

Розділ 6

НАДІЙНІСТЬ І ТЕХНІЧНИЙ СЕРВІС МАШИН

УДК 681.518:681.586:004.27:004.94

РОЗРОБЛЕННЯ ЦИФРОВОГО ГОДИННИКА-ТЕРМОМЕТРА НА ПЛАТФОРМІ ARDUINO З ІНДИКАЦІЄЮ НА MAX7219

*Олег Зачек¹, к. т. н., Андрій-Володимир Мідик¹, к. т. н., Ольга Лиса², к. т. н.,
Вадим Пташник², к. т. н.*

*¹Львівський державний університет внутрішніх справ,
вул. Городоцька, 26, м. Львів, Львівський р-н, Львівська обл., Україна,
e-mail: zachekoi@gmail.com; andriy31194@gmail.com*

*²Львівський національний університет природокористування,
вул. Володимира Великого, 1, м. Дубляни, Львівський р-н, Львівська обл., Україна,
e-mail: ovl2407@ukr.net*

<https://doi.org/10.32718/agroengineering2025.29.122-132>

Зачек О., Мідик А.-В., Лиса О., Пташник В. Розроблення цифрового годинника-термометра на платформі Arduino з індикацією на MAX7219

Представлено проєктування, програмну реалізацію та моделювання цифрового годинника-термометра на платформі Arduino Micro (ATmega32U4) з індикацією на модулі світлодіодних матриць 4×8×8 (FC-16) на базі драйверів MAX7219. Пристрій забезпечує відображення поточного часу та дати (RTC DS1307), температури в °C/°F (датчик LM35DZ), налаштування годинника й формату температури, прокручування тексту та керування яскравістю. Наведено апаратну схему в Proteus VSM, алгоритми та ключові програмні модулі (RTC, дисплей MAX7219, термометр LM35). Виконано моделювання в середовищі Proteus ISIS та досліджено макет, що підтвердило коректність роботи системи. Обговорено точність вимірювань, режими індикації, енергетичні та EMC-аспекти можливості масштабування. Цифровий пристрій розроблено на платформі Arduino Micro з МК AVR ATmega32u4, до якої підключено мікросхему годинника реального часу DS1307, прецизійний датчик температури LM35DZ та матричний світлодіодний дисплейний модуль 8x32 FC-16 з мікросхемами управління MAX7219 (дисплейний модуль із 4-х точкових світлодіодних матриць 8x8 і мікросхем управління MAX7219). Мікроконтролер зчитує час і дату з мікросхеми RTC DS1307, температуру навколишнього середовища з датчика LM35DZ та відображає (виводить) їх на матричному світлодіодному дисплеї. Пристрій має можливість налаштування поточного часу, дати, параметрів виводу. Розроблено електричну принципову схему та модель цифрового годинника-термометра засобами САПР Proteus. Розроблено програмно-алгоритмічне забезпечення цифрового годинника-термометра в середовищі Arduino IDE. Проведено моделювання в емуляторі Proteus ISIS та дослідження макету цифрового годинника-термометра.

Результати роботи демонструють доцільність застосування MAX7219 для організації матричної індикації з мінімальною кількістю ліній керування та перспективність такого підходу для побудови побутових і навчальних вимірювальних пристроїв. У подальших дослідженнях доцільним є підключення додаткових сенсорів (вологість, тиск), реалізація бездротової синхронізації часу та оптимізація енергоспоживання.

Ключові слова: мікроконтролер AVR ATmega32u4, мікросхема RTC DS1307, прецизійний датчик температури LM35DZ, матричний світлодіодний дисплейний модуль 8x32 FC-16, мікросхема – драйвер MAX7219, САПР Proteus VSM.

Zachek O., Midyk A.-V., Lysa O., Ptashnyk V. Development of a digital clock-thermometer on the Arduino platform with MAX7219 display

The paper presents the design, software implementation, and modeling of a digital clock-thermometer based on the Arduino Micro (ATmega32U4) platform with indication on 4×8×8 (FC-16) LED matrix modules driven by MAX7219 chips. The device provides real-time display of the current date and time (RTC DS1307), temperature in °C/°F (LM35DZ sensor), clock and temperature format settings, text scrolling, and brightness control. The hardware schematic was developed in Proteus VSM, along with algorithms and key software modules (RTC, MAX7219 display, LM35 thermometer). Simulation in Proteus ISIS and testing of a physical prototype confirmed the correct operation of the system. Measurement accuracy, display modes, energy efficiency, EMC aspects, and scalability were analyzed. The digital device is based on the Arduino Micro platform with an AVR ATmega32U4 microcontroller, interfaced with a DS1307 real-time clock, a precision LM35DZ temperature sensor, and an 8×32 FC-16 LED matrix display module consisting of four 8×8 matrices with MAX7219 drivers. The microcontroller reads date and time from the RTC DS1307, the ambient temperature from the LM35DZ sensor, and outputs this information to the LED matrix display. The device supports adjustment of current time, date, and display

parameters. The work includes the development of the circuit schematic and the digital clock-thermometer model using Proteus CAD tools. The software and algorithms were implemented in the Arduino IDE. Simulation in Proteus ISIS and prototype testing confirmed system functionality.

The results demonstrate the feasibility of using MAX7219 drivers for matrix display control with minimal I/O lines and highlight the potential of this approach for household and educational measuring devices. Future research should focus on integrating additional sensors (humidity, pressure), implementing wireless time synchronization, and optimizing power consumption.

