

Міністерство освіти і науки України
Національний технічний університет України
«Київський Політехнічний Інститут ім. І. Сікорського»
Кафедра КЕОА

Звіт з виконання лабораторної роботи №1-4

З курсу «Автоматизація конструкторського проектування радіоелектронної апаратури»

На тему «Створення віртуального приладу для роботи з датчиком температури DS18B20 в середовищі LabVIEW»

Виконав:
студент IV курсу ФЕЛ
гр. ДК-61
Сільчук В.І.

Перевірив:
доцент Яганов П.О.

Зміст

1. Вступ.
2. Загальний опис пристрою. Температурний датчик DS18B20.
3. Опис реалізації створеного віртуального пристрою в середовищі LabVIEW.
4. Опис реалізації програми роботи з датчиком для плати Arduino UNO.
5. Висновок.
6. Додатки.

Вступ

В повсякденному житті, а також робочому процесі інженера часто може виникати потреба в вимірах температури навколишнього середовища, або ж певних матеріалів.

В ході виконання лабораторних робіт №1-4 в рамках курсу «Автоматизація конструкторського проектування радіоелектронної апаратури», в середовищі LabVIEW був створений віртуальний прилад (ВП), який дозволяє виконувати це завдання за допомогою датчика температури DS18B20, надаючи, при цьому, можливість відображення отриманих даних в зручному для сприйняття форматі та збереження їх в файл.

Загальний опис пристрою. Температурний датчик DS18B20

Зв'язок датчика температури DS18B20 з комп'ютером відбувається за допомогою плати Arduino UNO, з якою він зв'язаний за допомогою послідовного інтерфейсу 1-Wire. Для зручної роботи з датчиком використовуються загальнодоступні бібліотеки OneWire та DallasTemperature для платформи Arduino.

Цифровий датчик DS18B20 є одним з найбільш популярних температурних датчиків, часто він використовується в водонепроникному корпусі для вимірювання температури води або інших рідин.

DS18B20 — це цифровий температурний датчик, що володіє безліччю корисних функцій. Фактично, це цілий мікроконтролер, який може зберігати значення вимірювань, сигналізувати про вихід температури за встановлені межі (самі межі можна встановлювати і змінювати), міняти точність вимірювань, спосіб взаємодії з контролером і багато іншого. Все це — в дуже невеликому корпусі, який, до того ж, як було згадано вище, доступний в водонепроникному виконанні.

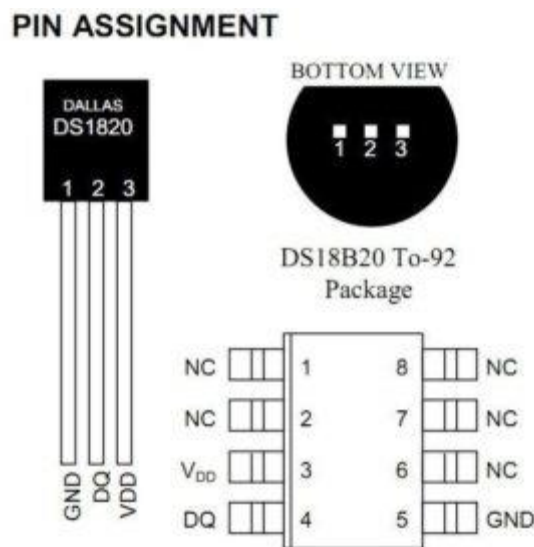


Рисунок 1 — Призначення виводів датчика.

Мікросхема має три виходи, з яких для даних використовується тільки один, два інших — це земля і живлення. Число провідників можна скоротити до двох, якщо використовувати схему з паразитним живленням, з'єднавши Vdd з землею. До одного виводу з даними можна підключити відразу декілька датчиків DS18B20 і в платі Arduino, при цьому, буде задіяний лише один пін.

