коду. Декілька з них описано нижче.

Arduino IDE – це кросплатформний додаток написаний на функціях мов програмування C++ та C. Використовується для написання та завантаження програм на Arduino-сумісні плати та на плати інших виробників (за допомогою

ядер цих виробників). Інтерфейс додатку відносно простий та зрозумілий. Arduino IDE підтримує мови програмування C, C++ та Асемблер, використовуючи спеціальні правила структуризації коду. Бібліотеки програмного забезпечення постачаються з проекту Wiring. Написаний користувачем код вимагає лише дві основні функції для запуску скетчу (скетч – програма, написана в середовищі Arduino IDE): функцію setup та функцію loop. У функції setup проводяться налаштування і дана функція виконується лише один раз відразу після запуску. У функції loop відбувається виконання написано коду циклічно, тобто як тільки виконання коду дійде до кінця цієї функції, програма почне виконуватися спочатку. На рис 1.5 показано стартову сторінку Arduino IDE.

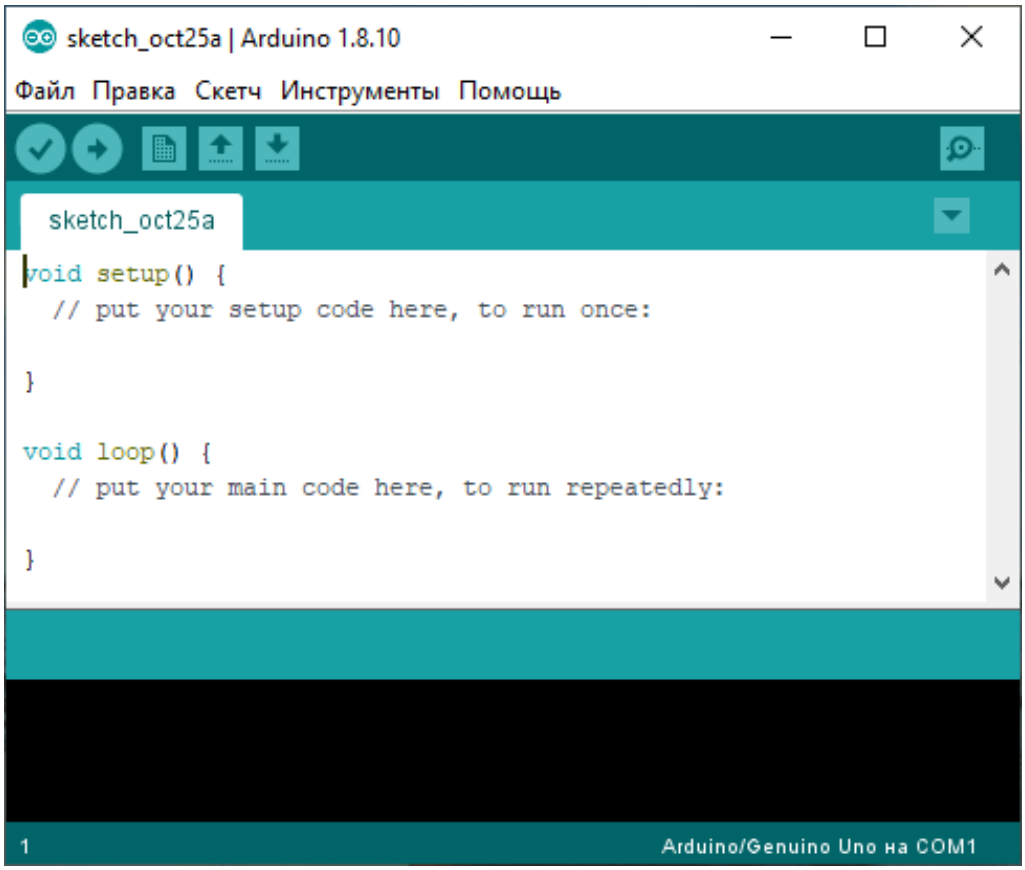


Рис. 1.5. Стартова сторінка Arduino IDE

MPIDE – середовище розробки, виконане на базі відкритої системи Arduino IDE і призначене спеціально для контролерів PIC32 від компанії Microchip Technology. MPIDE являє собою середовище для розробки додатків з відкритим вихідним кодом. Програма має свій власний унікальний GCC-компілятор під PIC32 і має повну сумісність з мовою програмування Wiring та з додатком Arduino.

Середовище MPIDE взаємодіє з високопродуктивними 32-бітними мікропроцесорами PIC32 від Microchip Technology, пропонуючи функції із створення, компіляції та програмування призначених для користувача додатків. Для прошивки не потрібен спеціальний апаратний програматор, процес виконується через віртуальний COM-порт за допомогою USB-порту і модуля avrdude, що входить до складу MPIDE. Стартова сторінка MPIDE показана на рис. 1.6.

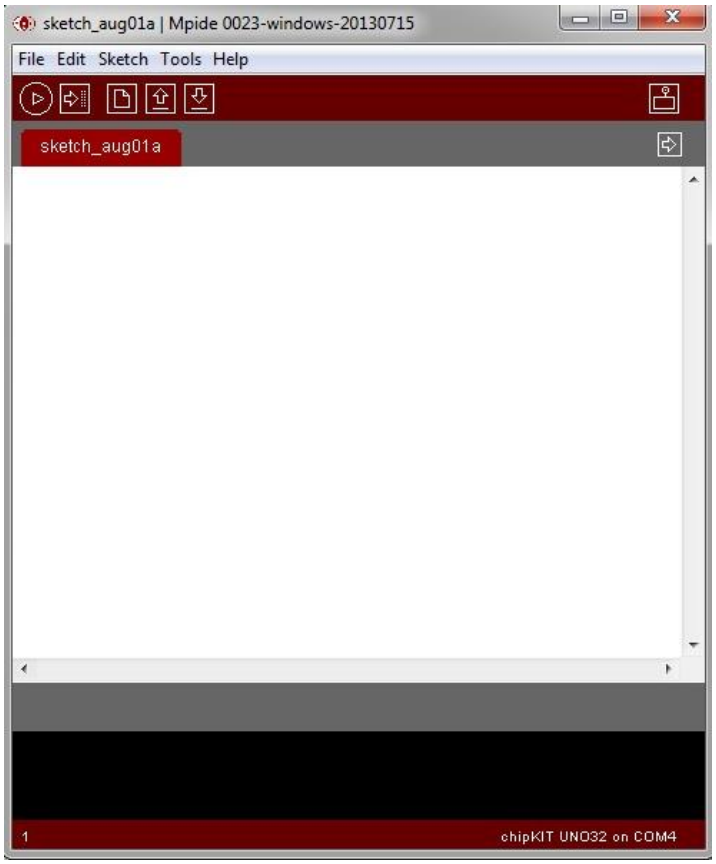


Рис. 1.6. Стартова сторінка MPIDE

Atmel Studio – інтегроване середовище розробки від компанії Atmel для розробки додатків під мікроконтролери ARM Cortex-M та AVR. Програма дозволяє працювати на мовах програмування C, C++ та Асемблері. Містить в собі майстер проектів, віртуальний симулятор, редактор вихідного коду, модуль внутрішньосхемного відлагодження та інтерфейс командного рядка. Підтримує компілятор GCC та плагін AVR RTOS (операційної системи реального часу). Користувачі можуть вибрати найбільш оптимальні для їх проекту способи кодування. Візуальні інструменти дозволяють прискорити написання програми.

Завдяки зв’язку програмних пакетів Atmel Studio та Proteus від фірми Labcenter Electronics можливе програмування мікроконтролерів без наявності будь-якої матеріальної бази. На рис. 1.7 показано стартову сторінку додатку.

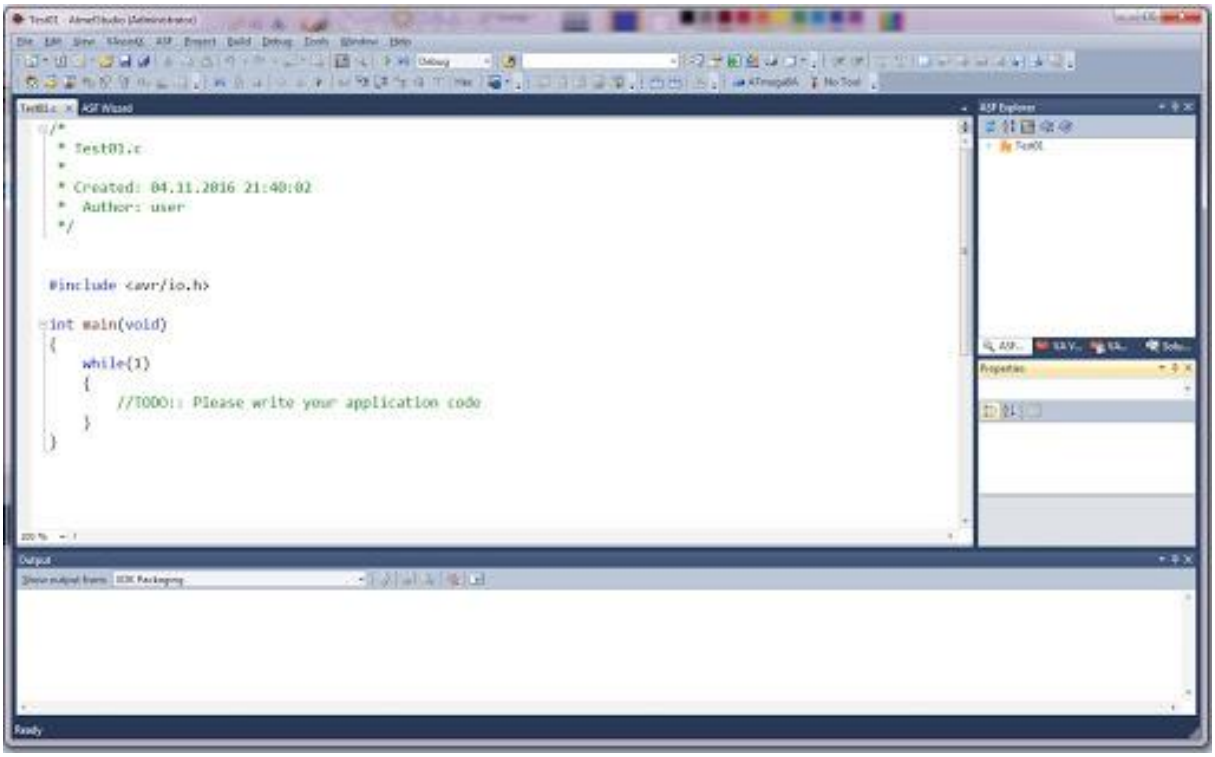


Рис. 1.7. Стартова сторінка Atmel Studio

1.5.2. Апаратні засоби

На наведеній вище структурній схемі системи єдиного часу (див. рис. 1.4) показано три основні елементи системи: мікроконтролер, мережевий модуль та годинник реального часу. Нижче наведено деякі варіації цих елементів.

Arduino Leonardo – контролер на базі ATmega32u4, зображено на рис. 1.8. Платформа має 20 цифрових вхід/виходів (7 з яких можуть використовуватися як виходи ШІМ і 12 як аналогові входи), кварцовий генератор 16 МГц, роз'єм micro-USB, силовий роз'єм, роз'єм ICSP і кнопку перезавантаження. Для роботи необхідно підключити платформу до комп'ютера за допомогою кабелю USB або подати живлення за допомогою адаптера AC/DC або батареї. Робоча напруга – 5В, вхідна напруга – 7-12В. Постійний струм через вхід/вихід – 40мА.