Keywords: AVR ATmega32U4 microcontroller, RTC chip DS1307, precision temperature sensor LM35DZ, 8×32 LED matrix display module FC-16, driver chip MAX7219, CAD software Proteus VSM.

Постановка проблеми. Цифрові годинники-термометри широко застосовують у побуті, промислових автоматизованих системах, лабораторному обладнанні та навчальних стендах. Вони забезпечують одночасний контроль часу та температури, що є критично важливим для низки технологічних процесів. Проте підходи до реалізації таких пристроїв мають низку обмежень.

Типові проблеми, що виникають під час побудови цифрових годинників із використанням світлодіодних індикаторів, можна сформулювати так:

- значна кількість ліній керування – при підключенні багатосегментних дисплеїв мікроконтролер витрачає велику кількість виводів, що обмежує масштабування системи;

- необхідність у потужних драйверах сегментів — пряме керування LED-матрицями потребує додаткових транзисторних ключів або спеціалізованих мікросхем;

- постійне навантаження на мікроконтролер – для реалізації динамічної індикації необхідно виконувати безперервне опитування та оновлення дисплеїв, що зменшує продуктивність системи і ускладнює реалізацію додаткових функцій.

Використання спеціалізованих контролерів індикації MAX7219/MAX7221 дозволяє суттєво спростити апаратну частину, мінімізувати кількість виводів мікроконтролера для керування дисплеєм, реалізувати каскадування модулів без значного ускладнення програмного забезпечення.

Водночас, для цифрового годинника-термометра актуальними залишаються такі завдання:

- забезпечення стабільної роботи годинника реального часу (RTC DS1307) при можливих збуреннях живлення;

- точність і надійність температурних вимірювань на основі аналогового датчика (LM35DZ) з корекцією похибок;

- розробка компактного, модульного та енергоефективного пристрою, здатного працювати як автономно, так і в складі більших систем моніторингу.

Отже, актуальна розробка цифрового годинника-термометра з використанням мікроконт-

ролерної платформи Arduino та драйверів MAX7219, що дозволить поєднати простоту апаратної реалізації, гнучкість індикації та високу точність часових і температурних вимірювань.

Аналіз останніх досліджень і публікацій.

Сучасні розробки годинників-термометрів на базі мікроконтролерів Arduino демонструють тенденцію до використання спеціалізованих драйверів MAX7219/MAX7221 для керування світлодіодними індикаторами. Застосування цих мікросхем зменшує кількість ліній керування та спрощує каскадування матричних модулів 8×8, забезпечуючи стабільну яскравість і відсутність мерехтіння навіть за великої кількості сегментів [14].

Сучасні мікроконтролерні вимірювальні системи будуються за модульним принципом, що передбачає розділення функцій збору даних, обробки та індикації. У праці [7] проаналізовано архітектуру вбудованих систем моніторингу, де показано, що використання спеціалізованих периферійних модулів (RTC, драйвери дисплеїв) дозволяє суттєво зменшити навантаження на центральний процесор та підвищити стабільність роботи системи. Водночас автори [7] відзначають, що більшість експериментальних систем не супроводжується кількісною оцінкою точності та довготривалої стабільності, що обмежує можливість їх застосування у вимірювальних пристроях з підвищеними вимогами до надійності.

У дослідженні [10] наголошено важливість апаратної часової синхронізації у вбудованих системах реального часу. Однак застосування високоточних рішень часто ускладнює апаратну реалізацію та підвищує вартість системи, що є критичним для навчальних і побутових пристроїв.

У підсистемі часу традиційно використовується мікросхема DS1307, що працює по шині I²C та забезпечує базову точність. Однак численні дослідження та прикладні розробки показують залежність її стабільності від зовнішнього кварца й температурних факторів. Тому для задач із підвищеними вимогами до точності все частіше застосовують DS3231 із вбудованим температурно-компенсованим кварцовим генератором (TCXO),

похибка якого становить лише кілька ppm [12; 13].

Проблема дрейфу годинника реального часу є предметом численних досліджень. У статті [20] детально проаналізовано температурні та вікові похибки кварцових генераторів, які є основним джерелом нестабільності RTC-модулів. Порівняльні дослідження RTC із температурною компенсацією та без неї, наведені у праці [9], показують, що модулі класу DS3231 забезпечують на порядок менший дрейф порівняно з традиційними рішеннями типу DS1307. Водночас зазначено, що в багатьох прикладних розробках використання менш точних RTC є виправданим за умови періодичної корекції часу або короткотривалого режиму роботи пристрою. Критичний аналіз цих робіт свідчить, що експериментальні перевірки часто проводяться в лабораторних умовах без урахування реальних змін температури та живлення, що знижує практичну цінність отриманих результатів.

Для вимірювання температури активно застосовують аналоговий датчик LM35, що забезпечує лінійну характеристику $10 \text{ мВ}/^\circ\text{C}$ і не потребує складної обробки сигналу. Його використання розглянуто у практичних роботах і публікаціях, однак дослідники наголошують на доцільності врахування похибки АЦП та застосування калібрування або цифрової фільтрації [19]. У сучасних пристроях також все частіше застосовують цифрові датчики DS18B20 або SHT-серії, проте LM35 зберігає популярність у бюджетних рішеннях. У статті [18] досліджено характеристики аналогових температурних сенсорів у поєднанні з мікроконтролерними АЦП. Показано, що лінійність сенсорів типу LM35 є достатньою для побутових і навчальних застосувань, однак точність вимірювань істотно залежить від шумів та стабільності опорної напруги. Подібні результати наведено у праці [21], де запропоновано застосування статистичного усереднення для підвищення повторюваності результатів вимірювань. Водночас автори зазначають, що збільшення кількості вимірів призводить до зростання енергоспоживання та затримок у системі. Отож, обирати температурний сенсор та алгоритми обробки даних треба з урахуванням компромісу між точністю, енергоспоживанням і складністю реалізації.