Даний температурний датчик випускається в різноманітних корпусах. Можна вибрати один з трьох — 8-Pin SO, 8-Pin μ SOP, або 3-Pin TO-92. Останній є найбільш поширеним і виготовляється в спеціальному вологозахищеному корпусі, що дозволяє використовувати його під водою.

Особливості датчика температури DS18B20:

- похибка вимірювання не більше $0,5\text{ }^{\circ}\text{C}$ (для температур від $-10\text{ }^{\circ}\text{C}$ до $+85\text{ }^{\circ}\text{C}$), що забезпечує можливість точного визначення значень температури; відсутня також потреба в додатковому калібруванні;
- температурний діапазон вимірювань лежить в межах від $-55\text{ }^{\circ}\text{C}$ до $+125\text{ }^{\circ}\text{C}$;
- датчик живиться напругою від 3,3 до 5В;
- можна програмно задати максимальну роздільну здатність до $0,0625\text{ }^{\circ}\text{C}$, найбільша роздільна здатність — 12 біт;
- кожен пристрій має свій унікальний серійний код;
- не потрібні додаткові зовнішні елементи;
- можна підключити відразу до 127 датчиків до однієї лінії зв'язку;
- інформація передається по протоколу 1-Wire;
- для підключення до мікроконтролера потрібні тільки 3 провідники;
- час перетворення температури при 12-розрядній роздільній здатності — 750мс (макс.);

Існує також так званий режим паразитного живлення — в ньому відбувається живлення датчика безпосередньо від лінії зв'язку. Для підключення в цьому випадку потрібні тільки 2 провідники. Важливо, що в цьому режимі не гарантується коректна робота при температурах вище $100\text{ }^{\circ}\text{C}$. Режим паразитного живлення зазвичай застосовується для пристроїв з віддаленим температурним датчиком.

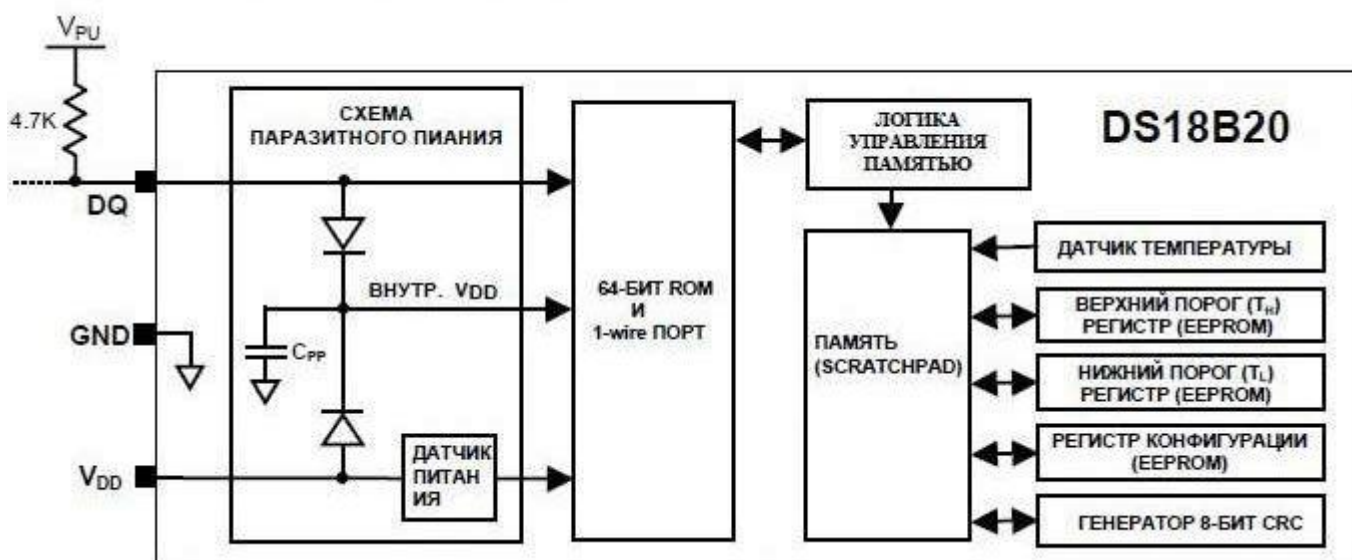


Рисунок 2 — Блок-схема датчика температури DS18B20.