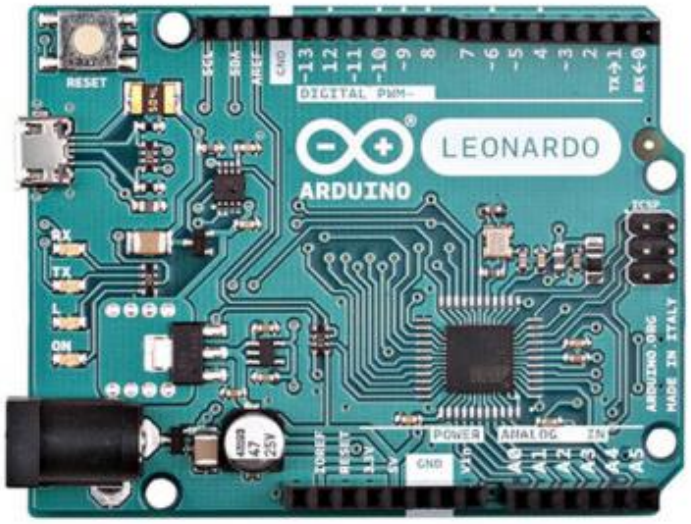


Рис. 1.8. Arduino Leonardo

Arduino Uno – контролер, що побудований на базі ATmega328, зображено на рис. 1.9. Платформа має 14 цифрових вхід/виходів (6 з яких можуть використовуватись як виходи ШІМ), 6 аналогових входів, кварцовий генератор 16 МГц, роз'єм USB, силовий роз'єм, роз'єм ICSP і кнопку перезавантаження. Для роботи необхідно підключити платформу до комп'ютера за допомогою кабелю USB або подати живлення за допомогою адаптера AC/DC або батареї. Робоча напруга – 5В, вхідна напруга – 7-12В. Постійний струм через вхід/вихід – 40мА.



Рис. 1.9. Arduino Uno

Arduino Mega – контролер на базі ATmega2560, зображено на рис. 1.10. Плата має 54 цифрових вхід/виходів (14 з яких можуть використовуватись як виходи

ШІМ), 16 аналогових входів, 4 послідовних порти UART, кварцовий генератор 16 МГц, роз’єм USB, роз’єм живлення, роз’єм ICSP і кнопка перезавантаження. Для роботи необхідно підключити платформу до комп’ютера за допомогою кабелю USB або подати живлення за допомогою адаптера AC/DC або акумуляторною батареєю. Робоча напруга – 5В, вхідна напруга – 7-12В. Постійний струм через вхід/вихід – 40мА.

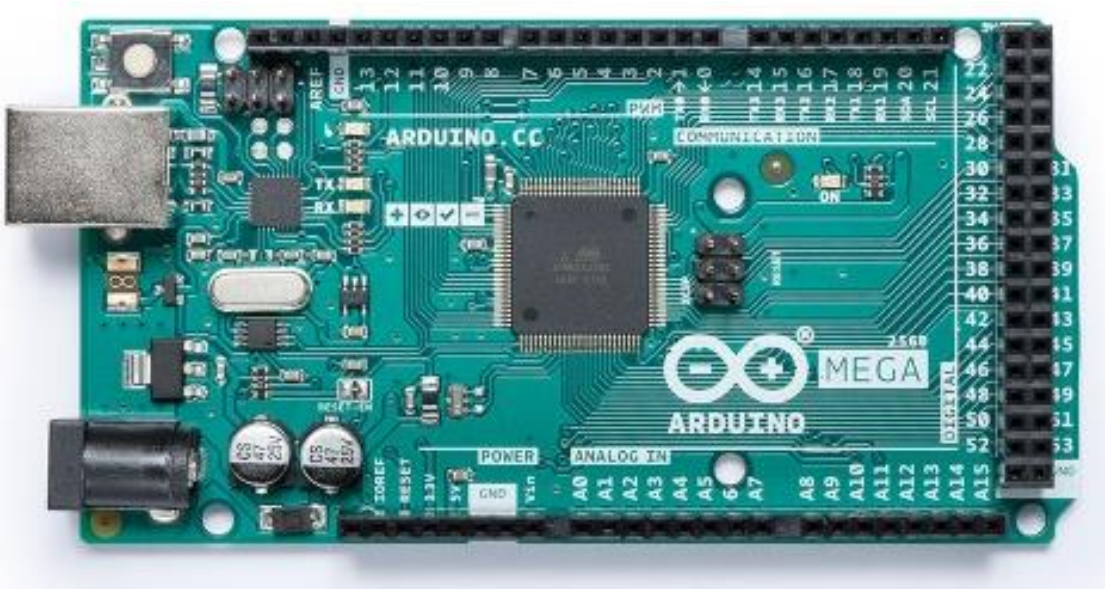


Рис. 1.10. Arduino Mega

Мережеві модулі дозволяють підключити плату Arduino до мережі.

W5100 Ethernet Shield – мережевий модуль, побудований на Ethernet-мікросхемі Wiznet W5100, зображено на рис. 1.11. Підтримує стеки TCP та UDP в IP-мережі та до чотирьох одночасних підключень до сокетів. Сумісний з Arduino Uno та Mega при використанні бібліотеки Ethernet Library. З'єднується з платою Arduino за допомогою довгих штирів, що проходять через даний модуль, що дозволяє не змінювати розташування виводів і встановлювати інші плати поверх цього модуля. Має стандартний роз'єм RJ45 з вбудованим лінійний трансформатором і опцією Power over Ethernet (PoV) Останні версії плати мають роз'єм для карт типу micro-SD, який може використовуватися для зберігання файлів і роботи з ними по мережі.



Рис. 1.11. W5100 Ethernet Shield

Nano W5100 Ethernet Shield – мережевий модуль, побудований на базі Wiznet W5100, зображено на рис. 1.12. Підтримує стеки TCP та UDP в IP-мережі. З’єднується з Arduino Nano за допомогою довгих штирів, що проходять через даний мережевий модуль. Має стандартний роз'єм RJ45 та роз'єм для карт типу micro-SD.

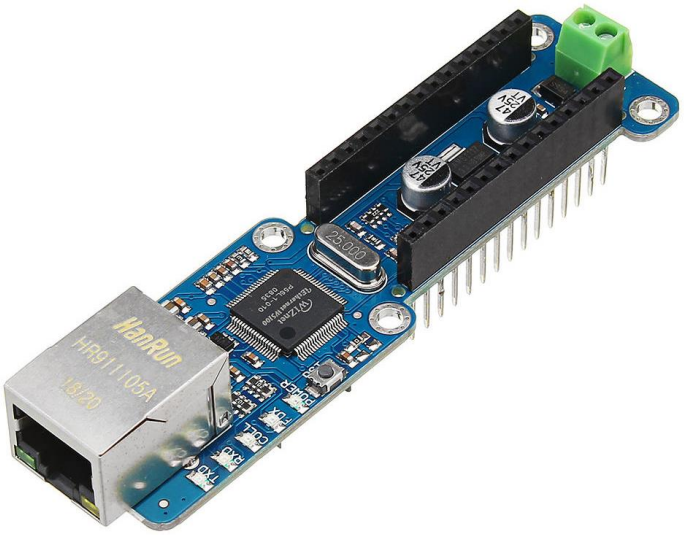


Рис. 1.12. Nano W5100 Ethernet Shield

W5500 Ethernet Shield – мережевий модуль, побудований на базі Wiznet W5500, зображено на рис. 1.13. Це більш нова модель мережевого модуля, попередником якої є W5100. На модулі встановлено обв’язку чипа, стабілізатор на 3,3В та роз’єм RJ45 з узгоджуючим трансформатором. Модуль може живитися від напруги 3,3В та 5В. Логічні рівні на інформаційних виводах 3,3В, проте можуть нормально працювати при 5В.

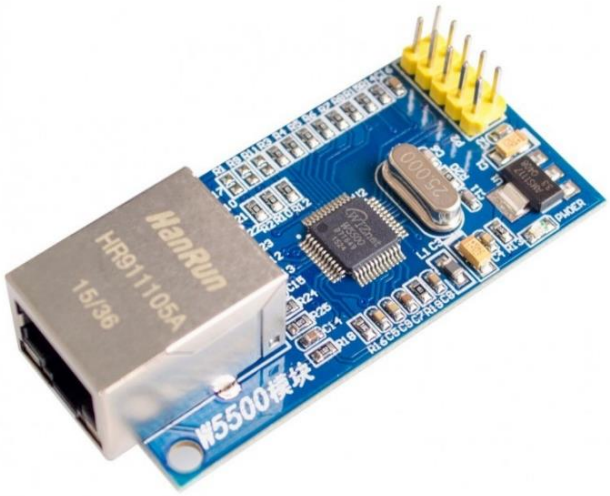


Рис. 1.13. W5500 Ethernet Shield

DS3231 – це недорогий, високоточний годинник реального часу, що працює по протоколу I2C. Модуль являє собою систему із автономного живлення та обчислюваного пристрою. На платі DS3231 встановлено джерело резервного живлення на базі елемента CR2032. Напруга живлення становить 2,7-5,3В. Модуль підтримує секунди, хвилини, години, дні тижня, дату, місяць, рік (високосні роки враховуються до 2100 року). Годинник працює в 24-и годинному або 12-и годинному форматі. Модуль можна використовувати як годинник або будильник або в якості оповіщення для різних систем в «Розумному будинку». Похибка ходу становить не більше 2-х хвилин на рік. Зовнішній вигляд DS3231 з двох сторін показано на рис. 1.14.

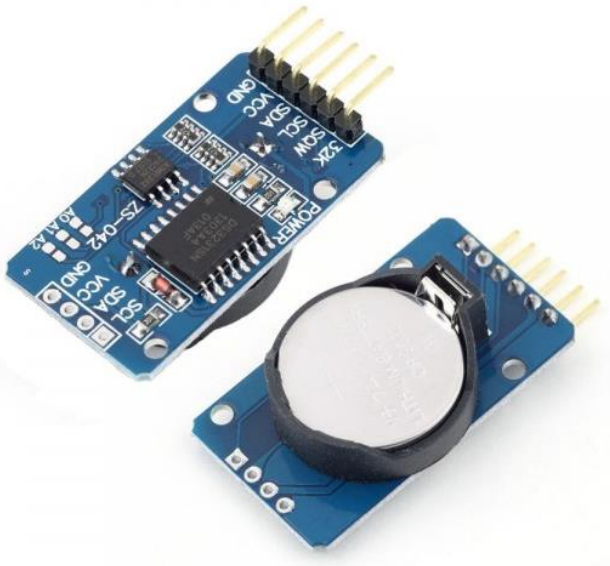


Рис. 1.14. DS3231

DS1307 – годинник реального часу, створений на базі мікросхеми DS1307ZN. Є попередником DS3231. Має автономну систему живлення від літієвої батарейки LIR2032. Напруга живлення становить 5В. Інші параметри схожі до DS3231. Зовнішній вигляд з двох сторін показано на рис. 1.15.

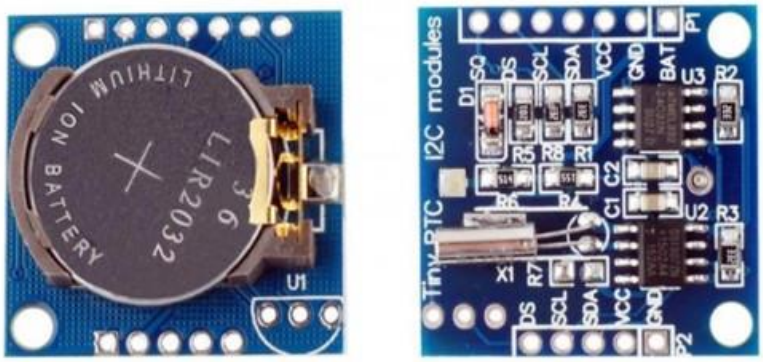


Рис. 1.15. DS1307

Висновки з першого розділу

В даному розділі дипломної роботи було опрацьовано теоретичний матеріал стосовно розробки системи єдиного часу: визначено, які задачі вимагають використання системи єдиного часу; проаналізовано існуючі системи; описано способи синхронізації часу та структурну схему; розглянуто програмні та апаратні засоби для реалізації системи.

РОЗДІЛ 2. ВИБІР ЕЛЕМЕНТНОЇ БАЗИ, РОЗРОБКА СИСТЕМИ ЄДИНОГО ЧАСУ

2.1. Визначення вузлів структурної схеми системи

В попередньому розділі було розглянуто теоретичний матеріал стосовно систем єдиного часу. Тепер можна переходити до розробки системи. Але спочатку необхідно точно визначитися з вузлами структурної схеми.