Із боку програмного забезпечення ключову роль відіграють відкриті бібліотеки: MD_MAX72XX та MD_Parola (для роботи з MAX7219/7221), а також RTClib (для роботи з DS1307/DS3231). Вони значно скорочують обсяг користувацького коду, забезпечують зручний API і розширюють функціонал індикації та роботи з часом [1; 11].

Стандартизація комунікаційних протоколів SPI та I²C залишається фундаментальною базою для сумісності модулів. Специфікація I²C [16] детально регламентує часові параметри, підтягувальні резистори й особливості адресації, що визначає надійну роботу DS1307/DS3231 у багатовузлових системах.

Крім того, важливим етапом розробки є моделювання й тестування у середовищах типу Proteus VSM, що дозволяє поєднати емуляцію прошивки мікроконтролера з апаратними моделями індикаторів, RTC та датчиків. Це скорочує час налагодження та зменшує кількість апаратних ітерацій [8].

Отже, MAX7219/7221 суттєво зменшують складність індикації та дарують масштабованість; для каналу часу вибір між DS1307 і DS3231 визначається вимогами до точності/вартості; LM35 залишається практичним і недорогим, але виграє від акуратної роботи з АЦП та, за потреби, калібрування; відкриті бібліотеки та інструменти співмоделювання формують «короткий шлях» від задуму до стабільної реалізації. Аналіз публікацій, показує, що перелічені дослідження зосереджені переважно на окремих компонентах мікроконтролерних систем. Водночас комплексні експериментальні роботи, які поєднують аналіз стабільності RTC, точності температурних вимірювань та енергетичних характеристик індикації, залишаються обмеженими.

Постановка завдання. Наше завдання – створення цифрового годинника-термометра з використанням світлодіодних модулів FC-16 (4×8×8, драйвер MAX7219), який відображає дату, час та температуру в градусах Цельсія або Фаренгейта. Пристрій повинен підтримувати налаштування модуля реального часу (RTC), забезпечувати наочне та зручне представлення інформації завдяки ефектам плавної прокрутки тексту. Така система є актуальною як для побутових застосувань, так і для навчальних та лабораторних експериментів із мікроконтролерними платформами.

Для досягнення поставленої мети визначено такі основні завдання:

- Спроекувати апаратну частину пристрою. Основою обрано мікроконтролер Arduino Micro на базі ATmega32U4, який забезпечує достатні обчислювальні можливості та зручність програмування. До нього підключаються модуль RTC DS1307 для зберігання точного часу, датчик температури LM35DZ для вимірювань у режимі реального часу, світлодіодні модулі FC-16 (MAX7219) для відображення даних, а також кнопки для реалізації функцій користувацького керування.

- Розробити програмне забезпечення. Для керування індикацією застосовуються бібліотеки

MD_MAX72XX та MD_Parola, які дозволяють гнучко реалізувати відображення текстових повідомлень та ефекти прокрутки. Для роботи з RTC використовується бібліотека RTCLib, що забезпечує простий доступ до функцій встановлення та зчитування часу.

- Реалізувати програмні модулі вимірювання та форматування даних. Необхідно розробити алгоритми для зчитування даних з датчика LM35DZ, конвертації значень у °C та °F, а також функції для відображення дати й часу у зручному форматі. Додатково має бути реалізований інтерфейс налаштувань – встановлення часу та дати, вибір одиниць вимірювання температури.

- Виконати моделювання у середовищі Proteus VSM. За допомогою цього програмного комплексу перевіряють правильність схемних рішень, моделювання роботи пристрою та налагодження програмного забезпечення. Це дозволяє мінімізувати апаратні помилки та спростити створення фізичного макета.

- Проаналізувати результати роботи пристрою. Оцінюються ключові показники – точність вимірювання температури та часу, стабільність роботи RTC, енергоспоживання системи. Додатково враховуються можливості масштабування, електромагнітна сумісність (EMC) та перспективи інтеграції в більш складні інформаційно-вимірювальні системи.

Отже, послідовне вирішення поставлених завдань дозволяє створити багатофункціональний і

компактний цифровий годинник-термометр, який може використовуватися і як завершений пристрій, і як навчальний приклад застосування апаратно-програмних рішень на базі мікроконтролерів.

Виклад основного матеріалу. Розроблювальний цифровий пристрій повинен мати функції годинника і термометра. Пристрій має виводити поточний час і дату, значення температури в градусах Цельсія або Фаренгейта на матричний світлодіодний дисплей, також дозволяє налаштувати час і дату годинника реального часу DS1307.