Основним завданням DS18B20 є визначення температури і перетворення отриманого результату в цифровий вигляд. Ми можемо самостійно задати необхідну роздільну здатність, встановивши кількість бітів точності — 9, 10, 11 і 12. У цих випадках роздільні здатності будуть відповідно рівні $0,5\text{ }^{\circ}\text{C}$, $0,25\text{ }^{\circ}\text{C}$, $0,125\text{ }^{\circ}\text{C}$ і $0,0625\text{ }^{\circ}\text{C}$.

Під час включення живлення датчик знаходиться в стані спокою. Перш за все, для вимірювання контролер Arduino виконує команду «перетворення температури». Отриманий результат збережеться в 2 байтах регістру температури, після чого датчик повернеться в початковий стан спокою. Якщо схема підключена в режимі зовнішнього живлення, мікроконтролер контролює стан конвертування. Під час виконання команди лінія знаходиться в низькому стані, після закінчення — лінія переходить в високий стан. Такий метод не є допустимим при паразитному способі живлення, так як на шині постійно повинен зберігатися високий рівень сигналу.

Отримані температурні вимірювання зберігаються в SRAM-пам'яті датчика. 1 і 2 байти зберігають отримане значення температури, 3 і 4 зберігають межі вимірювання, 5 і 6 є зарезервованими, 7 і 8 використовуються для високоточного визначення температури, а останній 9 байт зберігає стійкий до перешкод CRC-код.

Для зв'язку з мікроконтролерами, як було вказано вище, датчик використовує протокол 1-Wire.

Загалом, обмін інформацією в 1-Wire відбувається завдяки таким операціям:

- 1) Ініціалізація — визначення послідовності сигналів, з яких починається вимірювання і інші операції. Ведучий пристрій подає імпульс скидання, після цього датчик повинен подати імпульс присутності, повідомляючи про готовність до виконання операції;
- 2) Запис даних — відбувається передача байту даних в датчик;
- 3) Читання даних — відбувається прийом байту з датчика;

До плати Arduino UNO датчик підключається наступним чином: GND з термодатчика приєднується до GND Arduino, Vdd підключається до 5V, Data — до будь-якого цифрового піну.