Перший елемент, з яким необхідно визначитись, є мікроконтролер, який буде ядром розрахунків та керівних процесів. Серед наведених в першому розділі, найбільше, до розроблюваної системи, підходить контролер Arduino Leonardo (див. рис. 1.8) на базі ATmega32u4, через свою доступну ціну та відповідність необхідним характеристикам майбутньої системи.

Наступний елемент, з яким необхідно визначитись – годинник реального часу, який буде проводити облік хронометричних даних, таких як поточний час, дату, день тижня і т.д. З опрацьованої в першому розділі інформації про годинники реального часу, можна зробити висновок, що суттєвих відмінностей між розглянутими варіаціями даного елемента немає. Тому, виберемо ту модель, яка надає більші можливості у роботі з нею. Такою моделлю є DS3231 (див. рис. 1.14).

Далі необхідно вибрати елемент, який буде передавати вихідну інформацію про точні дату та час в мережу. Таким елементом є мережевий модуль. Варіації цього елемента були представлені в першому розділі, і на основі описаної інформації було визначено, що найбільше, до розроблюваної системи, підходить мережевий модуль W5100 Ethernet Shield (див. рис. 1.11). Модуль зручно з'єднується з контролером Arduino Leonardo за допомогою довгих штирів, що проходять через даний модуль, що дозволяє не змінювати розташування виводів і встановлювати інші плати поверх цього модуля.

Програмним засобом, для написання програмного коду для мікроконтролера Arduino Leonardo, було обрано кросплатформний додаток тієї ж компанії – Arduino IDE.

2.2. Розробка алгоритму роботи системи єдиного час

Після визначення елементів, на основі яких буде розроблятися система точного часу, та перед безпосередньою розробкою даної системи, необхідно розробити її алгоритм роботи. Але спочатку визначимо, звідки система буде брати інформацію про точні дату та час.

Для мережі розумного будинку, де точний час не є критичним показником і невелика похибка допустима, оптимальним джерелом інформації для системи єдиного часу є персональний комп'ютер, який синхронізується із своїм джерелом. Тому, як джерело інформації для розроблюваної системи єдиного часу, було вибрано персональний комп'ютер.

Після того, як було визначено джерело інформації про точні дату та час, можна переходити до розробки алгоритму роботи системи. Найкраще видно алгоритм роботи пристрою на його функціональній схемі.

Функціональна схема – схема, що роз'яснює певні процеси, що відбуваються у певних функціональних частинах виробу чи у виробі в цілому. Схеми такого виду складаються з дотриманням стандартів, які встановлені державою. Проте, у програмуванні, для візуалізації алгоритмів роботи додатку (системи), дозволяється створювати схеми такого виду так, як зручно автору.

Для зручності створення та пояснення функціональної схеми системи, дану схему слід розділити на окремі функціональні частини, які в результаті будуть об'єднані та сформують функціональну схему.

Перша функціональна частина – «персональний комп'ютер - мікроконтролер - годинник реального часу». Алгоритм роботи простий: мікроконтролер відправляє запит на отримання системних дати та часу з персонального комп'ютера. Отримавши інформацію про дату та час, мікроконтролер записує ці дані до годинника реального часу.

Алгоритм роботи першої функціональної частини показано на рис. 2.1.



Рис. 2.1. Алгоритм роботи першої функціональної частини

Друга функціональна частина – «мікроконтролер - мережевий модуль - годинник реального часу». Алгоритм роботи наступний: мікроконтролер опитує мережевий модуль:

- мережевий модуль відправляє «ні», якщо запитів на отримання інформації про точні дату та час немає, і мікроконтролер знову починає опитувати мережевий модуль;

- мережевий модуль відправляє «так», якщо був відправлений запит на отримання інформації про точні дату та час. Мікроконтролер запитує необхідну інформацію у годинника реального часу, отримує відповідь, упаковує отримані дані у відповідний пакет та відправляє його у мережу через мережевий модуль.

Алгоритм роботи другої функціональної частини показано на рис. 2.2.

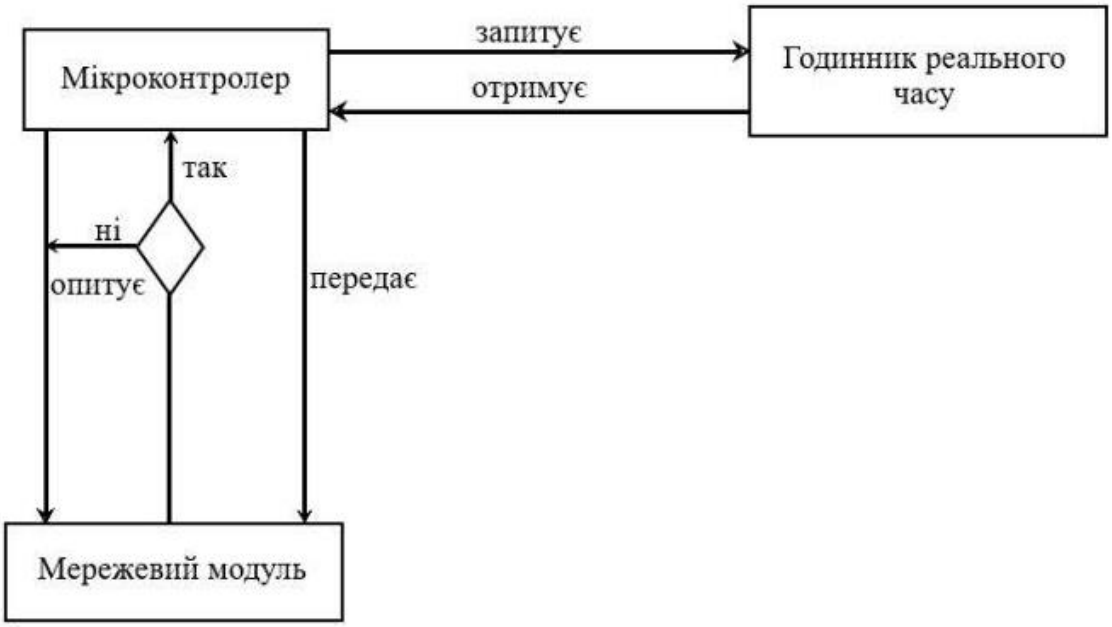


Рис. 2.2. Алгоритм роботи другої функціональної частини

Тепер необхідно об’єднати дві функціональні частини в одне ціле, щоб сформувати функціональну схему системи. Результат об’єднання показано на рис. 2.3.



Рис. 2.3. Функціональна схема системи єдиного часу

2.3. З’єднання елементів системи єдиного часу

Після визначення елементів системи єдиного часу та розробки її алгоритму роботи та перед написанням програмного коду, необхідно з’єднати (підключити) ці елементи один з одним.

Спочатку об’єднаємо мікроконтролер та мережевий модуль. Для цього, поставимо мікроконтролер на стійку тверду поверхню та «посадимо» зверху на нього мережевий модуль, вставляючи довгі штирі цього модуля у відповідні роз’єми на мікроконтролері. Сумісність мікроконтролера та мережевого модуля дозволяє не змінювати розташування виводів і встановлювати інші елементи поверх мережевого модуля. Результат об’єднання показано на рис. 2.4.

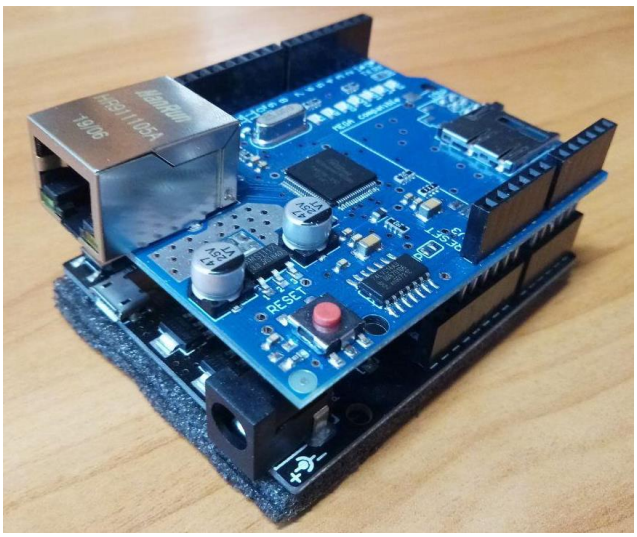


Рис. 2.4. З’єднані мікроконтролер та мережевий модуль

Тепер потрібно підключити годинник реального часу до об’єднаних мікроконтролера та мережевого модуля. Як було зазначено в першому розділі, DS3231 працює по протоколу I2C. Для цього на даному модулі є спеціальні виводи SDA та SCL, крім того є виводи VCC та GND, що відповідають за живлення та заземлення даного модуля відповідно. Також присутні виводи SQW та 32K, але їх використовувати не будемо.

Для кожного виду мікроконтролерів, що описані в першому розділі, крім однойменних роз’ємів, реалізовані резервні роз’єми для виводів SDA та SCL. На кожному з мікроконтролерів резервні виводи різні. Для Arduino Leonardo, який був вибраний для розробки даної системи, резервні роз’єми для виводів SDA та SCL є цифрові 2 та 3 відповідно. Тому підключатися будемо до них.

Виводи на DS3231 реалізовані штирями, а на мережевому модулі присутні роз’єми, куди можна підключатися. Для того, щоб виконати підключення, необхідно мати з’єднувальні проводи типу «мама-тато», які показані на рис. 2.5.

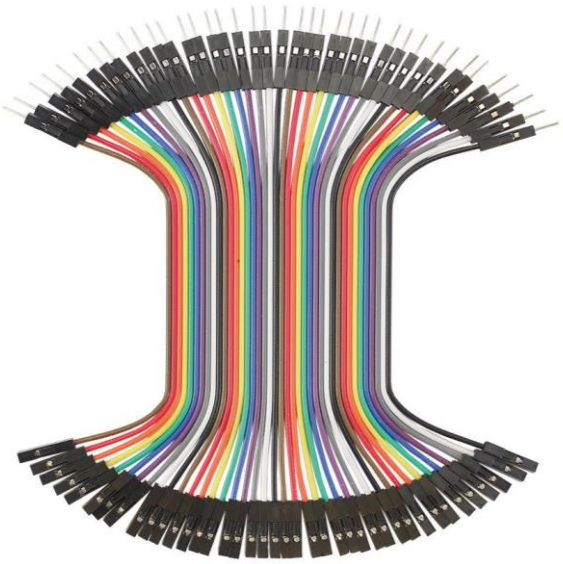


Рис. 2.5. З’єднувальні проводи

За допомогою цих з’єднувальних проводів підключаємо модуль реального часу до мережевого модуля: виводи SDA та SCL підключаємо до цифрових роз’ємів 2 та 3 відповідно, вивід GND до однойменного роз’єму GND, вивід VCC до роз’єму 5V (живитися модуль реального часу буде від напруги 5В). Результат підключення показано на рис. 2.6.

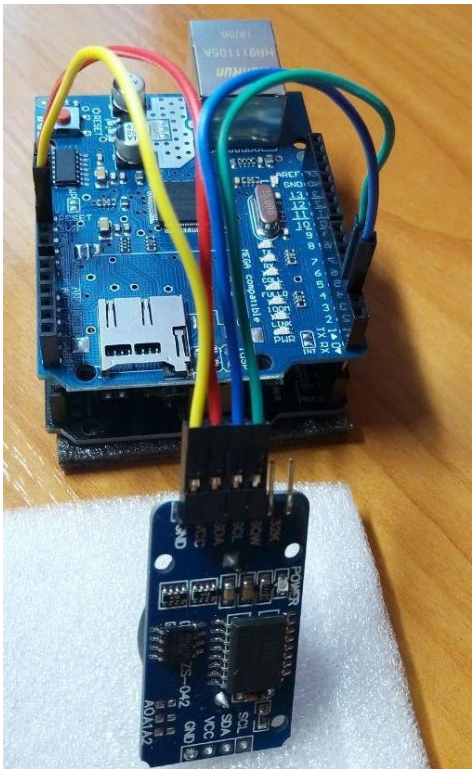


Рис. 2.6. Підключений модуль годинника реального часу

Після з’єднання основних елементів, систему необхідно підключити до мережі. Для цього нам необхідно мати мережевий кабель RJ45 типу «тато-тато», що показано на рис. 2.7.