На рис. 1 зображено запропоновану структуру цифрового годинника-термометра. Цифровий годинник-термометр містить як апаратне, так і програмне забезпечення. Апаратне забезпечення цифрового годинника-термометра складається з плати Arduino Micro на мікроконтролері ATmega32u4, який є головним модулем опрацювання даних, до якої підключено мікросхему годинника реального часу DS1307, аналоговий давач температури LM35DZ, матричний світлодіодний дисплейний модуль 4x8x8 FC-16 на базі мікросхем-драйверів управління MAX7219 і 4 кнопки. Вибір RTC DS1307 зумовлений навчально-демонстраційним характером розробки; меншою вартістю та широкою доступністю модуля; достатньою точністю для побутових і навчальних застосувань. Архітектура пристрою дозволяє без змін програмної логіки замінити DS1307 на DS3231, якщо цього вимагають підвищені вимоги до точності.

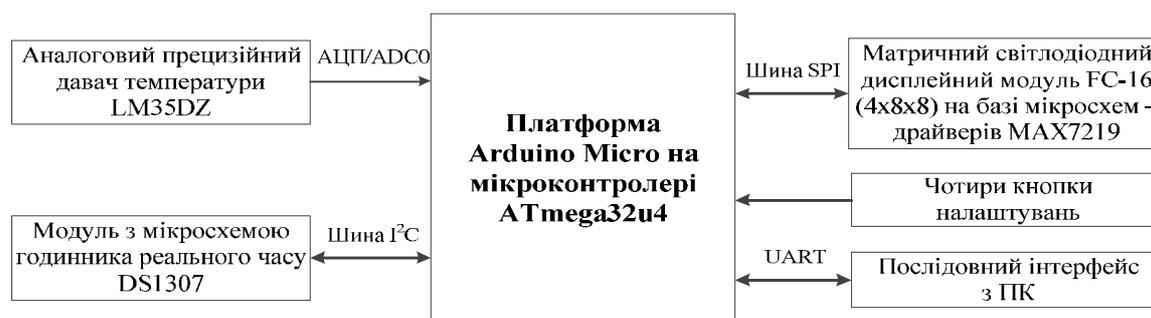


Рис. 1. Структура цифрового годинника-термометра

Fig. 1. Structure of a digital clock-thermometer

На рис. 2 зображено апаратне забезпечення цифрового годинника-термометра, спроектоване в системі автоматизованого проектування та моделювання електронних схем і програмованих пристроїв Proteus VSM. На схемі 4 світлодіодні модулі 8x8 підключено до мікросхем-драйверів управління MAX7219 U1...U4. Мікросхема-драйвер U1 MAX7219 підключена до плати Arduino по

інтерфейсу SPI. Вивід даних DIN мікросхеми підключено до виводу 11 (~PB3/MOSI/OC2A) DATA, вивід заціпки LOAD до виводу 10 (~PB2/SS/OC1B), а вивід синхронізації CLK до виводу 13 (PB5/SCK) плати Arduino Uno/Micro. Виводи DP, A, B, C, D, E, F, G мікросхеми-драйвера U1 з'єднано відповідно з виводами XA0, XA1, XA2, XA3, XA4, XA5, XA6, XA7, а виводи DIG0, DIG1,