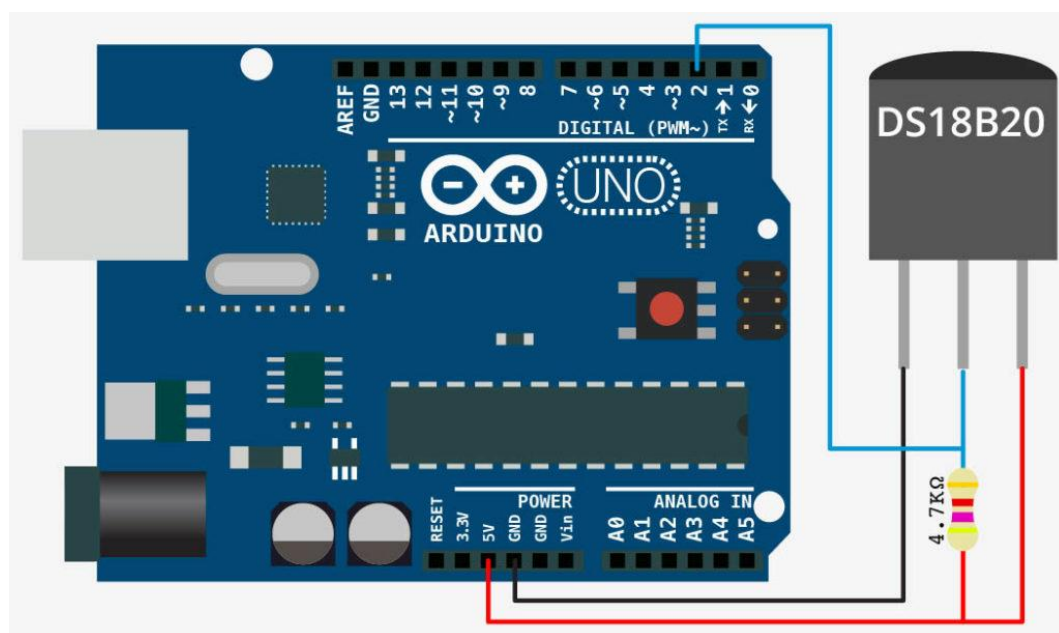


Рисунок 3 — Найпростіша схема підключення цифрового датчика DS18B20.

Як вже було згадано вище, для спрощення роботи з датчиком було використано бібліотеки OneWire та DallasTemperature, що дало змогу обійтись без складних маніпуляцій в роботі з послідовними протоколами, а також з тонкощами послідовностей конфігурування датчика температури.

Опис реалізації створеного віртуального пристрою в середовищі LabVIEW

Панель керування створеного в середовищі LabVIEW віртуального приладу має наступний вигляд:

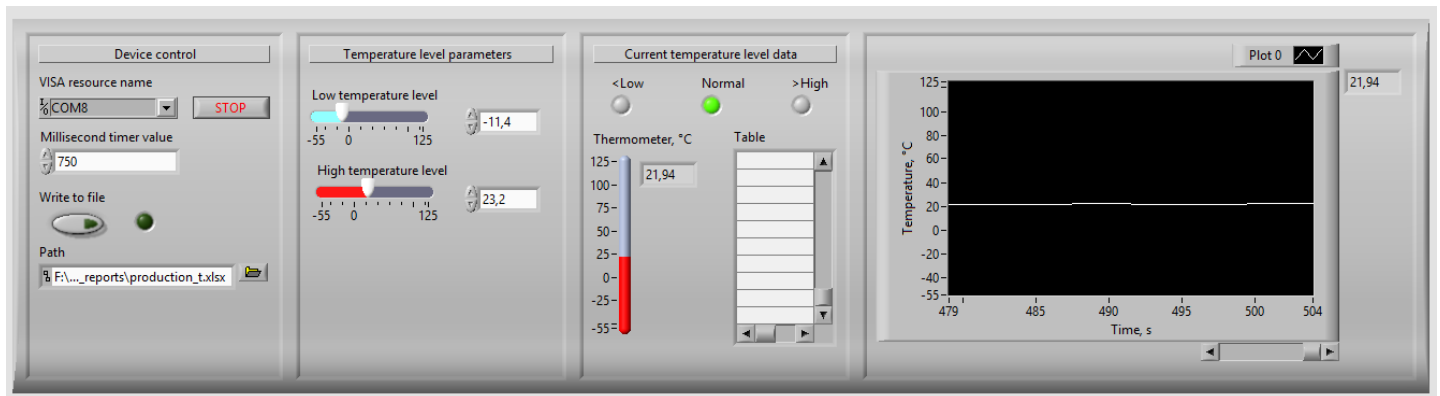


Рисунок 4 — Панель керування ВП.

Складається вона з наступних компонентів:

- 1) Елементи керування параметрами з'єднання через послідовний порт.

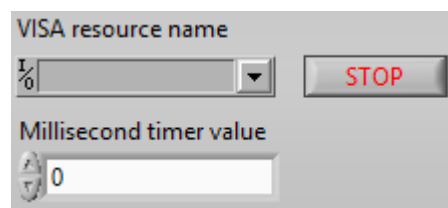


Рисунок 5 — Компоненти панелі керування ВП.