Рис. 2.7. Мережевий кабель

Один кінець даного кабелю потрібно вставити у відповідний роз’єм на мережевому модулі, а інший – підключити до маршрутизатора. Результат підключення показано на рис. 2.8.

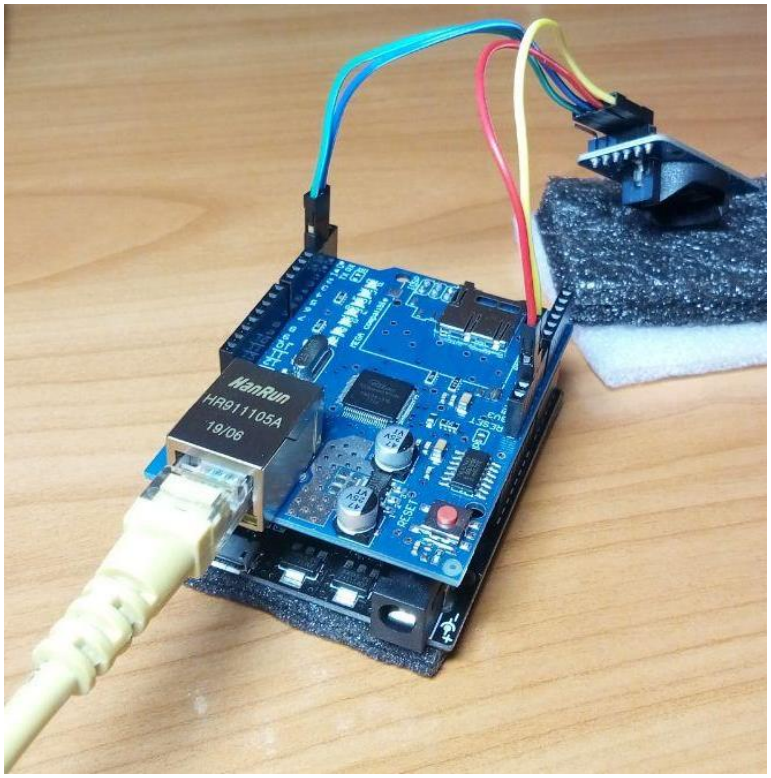


Рис. 2.8. Підключений мережевий кабель

Тепер залишилося подати живлення на систему. Для цього на мікроконтролері є роз’єм micro-USB, через який, крім того, можна буде заливати програмний код. Для підключення необхідно мати кабель micro-USB, який представлено на рис. 2.9.



Рис. 2.9. Кабель micro-USB

Кабель необхідно вставити однією частиною в відповідний роз’єм на мікроконтролері, а іншою – в USB-порт увімкненого комп’ютера. Результат підключення показано на рис. 2.10.

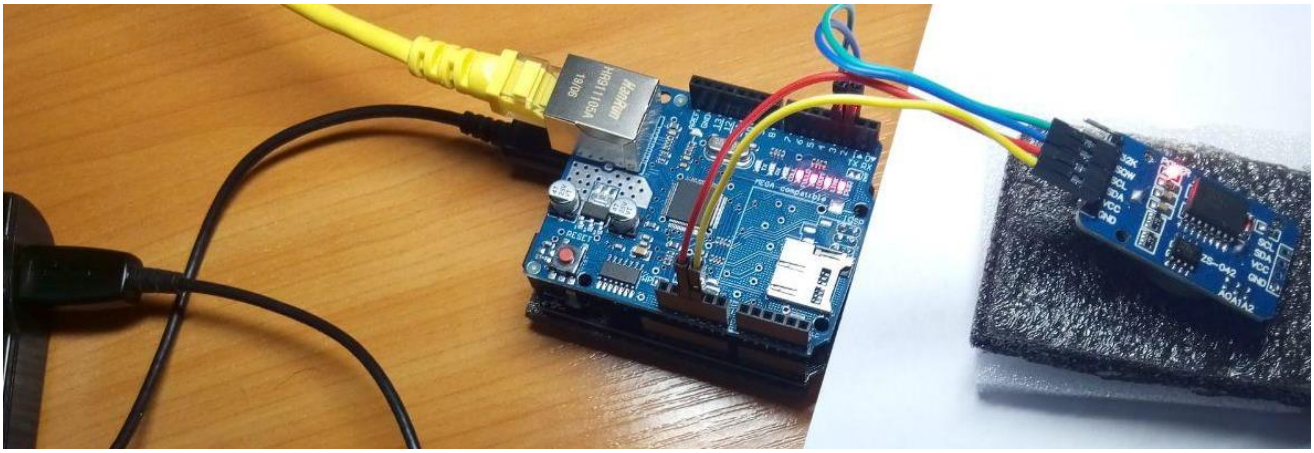


Рис. 2.10. Результат підключення системи до комп’ютера

На рис. 2.10 видно, що на всі елементи системи надійшло живлення, про що свідчать увімкнені діоди з підписом PWR, отже підключення пройшло правильно і можна переходити до написання програмного коду.

2.4. Розробка програмного коду мікроконтролера

Після визначення елементів системи, їх об’єднання та розробки функціональної схеми, можна переходити до програмної реалізації алгоритму роботи системи єдиного часу – написання коду.

Для полегшення написання деяких частин коду, будемо використовувати функції, які описані в бібліотеках. Необхідні функції описані в таких бібліотеках: Wire.h, Ethernet.h, EthernetUdp.h та RTClib.h. Підключення даних бібліотек показано на рис. 2.11.

```
#include <Wire.h>
#include <Ethernet.h>
#include <EthernetUdp.h>
#include "RTClib.h"
```

Рис. 2.11. Підключення бібліотек

Для того, щоб використати описані в приведених вище бібліотеках функції, необхідно створити об’єкти класів цих бібліотек. Таких об’єктів має бути три. Створення об’єктів показано на рис. 2.12.

```
EthernetUDP Udp;
EthernetServer server(80);
RTC_DS1307 rtc;
```

Рис. 2.12. Створення об’єктів

Далі необхідно вказати IP-адресу, MAC-адресу та порт NTP-сервера (рис. 2.13). IP-адресу вибираємо з доступних локальних IP-адрес, які можна побачити в налаштування маршрутизатора (в процесі розробки було вибрано IP-адресу 192.168.0.100). MAC-адресу вказуємо ту, яка зазначена на нижній частині мережевого модуля. Значення порту має бути 123 – це стандартне значення.

```
IPAddress ipAddrs(192, 168, 0, 100);
byte macAddrs[] = { 0x00, 0xAA, 0xBB, 0xCC, 0xDE, 0x02 };
#define NTP_PORT 123
```

Рис. 2.13. Зазначення IP-адреси, MAC-адреси та порту

Наступним кроком створюємо необхідні глобальні змінні, які будемо використовувати в подальшій розробці. Всі створені глобальні змінні представлено на рис. 2.14.

```
uint32_t timestamp, timestamptmp;
int years, months, days, hours, minutes, seconds;
const unsigned long unixTimeStamp = 2208988800UL;
const uint8_t daysInMonth [] PROGMEM = { 31,28,31,30,31,30,31,31,30,31,30,31 };
static const int NTP_SIZE_OF_PACKET = 48;
byte arrSendPacket[NTP_SIZE_OF_PACKET];
```

Рис. 2.14. Створення глобальних змінних

Тепер перейдемо до написання власних функцій.

Для отримання даних про дату та час та запису цих даних до годинника реального часу будемо використовувати власну функцію writeTimeToRTC(). В тілі цієї функції описана інша функція rtc.adjust(DateTime(…)), яку надає бібліотека RTClib.h. Ця функція (на місці трьох крапок) може приймати шість аргументів, які вказують рік, місяць, день місяця, години, хвилини та секунди та записувати ці значення до годинника реального часу. Значення аргументів можна передавати через змінні або вводити в ручну. Також, дату та час можна запитувати з головного

комп’ютера, для цього, як аргументи, потрібно передати наступну конструкцію: __DATE__, __TIME__. Тіло функції writeTimeToRTC() зображено на рис. 2.15.

```
void writeTimeToRTC(){
  //rtc.adjust(DateTime(__DATE__, __TIME__)); //запитуємо дані з персонального комп'ютера
  //rtc.adjust(DateTime(years, months, days, hours, minutes, seconds)); //передаємо через змінні
  rtc.adjust(DateTime(2020, 5, 13, 17, 0, 0));//вводимо в ручну
}
```

Рис. 2.15. Функція writeTimeToRTC()

Для того, щоб зчитати дані про дату і час з модуля реального часу та записати їх у створені глобальні змінні, описуємо функцію getTimeFromRTC() (рис. 2.16). В тілі функції створюємо об’єкт now та, за допомогою наданих бібліотекою RTClib.h функцій, записуємо запитані дані в створені глобальні змінні.

```
void getTimeFromRTC(){//отримання даних з RTC
  DateTime now = rtc.now();

  years = now.year(), DEC;
  months = now.month(), DEC;
  days = now.day(), DEC;
  hours = now.hour(), DEC;
  minutes = now.minute(), DEC;
  seconds = now.second(), DEC;
}
```

Рис. 2.16. Функція getTimeFromRTC()

Тепер необхідно створити функцію, яка буде обробляти запити до нашого NTP-сервера. Для цієї задачі буде використовуватись власна функція receivingAndDataSending(). В тілі цієї функції, за допомогою об’єкта Udp класу EthernetUdp та функції parsePacket(), яка надається бібліотекою EthernetUdp.h, перевіряється наявність пакету: функція parsePacket() повертає розмір пакету у локальну змінну sizeOfPacket. Якщо пакет був прийнятий (його розмір відмінний від 0), то виконується код, який описаний в конструкції if(), а саме: отримуються дані з модуля реального часу, передаються в функцію numberOfSeconds(), яка формує мітку часу з 1900 року. Далі дані упаковуються в байтовий масив arrSendPacket, який формує пакет, який має бути відправлений запитувачу. Останнім кроком сформований пакет з даними про дату та час відправляється запитувачу. Повний код, що міститься в тілі функції receivingAndDataSending(), можна знайти в Додатку.

Для зручності роботи з системою під час її розробки та можливого подальшого вдосконалення, створюємо дві функції printToSerialPort() та printToLocalHost(). Перша функція відправляє зчитані з модуля реального часу дані про час в серійний порт. Друга створює на локальній IP-адресі, яка вказана для NTP-сервера, просту html-сторінку, на якій відображаються зчитані з модуля реального часу дані про час. Опис та звернення до цих функцій можна видалити з коду або не записувати (закоментувати) звернення до них в функцію loop(). Відсутність цих функції не впливає на роботу системи єдиного часу. Код функцій printToSerialPort() та printToLocalHost() показано на рис. 2.17 та 2.18 відповідно.

```
void printToSerialPort(){//виведення даних про час в серійний порт
  getTimeFromRTC();

  Serial.print(hours);
  Serial.print(":");
  Serial.print(minutes);
  Serial.print(":");
  Serial.println(seconds);
}
```

Рис. 2.17. Функція printToSerialPort()

```
void printToLocalHost(){//виведення даних про час на вказаний IP-адрес
  getTimeFromRTC();

  EthernetClient client = server.available();

  client.println("<!DOCTYPE HTML>");
  client.println("<html>");
  client.print(hours);
  client.print(":");
  client.print(minutes);
  client.print(":");
  client.print(seconds);
  client.println("<br/>");
  client.println("</html>");
}
```

Рис. 2.18. Функція printToLocalHost()

Після написання необхідних функцій потрібно встановити налаштування в стандартній функції setup(), які виконуються один раз під час запуску. В цій функції

ми запускаємо серійний порт, інтерфейс I2C, роботу з бібліотекою RTClib.h, мережевий модуль, роботу з об'єктом server, викликаємо функцію writeTimeToRTC(), для запису значень дати та часу в модуль реального часу. Всі налаштування показано на рис. 2.19.

```
void setup () {
  Serial.begin(9600);
  Wire.begin();
  rtc.begin();
  Ethernet.begin(macAddrs, ipAddrs);
  server.begin();
  Udp.begin(NTP_PORT);
  writeTimeToRTC();
}
```

Рис. 2.19. Налаштування в функції setup()

Далі необхідно заповнити стандартну функцію loop(), в якій написаний код буде виконуватись циклічно. Сюди вписуємо функцію receivingAndDataSending(), для обробки запитів, які приходять на розроблювану систему єдину часу. Також, можна додати функції printToSerialPort() та (чи) printToLocalHost(), якщо є необхідність виводи час в серійний порт та (чи) на html-сторінку. Код функції loop() зображено на рис. 2.20.

```
void loop () {
  receivingAndDataSending();
}
```

Рис. 2.20. Функція loop()

Тепер можна завантажувати написаний програмний код в мікроконтролер та відстежувати роботу.