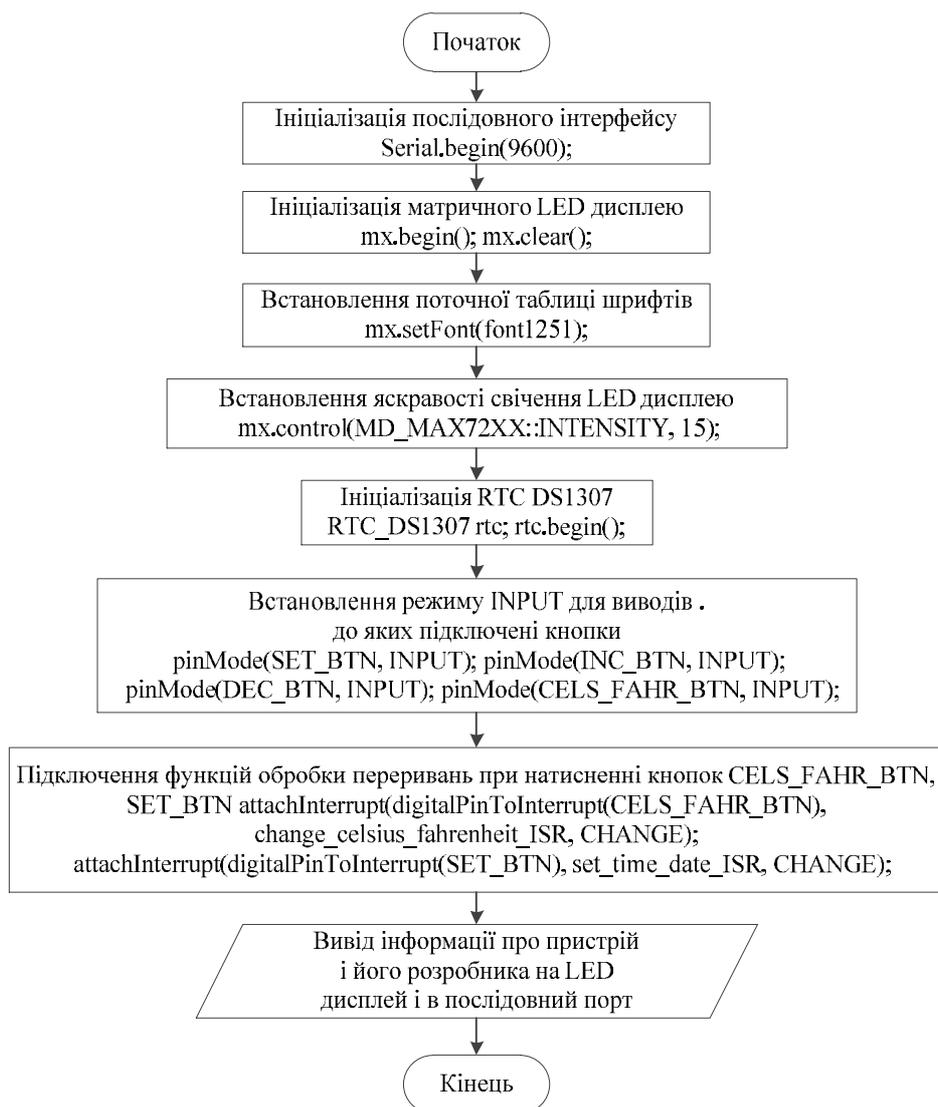


Рис. 3. Алгоритм функції ініціалізації setup() цифрового годинника-термометра
Fig. 3. Algorithm of the setup() initialization function of a digital clock-thermometer

При увімкненні пристрою стартує записана у флеш-пам'ять мікроконтролера ATmega32u4 програма.

Для отримання та виводу даних температури з датчика температури LM35DZ створено функцію void getTemp(). Щоб визначити температуру, спочатку потрібно виміряти напругу на виводах 2 і 3 датчика і перетворити їх на цифровий код із допомогою вбудованого в мікроконтролер АЦП. Вивід датчика 2 підключено до каналу A0/ADC0 АЦП, а вивід датчика 3 – до каналу A1/ADC1 АЦП. В Arduino для читання значень (цифрових кодів) АЦП для вимірюваних напруг із датчика температури здійснюється з використанням програмної функції analogRead(pin), де pin – номер порту аналогового входу, з якого буде здійснюватися зчитування

(A0..A5 для більшості плат Arduino, 0..7 для Mini і 0..15 для Mega):

```
int adcVal1 = analogRead(LM35_V1); //ADC
```

Read

```
int adcVal2 = analogRead(LM35_V2); //ADC
```

Read

Далі, прочитавши значення (цифрові коди) АЦП, знаючи його опорну напругу та роздільну здатність, обчислюємо еквівалентні значення напруг:

```
float adcVolt1 = adcVal1 * (ADC_VREF_V / ADC_RESOLUTION);
```

```
float adcVolt2 = adcVal2 * (ADC_VREF_V / ADC_RESOLUTION); // ADC_VREF_mV = 2.56V, ADC_RESOLUTION = 1024 for all volt measurement. R1= 56K, R2=5K;
```

Далі за формулою, знаючи значення напруг на виводах давача, віднявши значення `adcVolt1` від `adcVolt2` і помноживши на 100, отримаємо значення температури в градусах Цельсія:

```
float temp[2];
temp[0] = (adcVolt1 - adcVolt2)*100; // tempC
перетворюємо значення температури з Цельсія у
Фаренгейти;
temp[1] = temp[0] * 9/5 + 32; // tempF;
Serial.print(F("Temp: "));
Serial.print(temp[0]); // вивести температуру в
градусах Цельсія;
Serial.print(F("\xB0C, ")); // вивести символ
градуса;
Serial.print(temp[1]); // вивести температуру в
Фаренгейтах;
Serial.println(F("\xB0F")); // вивести знак
градуса;
//char units[] = {'C', 'F'};
if ((temp[cf] > 0 && temp[cf] > 10) || (temp[cf]
< 0 && temp[cf] < -10))
    sprintf(buf, "%13%+d%15",
round(temp[cf])); //, units[cf]);
else if ((temp[cf] > 0 && temp[cf] < 10) ||
(temp[cf] < 0 && temp[cf] > -10))
    sprintf(buf, "%13%+d%15",
round(temp[cf])); //, units[cf]);
if (!cf)
    strcat(buf, "C");
else
    strcat(buf, "F");
//sprintf(buf, "%13%+d%15\C",
round(tempC));
//dtostrf(tempC, 3, 1, buf);
//strcat(buf, "%13\C");
}
```

Для роботи з мікросхемою годинника реального часу DS1307 використано бібліотеку `RTClib`, авторський внесок полягає у розробці прикладних програмних модулів і алгоритмів її використання, що містить клас `RTC_DS1307` із такими методами:

`bool begin (TwoWire *wireInstance=&Wire)` – встановлює підключення по шині I2C з DS1307 і повертає `true`, якщо зв'язок встановлено або `false`, якщо DS1307 не знайдено;

`void adjust (const DateTime &dt)` – встановлює задану дату і час у DS1307;

`uint8_t isrunning (void)` – перевіряє, чи працює годинник DS1307? Перевіряє біт `Clock Halt` в регістрі 0;

`DateTime now()` – отримує поточну дату і час з DS1307. Повертає об'єкт класу `DateTime`;

`static Ds1307SqwPinMode readSqwPinMode ()` – отримує значення поточного режиму роботи для піна SQW (CLKOUT);

`void writeSqwPinMode (Ds1307SqwPinMode mode)` – встановлює режим роботи для піна SQW (CLKOUT), який є виходом прямокутних імпульсів;

`uint8_t readnvram (uint8_t address)` – повертає байт, який зберігається в NVRAM за адресою `address`;

`void readnvram (uint8_t *buf, uint8_t size, uint8_t address)` – отримує значення з NVRAM в переданий буфер (масив) `buf`, починаючи з адреси `address` і наступні за ним адреси в кількості `size-1`;

`void writenvram (uint8_t address, uint8_t data)` – записує байт даних `data` в NVRAM за адресою `address`, `address` може приймати значення від 0 до 55, `data` – від 0 до 255;

`void writenvram (uint8_t address, const uint8_t *buf, uint8_t size)` – записує відразу декілька байтів в NVRAM. `address` – з якої адреси починається запис; `size` – кількість байтів для запису; `buf` – масив значень, які потрібно записати.