- 2) Елементи керування записом результатів вимірів в файл.



Рисунок 6 — Компоненти панелі керування ВП.

- 3) Елементи керування значеннями рівнів високої та низької температури.

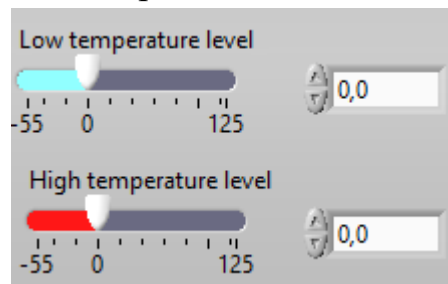


Рисунок 7 — Компоненти панелі керування ВП.

- 4) Індикатори поточного рівня температури.

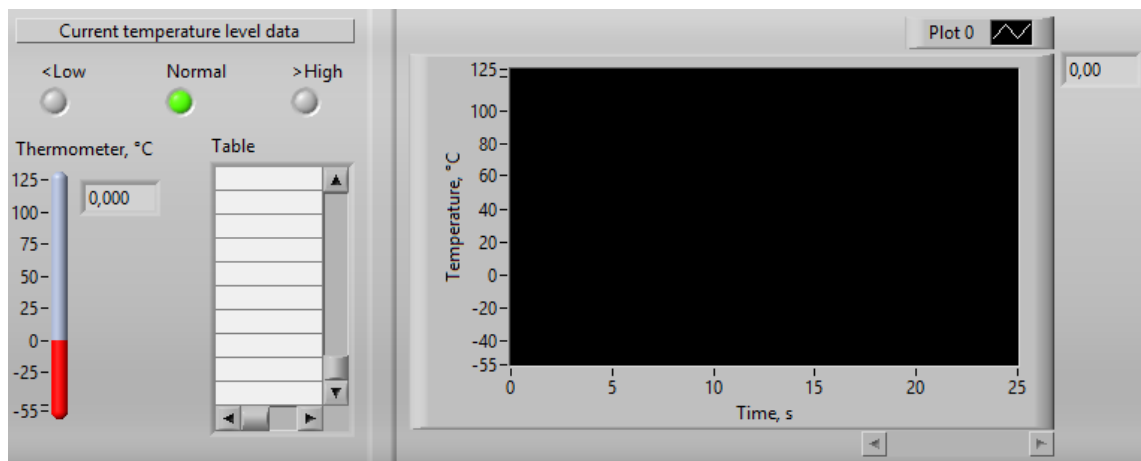


Рисунок 8 — Компоненти панелі керування ВП.

При створенні панелі керування ВП були використані наступні компоненти палітри елементів керування:

- 1) I/O → VISA Resource, для вибору COM-порту, з якого буде відбуватись зчитування даних;
- 2) Boolean → Stop Button, кнопка зупинки роботи ВП;
- 3) Numeric → Numeric Control, для введення значення затримки компонента Wait (ms) з палітри функцій, а також для налаштування умовних значень високої та низької температури;
- 4) Boolean → Push Button, кнопка активації функції запису значень у файл;
- 5) Boolean → Round LED, індикатор активації функції запису значень у файл та індикатори, що символізують про відповідність поточного значення температури заданим користувачем умовним значенням високої, низької та нормальної температури;
- 6) String & Path → File Path Control, для налаштування шляху до файлу, в який будуть зберігатись виміряні значення температури;
- 7) Numeric → Horizontal Pointer Slide, для налаштування умовних значень високої та низької температури;
- 8) Numeric → Thermometer, для відображення поточного значення температури;
- 9) Express → Text Indicators → Table, для відображення виміряних значень температури у вигляді таблиці;
- 10) Graph → Waveform Chart, для відображення зміни значень температури у часі;
- 11) Decorations → Horizontal Smooth Box, для налаштування зовнішнього вигляду панелі керування ВП;
- 12) Decorations → Horizontal Button Box, для налаштування зовнішнього вигляду панелі керування ВП;