Висновки з другого розділу

В даному розділі дипломної роботи було проведено розробку системи єдиного часу: визначено вузли структурної схеми, джерело отримання корисної інформації про дату та час; розроблено алгоритм роботи системи єдиного часу та

створено функціональну схему; наведено процес підключення (з’єднання) елементів системи; розроблено програмний код мікроконтролера.

РОЗДІЛ 3. ОПИС РОБОТИ З СИСТЕМОЮ ЄДИНОГО ЧАСУ ТА ДЕМОНСТРАЦІЯ РОБОТИ СИСТЕМИ

3.1 Опис роботи з системою єдиного часу

В попередньому розділі було розроблено систему єдиного часу. Тепер потрібно описати, як працювати з цією системою.

Для початку необхідно визначити, яка IP-адреса буде у даного сервера та записати її у програмному коді. Для запису значень в код, необхідно встановити середовище розробки Arduino IDE, запустити його та відкрити скетч з програмним кодом даної системи.

Для того, щоб визначити список доступних локальних IP-адрес, необхідно зайти в налаштування маршрутизатора, перейти на вкладку «DHCP» та вибрати «Налаштування DHCP» (рис. 3.1).

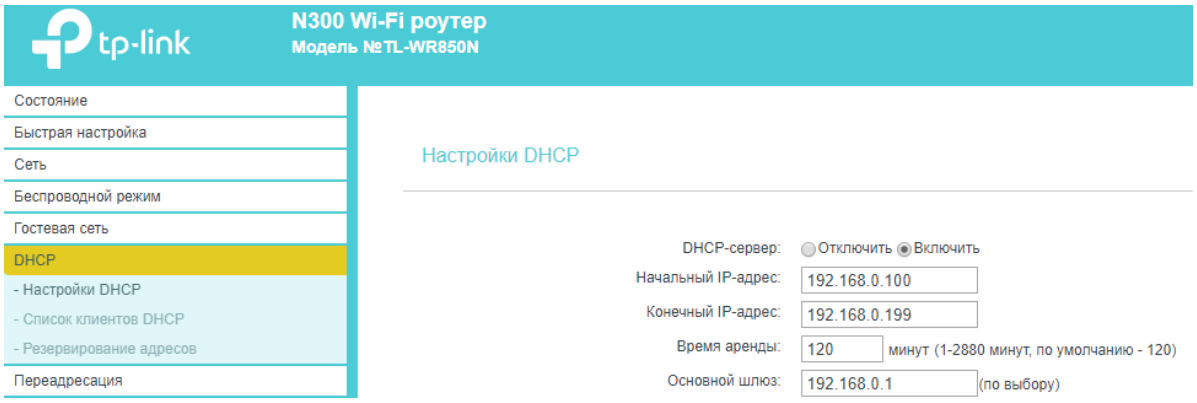


Рис. 3.1. Вкладка «Налаштування DHCP» в налаштування маршрутизатора

На рис. 3.1 видно, що доступний діапазон локальних IP-адрес від 192.168.0.100 до 192.168.0.199. Тепер необхідно визначити, які адреси не зайняті. Для цього, на цій же сторінці переходимо на вкладку «Список клієнтів DHCP» (рис. 3.2).

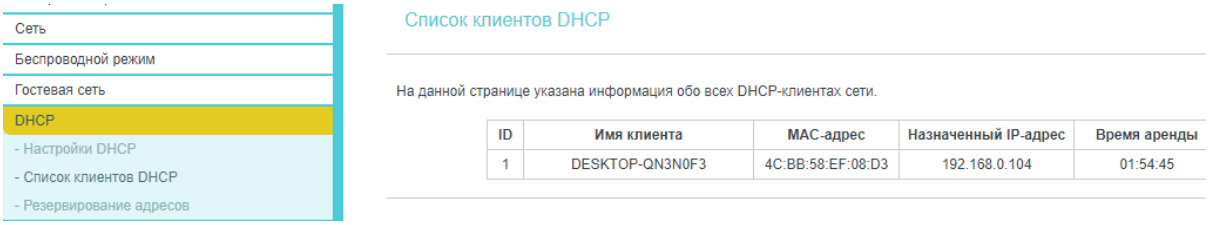


Рис. 3.2. Вкладка «Список клієнтів DHCP»

Як бачимо, з усього доступного діапазону зайнята лише IP-адреса 192.168.0.104. Тому, можна вибрати будь-яку IP-адресу, крім зайнятої та ввести її в спеціальне місце в коді (рис. 3.3).

```
IPAddress ipAddrs(192, 168, 0, 100);
```

Рис. 3.3. Введення IP-адреси

Тепер, необхідно ввести значення MAC-адреси в програмний код. Вказуємо те, яке зазначено на нижній частині мережевого модуля. Якщо на мережевому модулі нічого не вказано, можна залишити стандартне значення, яке вже вказано в програмному коді (рис. 3.4).

```
byte macAddrs[] = { 0x00, 0xAA, 0xBB, 0xCC, 0xDE, 0x02 };
```

Рис. 3.4. Введення MAC-адреси

Далі потрібно визначити, звідки будемо брати дані: вводити вручну, передавати через змінні чи запитувати з головного комп’ютера. Після прийняття рішення, розкоментувати потрібний варіант отримання даних та закоментувати два інших. На рис. 3.5 вибраний варіант введення значень вручну.

```
void writeTimeToRTC(){
  //rtc.adjust(DateTime(__DATE__, __TIME__)); //запитуємо дані з персонального комп'ютера
  //rtc.adjust(DateTime(years, months, days, hours, minutes, seconds)); //передаємо через змінні
  rtc.adjust(DateTime(2020, 5, 13, 17, 0, 0));//вводимо в ручну
}
```

Рис. 3.5. Введення значень вручну

Тепер потрібно завантажити програмний код в мікроконтролер. Для цього, в Arduino IDE, потрібно перейти на вкладку «Інструменти», далі – на розділ «Порт» та вибрати порт, до якого підключений мікроконтролер (рис. 3.6).

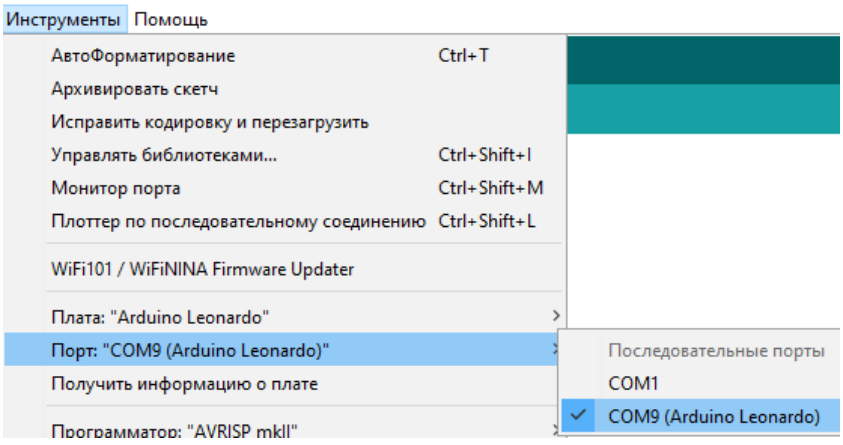


Рис. 3.6. Вибір порту

Далі потрібно натиснути кнопку «Завантажити» (рис. 3.7), дочекатися кінця процесу та закрити Arduino IDE.



Рис. 3.7. Кнопка завантаження

Щоб перевірити чи створився сервер, слід ввести в командному рядку комп'ютера, який підключений до локальної мережі, команду ping з вибраною для сервера IP-адресою (рис. 3.8).

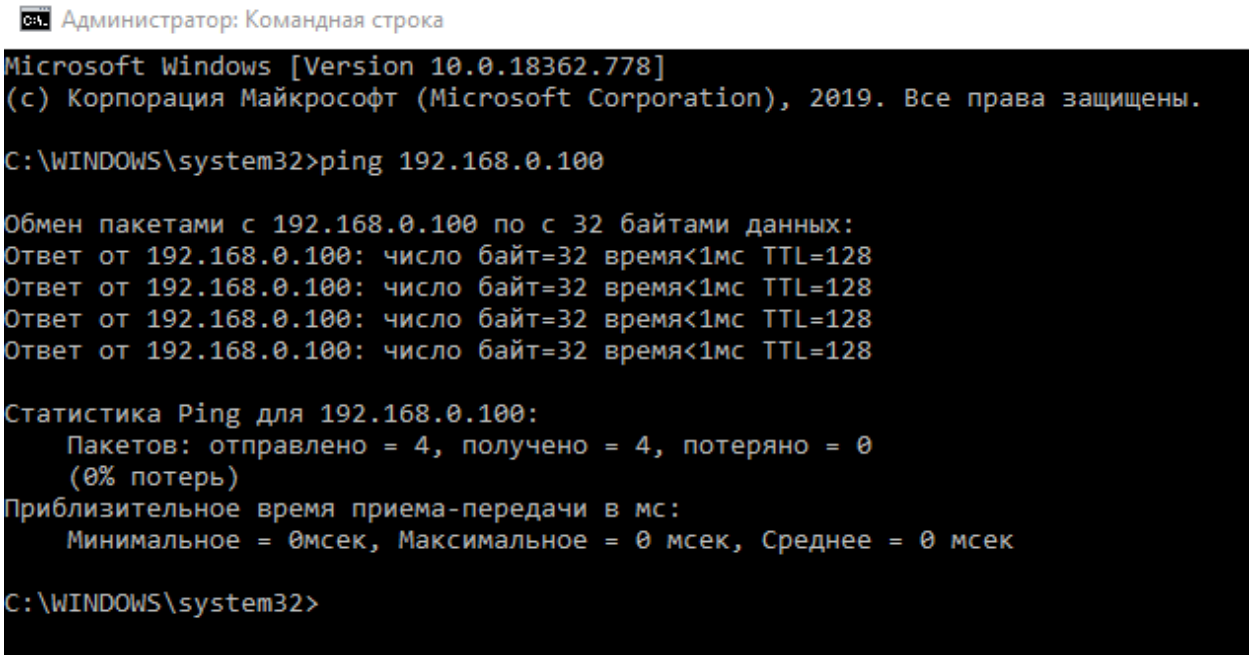


Рис. 3.8. Виконання команди ping

На рис. 3.8 показано, що всі відправлені пакети дійшли до отримувача, а це свідчить про те, що сервер створився та видимий в локальній мережі.

Далі на комп'ютері, який потрібно синхронізувати з нашим сервером, заходимо в «Налаштування дати та часу», встановлюємо значення «Встановлювати час автоматично» в статус «Увімкнено», а в розділі «Часовий пояс» встановлюємо значення «(UTC) Час в форматі UTC». Після проведених дій, сторінка «Налаштування дати та часу» має прийняти вигляд, як показано рис. 3.9.