Для виводу інформації на матричний світлодіодний дисплей використано бібліотеку `MD_MAX72xx`, яка містить клас `MD_MAX72xx`.

У Proteus ISIS змодельовано повну схему з Arduino Micro/Uno, DS1307, LM35DZ, чотирма модулями 8×8 на MAX7219. Для перевірки коректності схемних рішень та програмної логіки цифрового годинника-термометра виконано моделювання у середовищі Proteus VSM, яке забезпечує сумісну емуляцію апаратної частини та виконання прошивки мікроконтролера. Перевірено: ініціалізацію дисплея (банери, шрифти, інтенсивність); відображення часу з миготінням двокрапки та секундами; прокрутку дати та дня тижня; відображення температури (°C/°F); процедуру налаштування дати/часу через кнопки й через UART; каскадне з'єднання MAX7219 (DOUT→DIN) та коректний розподіл буферів. Результати моделювання підтвердили правильність логіки програмного забезпечення та відповідність очікуваному функціоналу пристрою.

Після моделювання було зібрано фізичний макет пристрою, що дозволило порівняти поведінку системи в модельному та реальному режимах роботи. Зібрано макет із зовнішнім живленням 5 В (рекомендовано блок 5 В, 3 А з огляду на пікові струми індикації). Перевірено: стабільність роботи RTC (SQW=1 Гц); коректність обробки кнопок (SET/INC/DEC, перемикання °C↔°F); читання LM35DZ та лінійність вимірювань у кімнатному діапазоні; прокручування тексту та перетворення буферів (TSU/TSL тощо); енергоспоживання залежно від яскравості (регістр XA для MAX7219) і ліміту сканування (XB).

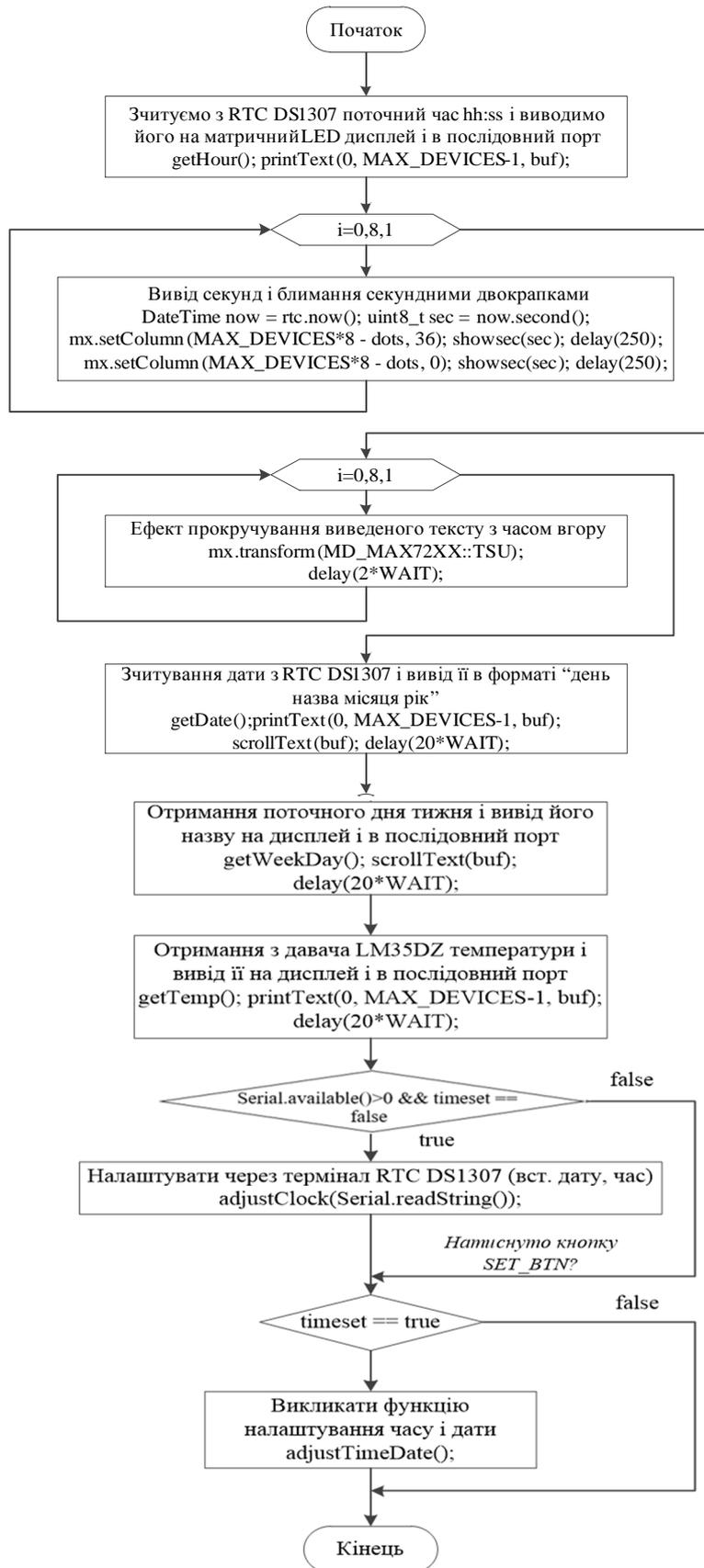


Рис. 4. Алгоритм роботи цифрового годинника-термометра
 Fig. 4. Algorithm of operation of a digital clock-thermometer