Блок-діаграма створеного ВП виглядає наступним чином:

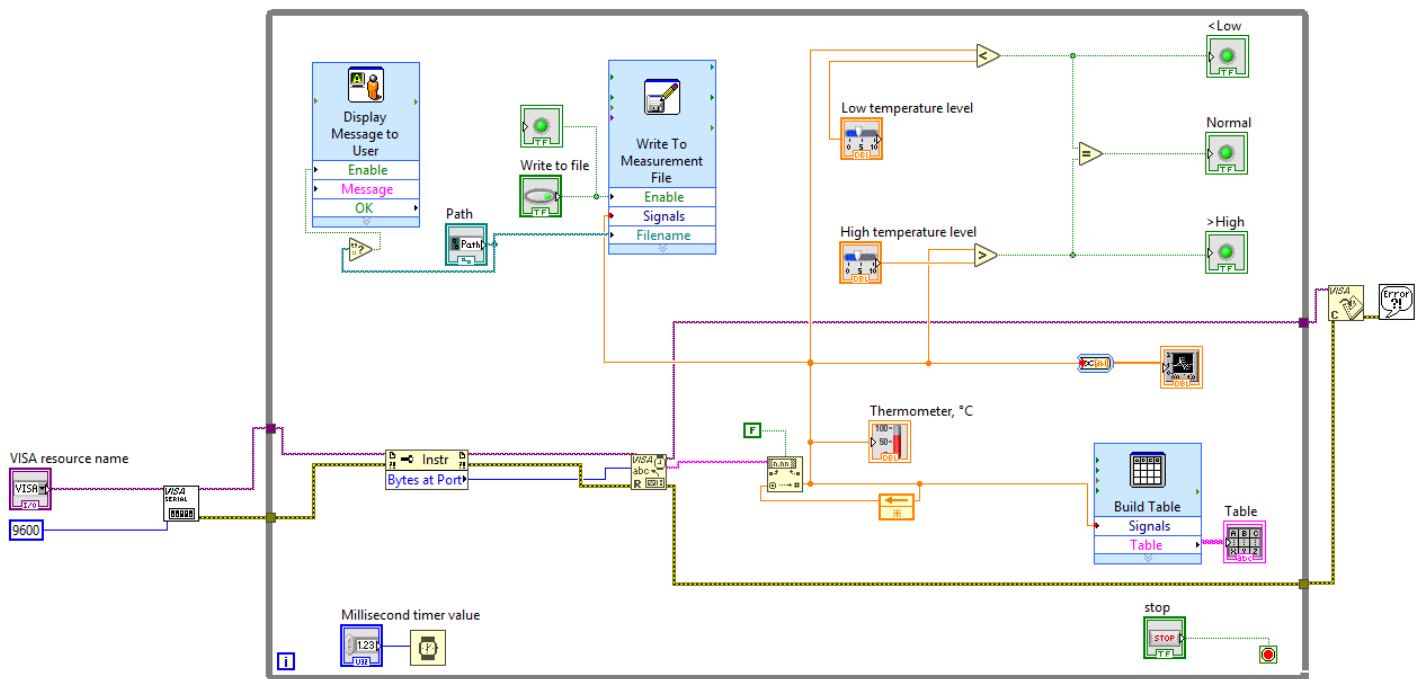


Рисунок 9 — Блок-діаграма створеного віртуального приладу.

Для зв'язку з платою Arduino UNO, до якої підключений датчик температури, використовуються компоненти палітри функцій VISA.