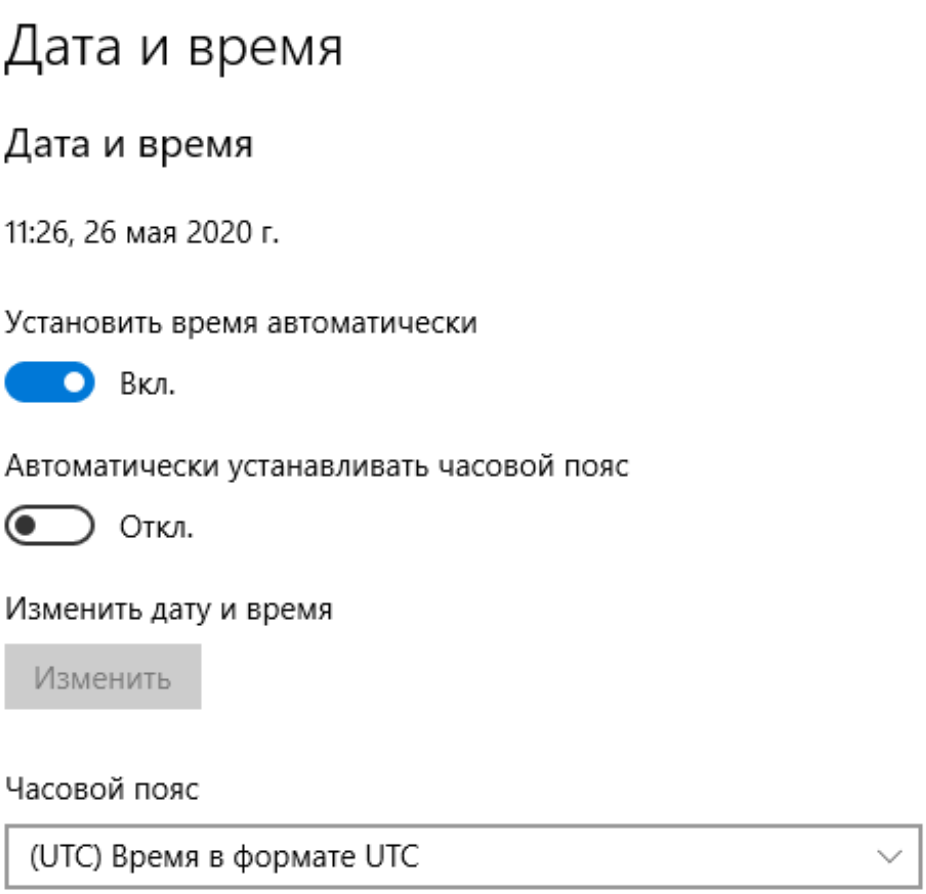


Рис. 3.9. Сторінка «Налаштування дати та часу»

Тепер необхідно налаштувати комп’ютер, щоб він звертався за даними про дату та час до нашого сервера. Для цього потрібно використати програму Network Time.

Встановивши та запустивши програму потрібно перейти в «Налаштування» («Settings»). В рядку «Time Servers» ввести IP-адресу серверу, з якого хочемо брати дані про дату та час (в даному випадку 192.168.0.100), вибирати протокол SNTP та номер порту 123. Також можна встановити інтервал опитування сервера та інші запропоновані налаштування. Після проведення описаних дій, вікно «Налаштування» має мати вигляд, як показано на рис. 3.10.

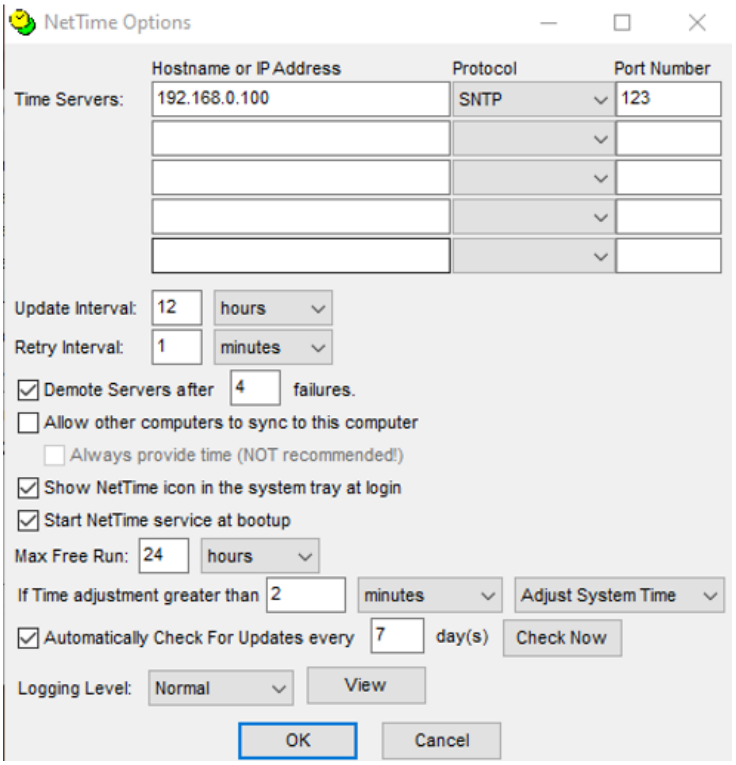


Рис. 3.10. Вікно «Налаштування» програми Network Time

Після проведення налаштувань необхідно зберегти зміни натиснувши кнопку «ОК».

Далі натискаємо кнопку «Update Now», щоб запустити процес синхронізації, та спостерігаємо за повідомленнями, що виводяться в верхній частині вікна програми. Якщо синхронізація пройшла успішно, то виведеться повідомлення «Time is synchronized» (рис. 3.11).

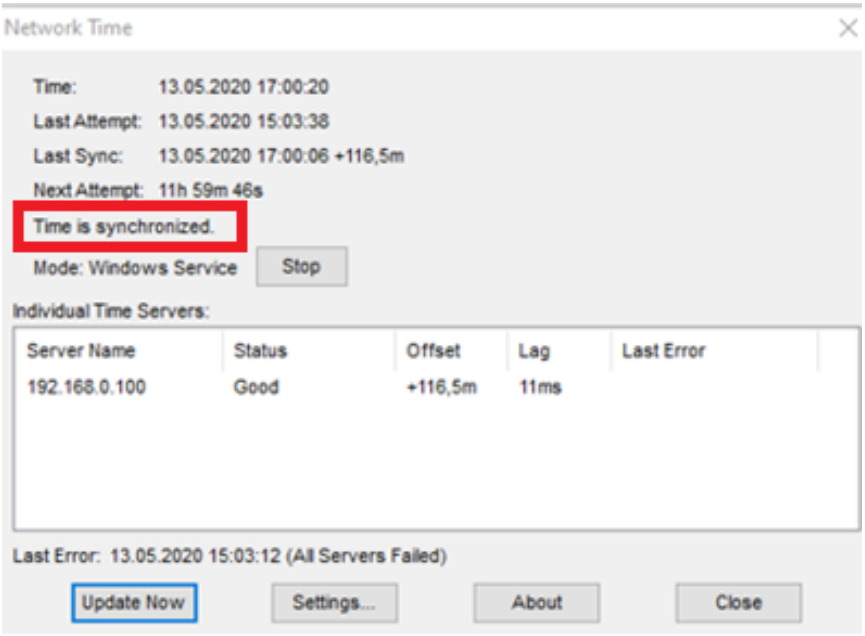


Рис. 3.11. Головне вікно програми Network Time

Тепер можна закрити програму Network Time. Всі налаштування збереглися в системі і наступна синхронізація з сервером відбудеться через проміжок часу, який був вказаний в налаштуваннях.

3.2 Демонстрація роботи системи єдиного часу

В даному пункті буде продемонстровано роботу розробленої системи єдиного часу.

Для демонстрації роботи обираємо не зайняту локальну IP-адресу 192.168.0.110 та визначаємо MAC-адресу мережевого модуля. Ці значення записуємо в програмний код (рис. 3.12).

```
IPAddress ipAddrs(192, 168, 0, 110);
byte macAddrs[] = { 0x00, 0xAA, 0xBB, 0xCC, 0xDE, 0x02 };
```

Рис. 3.12. Записані значення IP-адреси та MAC-адреси

Час в коді встановлюємо в ручну, щоб краще було видно результат роботи системи (рис. 3.13). Аргументи 2020, 5, 26, 17, 30, 0 відповідають року, місяцю, дню місяця, годинам, хвилинам та секундам відповідно.

```
void writeTimeToRTC(){
  //rtc.adjust(DateTime(__DATE__, __TIME__)); //запитуємо дані з системи
  //rtc.adjust(DateTime(years, months, days, hours, minutes, seconds)); //передаємо через змінні
  rtc.adjust(DateTime(2020, 5, 26, 17, 30, 0));//вводимо в ручну
}
```

Рис. 3.13. Встановлення значень дати та часу

В функцію loop() (рис. 3.14), крім функції receivingAndDataSending(), записуємо ще функцію printToSerialPort(), щоб продемонструвати, що встановлені значення дати та часу встановились коректно і відлік пішов.

```
void loop () {
  printToSerialPort();
  receivingAndDataSending();
}
```

Рис. 3.14. Тіло функції loop()

Тепер вибираємо порт (рис. 3.15), до якого підключена наша система, та завантажуємо програмний код в мікроконтролер.

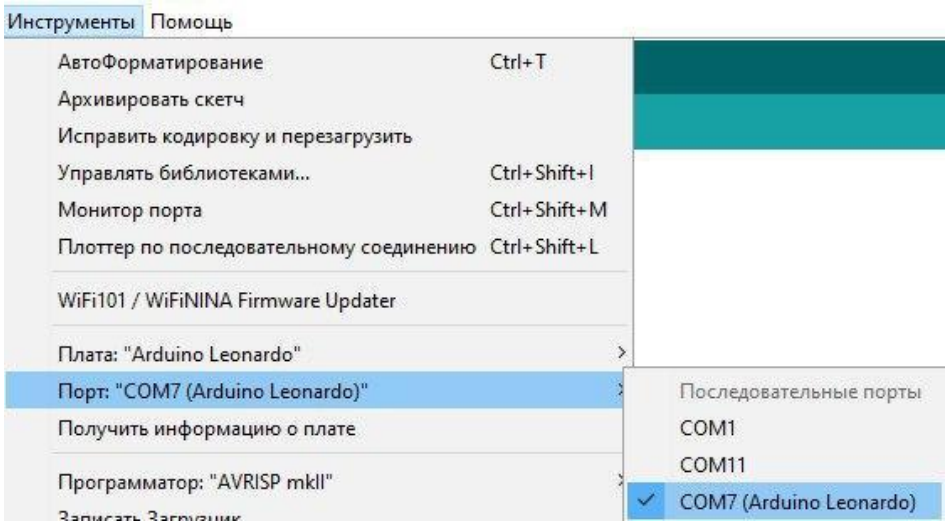


Рис. 3.15. Вибираємо порт

Після закінчення процесу завантаження, відкриваємо монітор порту і перевіряємо, чи коректно встановились записані в коді значення та чи пішов відлік часу (рис. 3.16).

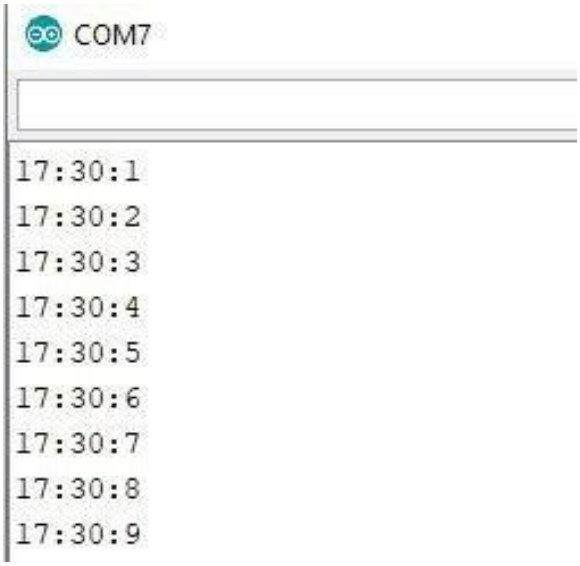


Рис. 3.16. Монітор порту

На рис. 3.16 видно, що вказані нами значення встановились коректно і відлік пішов.

Тепер виконаємо команду ping, щоб перевірити, чи створився сервер (рис. 3.17).