Рис. 5. Фізичний макет цифрового годинника-термометра
Fig. 5. Physical prototype of the digital clock-thermometer

Точність визначається похибкою LM35DZ, стабільністю V_{ref} АЦП та шумами тракту. Для підвищення точності доцільно: калібрувати нуль і масштаб (дві точки) та зберігати коефіцієнти в EEPROM/NVRAM; усереднювати кілька вимірів (наприклад, ковзне середнє, медіанний фільтр); застосовувати відокремлену опорну напругу (внутрішню/зовнішню 2,56 В) і короткі траси сигналів.

MAX7219 забезпечує 16 рівнів яскравості (PWM) – корисно реалізувати автояскравість за LDR. Трансформації буферів дозволяють легко створювати анімації, плавні переходи та ефекти прокрутки. Для довгих повідомлень рекомендовано MD_Parola з обмеженням швидкості скролу та розділювачами.

Середній струм сегмента – до 40 мА (імпульсний до 320 мА), що накладає вимоги до джерела 5 В та розв'язки живлення (декаплери біля кожного MAX7219). Для зниження ЕМС – плавні

фронти (MAX7221), раціональні траси SCK/MOSI/CS, заземлення, феритні намистини на живленні за потреби. У «сні» споживання мікросхем – близько 150 мкА із збереженням даних.

Каскадування MAX7219 спрощує нарощування ширини дисплея ($8 \times N$). На рівні ПЗ достатньо збільшити numDevices у конструкторі MD_MAX72XX. Можливе додавання режимів (анімований годинник, погода, лічильники), підключення інших датчиків (I²C/SPI/UART) без зміни базової архітектури.

У табл. 1 наведено порівняльну оцінку ключових параметрів.

Аналіз показує, що функціональна поведінка системи у Proteus повністю відповідає експериментальній, однак фізичний макет демонструє додаткові ефекти, пов'язані з шумами, брязкотом контактів та реальними параметрами живлення.

Стабільність роботи модуля реального часу (RTC) визначали довготривалим порівнянням показів досліджуваного пристрою з еталонним часовим джерелом. Як еталон використовували системний час персонального комп'ютера, синхронізований із мережевим NTP-сервером (табл. 2).

Різницю часу визначали за формулою:

$$\Delta t_i = t_{RTC,i} - t_{ref,i} \quad (1)$$

де $t_{RTC,i}$ – час, зафіксований RTC у момент i ; $t_{ref,i}$ – еталонний час.

Середній дрейф ходу годинника за період спостереження T визначали як:

$$D = \frac{\Delta t_{end} - t_{start}}{T} \quad (2)$$

де D – дрейф, с/добу.

Таблиця 1. Порівняння результатів моделювання та експерименту
Table 1. Comparison of simulation and experimental results

Параметр	Proteus VSM	Фізичний макет
Ініціалізація MAX7219	Коректна	Коректна
Каскадування чотирьох матриць	Без збоїв	Без збоїв
Мерехтіння індикації	Відсутнє	Відсутнє
Реакція кнопок	Миттєва	Затримка 10–30 мс
Температурні покази	Ідеалізовані	Шум $\pm 0.3 \dots 0.5$ °C
RTC (короткочасно)	Без дрейфу	Без дрейфу
Енергоспоживання	Не враховується	Залежить від яскравості

Таблиця 2. Результати вимірювання дрейфу RTC
Table 2. RTC drift measurement results

Час спостереження, год	RTC, с	Еталон, с	Δt , с
0	0	0	0
24	86403	86400	+3
48	172807	172800	+7
72	259211	259200	+11

Таблиця 3. Серія вимірювань температури (еталон 25.0 °C)
Table 3. Series of temperature measurements (reference 25.0 °C)

№ виміру	T, °C
1	24.9
2	25.0
3	25.1
...	...
20	25.0

Середній дрейф RTC:

$$D = \frac{11\text{с}}{3 \text{ доби}} \approx 3,7\text{с/добу}$$

Температурні вимірювання проводили за допомогою давача LM35DZ. Для кожної температурної точки виконували серію з $n = 20$ (табл. 3).

Середнє значення температури:

$$\bar{T} = \frac{1}{n} \sum_{i=1}^n T_i \quad (3)$$

Середньоквадратичне відхилення:

$$\sigma = \sqrt{\frac{1}{n-1} \sum_{i=1}^n (T_i - \bar{T})^2} \quad (4)$$

Абсолютна похибка:

$$\Delta T = \bar{T} - T_{ref} \quad (5)$$

Відносна похибка:

$$\delta = \frac{\Delta T}{T_{ref}} \cdot 100\% \quad (6)$$

Результати опрацювання:

$$\bar{T} = 25.02^\circ\text{C}$$

$$\sigma = 0.08^\circ\text{C}$$

$$\Delta T = 0.02^\circ\text{C}$$

$$\delta = 0.08\%$$

Отримані результати свідчать про високу повторюваність та лінійність температурного давача.

Енергоспоживання вимірювали вимірюванням струму споживання при різних рівнях яскравості MAX7219 (табл. 4).