VISA, Virtual Instrument Software Architecture — це широко використовуваний стандартизований інтерфейс введення-виведення в області тестування і вимірювань для управління приладами з персонального комп'ютера. Він підтримує інтерфейси IEEE-488 (GPIB), VXI, RS-232, а також USB для вимірювальних пристроїв. Інтерфейс VISA передбачає спілкування з приладом в формі «запит-відповідь». Комп'ютер відправляє специфічну для конкретного приладу команду-запит (наприклад, вимога виконати виміри певної фізичної величини) і чекає відповіді (наприклад, звіт про стан або результати вимірювань) від приладу. Інтерфейс також підтримує специфічні для шини особливості, наприклад, при роботі з GPIB спілкування може також бути ініційовано приладом за допомогою переривання, а для RS-232 можна конфігурувати бітрейт, кількість бітів в кадрі даних і т.д.

Таким чином, компоненти палітри VISA надають, в даному випадку, можливість зчитування даних, які являють собою результати вимірювань температури датчиком, і приходять на ПК за допомогою послідовного інтерфейсу UART, який і використовується для зв'язку Arduino з комп'ютером.

Розглянемо детальніше компоненти палітри функцій та палітри елементів керування, які були використані для реалізації створеного ВП та їх місцезнаходження в відповідних палітрах.

Перш за все, було використано функцію VISA Configure Serial Port (Instrument I/O → Serial → Configure port) для вибору послідовного порту та налаштування його параметрів, таких як бодрейт, кількість бітів в пакеті.

Для отримання параметрів поточного з'єднання — а саме кількості байтів, доступних в поточний момент часу в послідовному порту, використовується компонент Property Node (Instrument I/O → Serial → Bytes at Port).

Компонент VISA Read (Instrument I/O → VISA → Read) використовується безпосередньо для зчитування вибраної кількості байтів з необхідного пристрою, і збереження їх в буфер зчитаних даних.

Після зчитування даних і збереження в буфері читання, їх потрібно перетворити з типу «string» в відповідні числові значення, що виконується за допомогою компоненту Fract/Exp String To Number (Programming → String → Number/String Conversion → Fract/Exp String To Number). Для врахування ситуацій з маніпуляціями з некоректними значеннями даних, використовується також вхід default даного компоненту, куди заводиться значення з виходу number цього ж компоненту, за допомогою функції Feedback Node (Programming → Structures → Feedback Node). Також, для коректного виведення значень температури за допомогою елементу Waveform Chart з палітри елементів керування, використовується функція Convert from Dynamic Data (Express → Signal Manipulation → From DDT).

Після цього, отримані дані можна вивести у вигляді таблиці, а також у вигляді шкали термометра та графіка-вейвформи, що відображає зміну значення температури в часі.

Крім цього, за допомогою кнопки Write to file (функція Express → Output → Write Meas), користувач має змогу зберегти виміряні значення температури у, наприклад, Excel-файл, вказавши шлях до файлу-призначення. Для індикації активації даної функції ВП використовується відповідний індикатор на панелі керування.

24.10.2019 11:31:06,609	23,62
24.10.2019 11:31:07,359	23,5
24.10.2019 11:31:08,348	23,31
24.10.2019 11:31:09,313	23,19
24.10.2019 11:31:10,313	23,12
24.10.2019 11:31:11,300	23,06
24.10.2019 11:31:12,379	23,06

Рисунок 10 — Вигляд збережених вимірених значень температури у Excel-файлі.

Також реалізована можливість встановлення значень рівнів високої та низької температур, з якими буде порівнюватись поточне значення температури, і відбуватиметься відповідна індикація. Для цього слугують компоненти на панелі керування ВП з палітри елементів керування, а також такі компоненти з палітри функцій, як Greater? (Programming → Comparison → Greater?), Less? (Programming → Comparison → Less?) та Equal? (Programming → Comparison → Equal?) для порівняння поточного значення температури з умовними значеннями високого та низького рівнів температури, заданими користувачем.

Робота описаних вище функцій відбувається в циклі While (Programming → Structures → While Loop). Умовою зупинки виконання даного циклу — While Condition — є натискання кнопки Stop, розміщеної на панелі керування ВП.