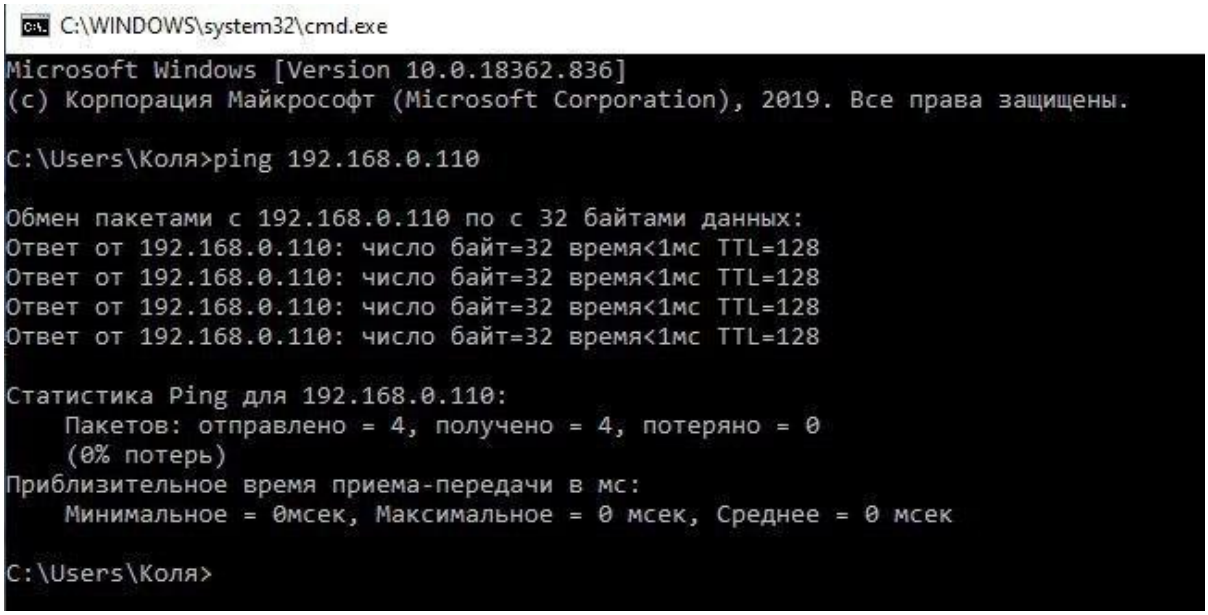


Рис. 3.17. Результат виконання команди ping

Підключимося до мережі з іншого персонального комп'ютера по Wi-Fi та спробуємо ще раз виконати команду ping (рис. 3.18).

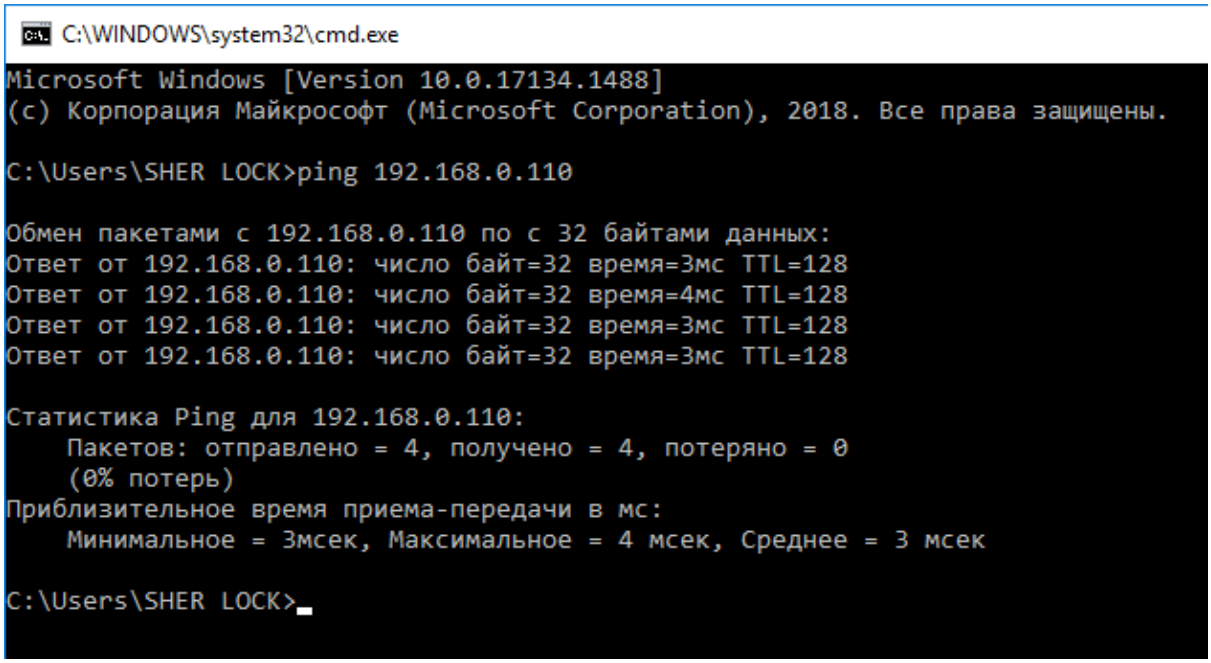


Рис. 3.18. Результат виконання команди ping

На рис. 3.17 та 3.18 видно, що втрат пакетів немає, отже можна зробити висновок, що сервер створився і видимий в локальній мережі.

Далі переходимо в «Налаштування дати та часу» та виконуємо дії, що стосуються даного вікна, описані в першому пункті даного розділу. Результат проведених дій показано на рис. 3.19.

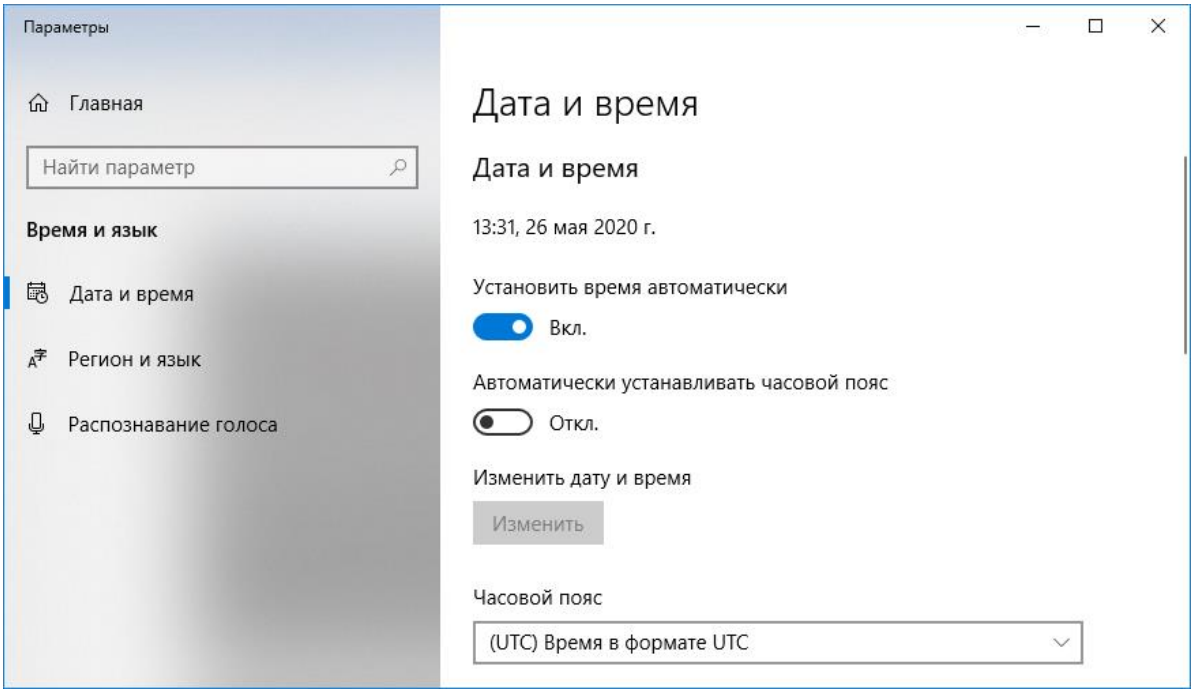


Рис. 3.19. Результат проведених налаштувань

Тепер запускаємо програму Network Time (рис. 3.20)

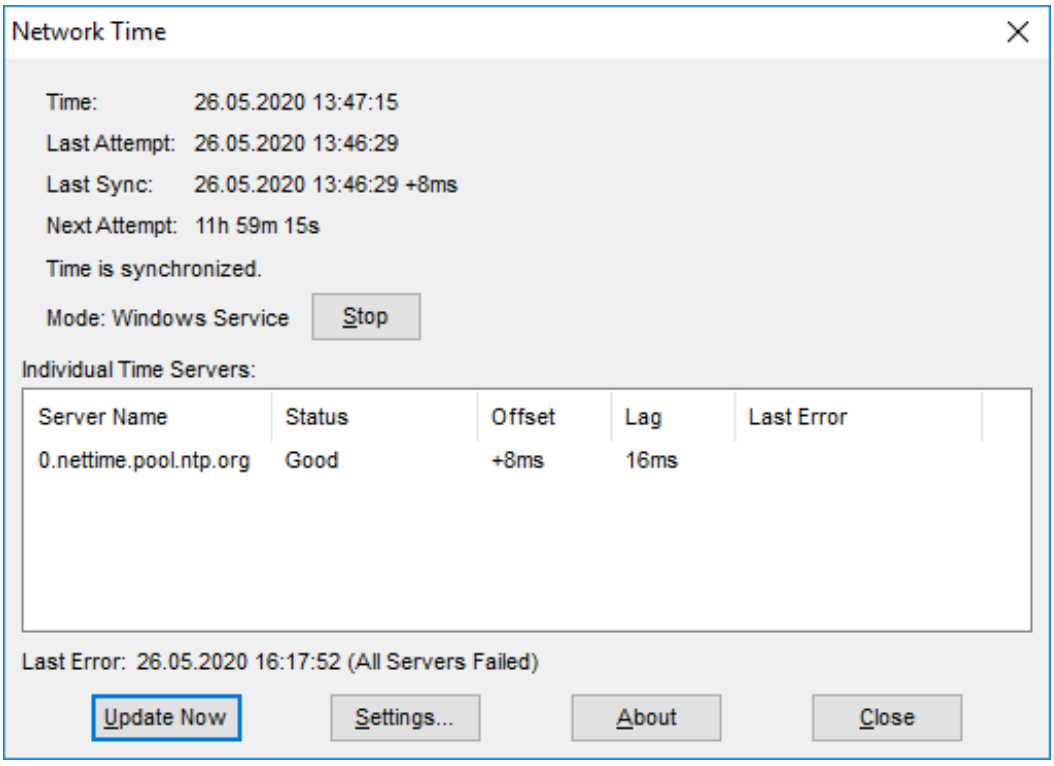


Рис. 3.20. Головне вікно програми Network Time

Далі переходимо в «Налаштування» («Settings») та виконуємо дії, що стосуються даного вікна, описані в першому пункті даного розділу. Результат проведених налаштувань показано на рис. 3.21.

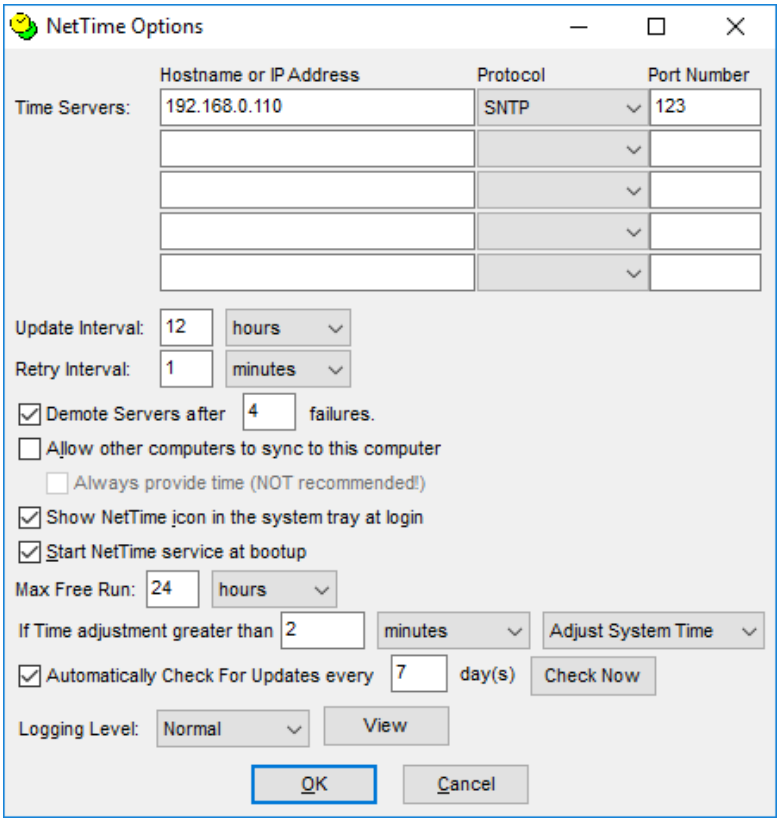


Рис. 3.21. Вікно «Settings» програми Network Time

Зберігаємо зміни, повертаємося на головне вікно програми та натискаємо кнопку «Update Now». На рис. 3.22 приведено результат успішної синхронізації.