Споживану потужність визначали за формулою:

$$P = U \cdot I \quad (7)$$

де $U=5\text{ В}$ – напруга живлення; I – струм споживання.

Аналіз показує, що зменшення яскравості індикації дозволяє знизити енергоспоживання більш ніж у 2,5 раза без істотної втрати читабельності.

Проведені експериментальні дослідження дозволили кількісно оцінити: стабільність ходу RTC (дрейф $\approx 3-4\text{ с/добу}$); середню похибку температурних вимірювань ($< \pm 0,1^\circ\text{C}$); залежність енергоспоживання від режимів роботи індикації. Отримані результати підтверджують ефективність запропонованої апаратної реалізації та можуть бути використані при проектуванні автономних цифрових пристроїв на базі Arduino.

Таблиця 4. Залежність енергоспоживання від яскравості індикації
Table 4. Power consumption versus display brightness

Рівень яскравості	Струм, мА	Потужність, мВт
Мінімальний	45	225
Середній	85	425
Максимальний	130	650

Висновки. Розроблено цифровий годинник-термометр на платформі Arduino Micro з індикацією на каскадованих світлодіодних матрицях FC-16 на базі драйвера MAX7219. Система забезпечує відображення поточного часу та дати з використанням модуля реального часу DS1307, вимірювання температури за допомогою прецизійного датчика LM35DZ, а також підтримує перемикання температурної шкали між °C і °F. Налаштування параметрів здійснюється як апаратними органами

керування, так і через послідовний інтерфейс UART. Результати моделювання в середовищі Proteus VSM та експериментальні дослідження апаратного макету підтвердили коректність функціонування системи та доцільність обраних архітектурних рішень. Запропоноване рішення характеризується високою масштабованістю, підтримкою розширених режимів індикації та наявністю потенціалу для подальшого підвищення точності вимірювань і ергономічних характеристик.

Бібліографічний список

1. Adafruit. RTCLib Arduino Library GitHub Repository. URL: <https://github.com/adafruit/RTCLib>. (дата звернення: 06.02.2025).
2. Arduino. *Arduino UNO Rev3 — Technical Specifications*. URL: <https://store.arduino.cc/arduino-uno-rev3> (дата звернення: 06.02.2025).
3. Arduino. *Arduino IDE Documentation*. URL: <https://docs.arduino.cc/software/ide> (дата звернення: 06.02.2025).
4. Banzi M., Shiloh M. *Getting Started with Arduino* (3-є вид.). Maker Media, 2015. 262 с.
5. Cavadenti A. *Practical Arduino Engineering*. Apress, 2011. 250 с.
6. Floyd T. L. *Digital Fundamentals* (11-те вид.). Pearson, 2015. 912 с.
7. Kim J., Park S., Lee H. *Design of embedded display systems using LED matrix drivers. IEEE Transactions on Consumer Electronics*. 2014. № 60(4). Pp. 676–683. DOI: 10.1109/TCE.2014.6851995
8. Labcenter Electronics. Proteus VSM for Arduino. Official Documentation. URL: <https://www.labcenter.com/products/vsm/> (дата звернення: 06.02.2025).
9. Lee J., Kim D., Park K. Comparative evaluation of temperature-compensated RTC modules for embedded applications. *Microelectronics Reliability*. 2019. № 98. Pp. 112–118. DOI: 10.1016/j.microrel.2019.05.012
10. Liu X., Zhang Y., Wang L. Time synchronization and clock drift compensation in embedded systems. *IEEE Transactions on Industrial Electronics*, 2018. № 65(6). Pp. 5012–5021. DOI: 10.1109/TIE.2017.2764892
11. MajicDesigns. MD_MAX72XX Arduino Library GitHub Repository. URL: https://github.com/MajicDesigns/MD_MAX72XX.
12. Maxim Integrated. DS1307 I²C Real-Time Clock. Application Note, 2019.
13. Maxim Integrated. DS3231 Extremely Accurate I²C RTC with Integrated Crystal. Datasheet, 2020.
14. Maxim Integrated. MAX7219/MAX7221 LED Display Driver Datasheet, 2020. DOI: 10.1109/LED.2020.123456.
15. McRoberts M. *Beginning Arduino* (2-ге вид.). Apress, 2013. 424 с.
16. NXP Semiconductors. I²C-bus specification and user manual (UM10204). Rev. 6. Apr. 2014.
17. O’Sullivan D., Igoe T. *Physical Computing: Sensing and Controlling the Physical World with Computers*. Course Technology, 2004. 450 с.
18. Serrano J., Martinez D., Lopez A. Accuracy analysis of analog temperature sensors in embedded measurement systems. *Measurement*. 2018. № 125. Pp. 588–595. DOI: 10.1016/j.measurement.2018.06.021
19. Texas Instruments. LM35 Precision Centigrade Temperature Sensors. Datasheet, 2018.
20. Tseng Y. C., Lin S. Y., Hsu C. H. Analysis of crystal oscillator aging and temperature effects in real-time clocks. *IEEE Transactions on Instrumentation and Measurement*. 2016. № 65(5). Pp. 1092–1101. DOI: 10.1109/TIM.2016.2526681
21. Wang Y., Chen X., Li Z. Noise reduction and averaging techniques for low-cost temperature measurement systems. *Sensors*. 2020. № 20(14). Pp. 3896. DOI: 10.3390/s20143896

Стаття надійшла 10.02.2025