Після закінчення їх виконання, відпрацьовує компонент VISA Close (Instrument I/O → Serial → Close), що закриває сесію обміну даними з відповідним пристроєм, а також експрес-ВП Simple Error Handler (Programming → Dialog & User Interface → Simple Error Handler.vi), який виконує функцію обробки помилок, які можуть виникнути в процесі роботи VISA-компонентів.

Варто також відмітити, що зчитування даних з послідовного порту відбувається з інтервалом, який є налаштовуваним, що реалізовано за допомогою функції Wait (ms) (Programming → Timing → Wait (ms)) та відповідного поля для введення на панелі керування ВП. Рекомендоване значення затримки — приблизно 750мс, так як саме такий проміжок часу проходить між послідовними отриманнями температури від датчика, що пов'язано з тим, що йому потрібен деякий час, щоб зробити виміри температури і здійснити необхідні перетворення, при вибраній за замовчуванням роздільній здатності, що рівна 12 біт, і саме з таким інтервалом в 750мс дані надсилаються платою Arduino через послідовний порт.

Також реалізовано виведення відповідного повідомлення про помилку користувачу у випадку введення ним некоректного шляху до файлу, в який будуть зберігатись результати вимірювань. Дана функція реалізована за допомогою функцій Empty String/Path (Programming → Comparison → Empty String/Path?) та Display Message to User (Programming → Dialog & User Interface → Display Msg). Варто, при цьому, відзначити, що подібна ситуація призводить до помилки виконання програми, тому варто вводити лише коректні шляхи до файл-призначення для логів, щоб уникнути помилкових завершень виконання роботи ВП.

Опис реалізації програми роботи з датчиком для плати Arduino UNO

Програма, що виконується на платі Arduino UNO і безпосередньо взаємодіє з датчиком, виконує наступні дії:

- конфігурування пінів плати для роботи з датчиками, які передбачають підключення виходу даних датчика до цифрового піну 2 плати Arduino UNO;
- створення необхідних об'єктів для роботи з сенсором;
- конфігурування роботи послідовного порту — швидкість передачі 9600 бод;
- запит на вимірювання температури датчиком;
- затримка, рівна 750мс, для того, щоб датчик встиг зібрати та підготувати дані для передачі;
- надсилання отриманого від датчика значення температури по послідовному порту.

Лістинг даної програми наведений в Додатку 1.

Висновок

В результаті виконання даних лабораторних робіт, я здобув навички створення віртуальних приладів в середовищі LabVIEW, створивши ВП, який дозволяє працювати з датчиком температури DS18B20. Загалом, LabVIEW виявився чудовим інструментом, що надає широкі можливості для створення віртуальних приладів, які можуть дати змогу дуже зручної роботи з різноманітною периферією.

Мікросхема Dallas DS18B20 є дуже цікавим пристроєм. Датчики температури і термометри, створені на її основі, володіють прийнятними для більшості завдань характеристиками, розвиненим функціоналом, є відносно недорогими. Особливу популярність датчик DS18B20 здобув як вологозахисений пристрій для вимірювання температури рідин.

Лістинг програми DS18B20_temperature_sensor.ino, що виконується на платі Arduino UNO:

```
// DS18B20 temperature sensor project

#include <OneWire.h>

#include <DallasTemperature.h>

#define SENSOR_DATA_PIN 2


OneWire one_wire_connection(SENSOR_DATA_PIN);
DallasTemperature ds18b20_sensor(&one_wire_connection);


void setup(void)
{
    Serial.begin(9600);
    ds18b20_sensor.begin();
    ds18b20_sensor.setWaitForConversion(false);
}


void loop(void)
{
    ds18b20_sensor.requestTemperatures();
    delay(750); // 750ms delay
    // 0 means the request to the first IC on the wire
    Serial.println(ds18b20_sensor.getTempCByIndex(0));
}
```