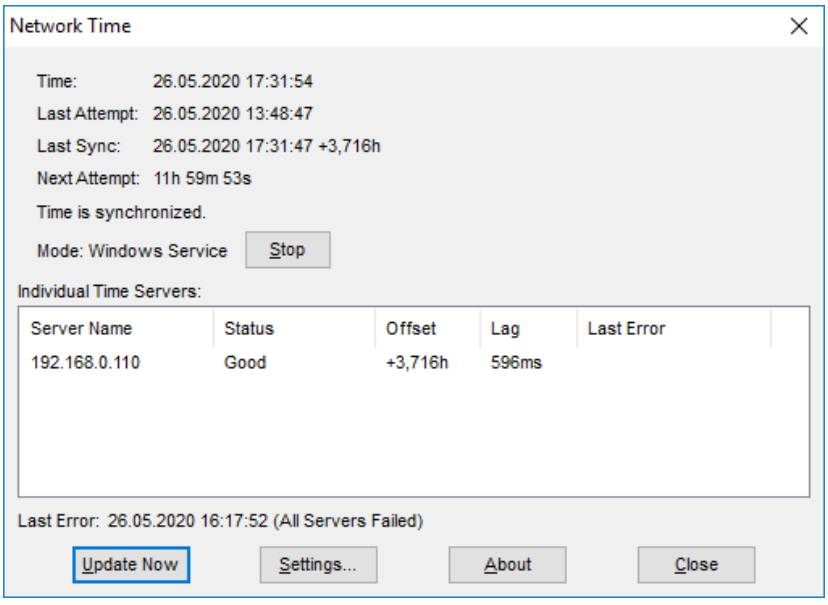


Рис. 3.22. Результат успішної синхронізації

Для перевірки збігання дати та часу в системі з тою, яка показана в вікні програми Network Time, переходимо в «Налаштування дати та часу» та перевіряємо значення (рис. 3.23).

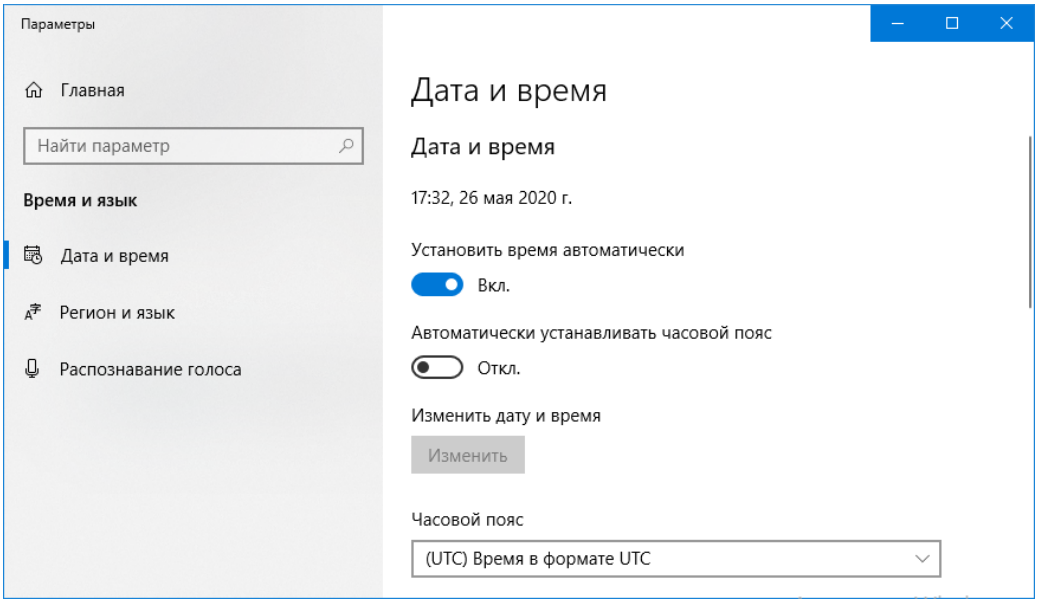


Рис. 3.23. Вікно «Налаштування дати та часу»

На рис. 3.23 видно, що дата та час збігаються.

Синхронізація дати та часу з сервером виконана успішно, отже розроблена система єдиного часу працює коректно.

Висновки з третього розділу

В даному розділі дипломної роботи було описано, як працювати з системою єдиного часу, які налаштування потрібно проводити та як перевіряти працездатність системи. Також, було проведено демонстрацію роботи розробленої системи єдиного часу.

ВИСНОВКИ

В першому розділі дипломної роботи було опрацьовано теоретичний матеріал стосовно розробки системи єдиного часу: визначено, які задачі вимагають використання системи єдиного часу; проаналізовано існуючі системи; описано способи синхронізації часу та структурну схему; розглянуто програмні та апаратні засоби для реалізації системи.

В другому розділі дипломної роботи було проведено розробку системи єдиного часу: визначено вузли структурної схеми та джерело отримання корисної інформації про дату та час; розроблено алгоритм роботи системи єдиного часу та створено функціональну схему; наведено процес підключення (з'єднання) елементів системи; розроблено програмний код мікроконтролера.

В третьому розділі дипломної роботи було описано, як працювати з системою єдиного часу, які налаштування потрібно проводити та як перевіряти працездатність системи. Також, було проведено демонстрацію роботи розробленої системи єдиного часу.

Результатом роботи є пристрій (система), який працює в режимі прослуховування та очікує на запит на отримання інформації про дату та час. Якщо такий запит був отриманий, система формує пакет-відповідь з необхідною інформацією та передає його запитувачу.

На основі результатів проведеної демонстрації роботи розробленої системи єдиного часу, можна зробити висновок, що дана система працює коректно.

СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ

1. Система единого времени «Тахион» [Електронний ресурс]. – Режим доступу до ресурсу: http://peleng.ru/products/catalog/sev/

2. Система времени «PLC2013» [Електронний ресурс]. – Режим доступу до ресурсу: http://timeserver.plcgroup.net/

3. Сервер точного времени Sinhron-M500 [Електронний ресурс]. – Режим доступу до ресурсу: http://krafttera.com/?page_id=157

4. Time from NPL (MSF) [Електронний ресурс]. – Режим доступу до ресурсу: https://en.wikipedia.org/wiki/Time_from_NPL_(MSF)?fbclid=IwAR2terZ6k7wYT6t8bUtCPH53dVj6qsfrCqYa_3vngyJImAIXnB6yOFJGOpk

5. WWVB [Електронний ресурс]. – Режим доступу до ресурсу: https://en.wikipedia.org/wiki/WWVB?fbclid=IwAR2iM7fGzOqPQpGoNQpRkuRHxiCs-UJFHLve8r10V4O6jTi99yKWIbXJ9AM

6. JJY [Електронний ресурс]. – Режим доступу до ресурсу: https://ru.wikipedia.org/wiki/JJY?fbclid=IwAR3nMPloMjdGnAFNP_bmZHXw-V5Vjr_PTlGzUQ9cPwNSzj9rwEUUiNi4VV8

7. RWM [Електронний ресурс]. – Режим доступу до ресурсу: https://ru.wikipedia.org/wiki/RWM?fbclid=IwAR1Ke-xoBMTbQBJp-iPWt70IHZTX8F3MgtcGjaj6iOjF6-MTPQMQdMSg8c8

8. DCF77 [Електронний ресурс]. – Режим доступу до ресурсу: https://ru.wikipedia.org/wiki/DCF77?fbclid=IwAR0xh4oj6yvVL8tW_6_zGMEeI6tQqZ5LUZJCshRCzllumR62lhWmG9r7I-E

9. Синхронизация точного времени. Стандарт IEEE 1588 [Електронний ресурс]. – Режим доступу до ресурсу: https://moxa.ru/tehnologii/power_systems/sinhronizaciya-tochnogo-vremeni-standart-ieee-1588/?fbclid=IwAR1v-oiaHh78EhI4OBg57ZY8if44Mw1k2dc0p6PK3Earm9AFwIDDclLKKxU

10. Pulse-per-second signal [Електронний ресурс]. – Режим доступу до ресурсу: https://en.wikipedia.org/wiki/Pulse-per-second_signal

11. IRIG timecode [Електронний ресурс]. – Режим доступу до ресурсу: https://en.wikipedia.org/wiki/IRIG_timecode

12. Протоколы синхронизации времени: NTP, IRIG, AFNOR [Електронний ресурс]. – Режим доступу до ресурсу: https://ptime.ru/timeprotocols.html

13. Алгоритм Марзулло [Електронний ресурс]. – Режим доступу до ресурсу:

https://ru.wikipedia.org/wiki/%D0%90%D0%BB%D0%B3%D0%BE%D1%80%D0%B8%D1%82%D0%BC_%D0%9C%D0%B0%D1%80%D0%B7%D1%83%D0%BB%D0%BB%D0%BE?fbclid=IwAR2Lzlq_WAe3C-YsSUT-w1MEwLiPehlpjco4fmfERr7cqafouEcJmO1f6WE

14. Протоколы синхронизации времени (NTP, SNTP, DAYTIME, TIME) [Електронний ресурс]. – Режим доступу до ресурсу:

http://time.in.ua/ntp.html?fbclid=IwAR3silvGrhjTpmBfDVz6UPNmI89DLqy9nfS-lAPLM0B52-x8aJj-P5UrUs0

15. Синхронизация времени согласно стандарту IEEE 1588 [Електронний ресурс]. – Режим доступу до ресурсу:

http://intent.gigatran.com/article/?id=118882&fbclid=IwAR2iM7fGzOqPQpGoNQpRkuRHxiCs-UJFHLve8r10V4O6jTi99yKWIbXJ9AM

16. ГЛОНАСС и GPS – в чем разница? [Електронний ресурс]. – Режим доступу до ресурсу: https://mapon.com.ua/glonass-i-gps-v-chem-raznica/

17. Спутниковая навигация: GPS, ГЛОНАСС и другие [Електронний ресурс]. – Режим доступу до ресурсу: https://itc.ua/articles/sputnikovaya-navigatsiya-gps-glonass-i-drugie/

18. Система передачи эталонных сигналов» [Електронний ресурс]. – Режим доступу до ресурсу:

https://web.archive.org/web/20101209082539/http://santel-group.com/_plugins/content/content.php?content.20

19. Программы для микроконтроллеров [Електронний ресурс]. – Режим доступу до ресурсу: https://cxem.net/software/soft_mcu.php

20. Arduino IDE [Електронний ресурс]. – Режим доступу до ресурсу: https://en.wikipedia.org/wiki/Arduino_IDE

21. Аппаратная часть платформы Arduino [Електронний ресурс]. – Режим доступу до ресурсу: http://arduino.ru/Hardware

22. Часы реального времени DS3231 [Електронний ресурс]. – Режим доступу до ресурсу: https://3d-diy.ru/wiki/arduino-moduli/chasy-realnogo-vremeni-ds3231/

23. Обзор часов реального времени DS1307 [Електронний ресурс]. – Режим доступу до ресурсу: https://robotchip.ru/obzor-chasov-realnogo-vremeni-ds1307/

ДОДАТКИ