

Д. П. Проценко¹
Ю. В. Шевчук¹
Р. В. Дворницький¹

КРОСПЛАТФОРМЕННИЙ ІНТЕРФЕЙС LABVIEW ДЛЯ МІКРОПРОЦЕСОРНИХ СИСТЕМ

¹Вінницький національний технічний університет

Проведено розробку та апробацію кросплатформенного інтерфейсу LabVIEW для мікропроцесорних систем, який забезпечує динамічну передачу необхідної кількості даних без впливу на основну програму функціонування мікропроцесорної системи, що дає можливість інтегрування інтерфейсу зв'язку з LabVIEW у вже існуючі мікропроцесорні системи

Ключові слова: інтерфейс, LabVIEW, Arduino, RS232, вимірювання.

Вступ

Наразі персональний комп'ютер широко використовується як засіб обробки та накопичення результатів вимірювань, а також для управління експериментальними установками в реальному часі. З додаванням зовнішнього модуля збору даних, комп'ютер можна перетворити в багатофункціональний лабораторний стенд.

Для програмування таких пристроїв можна використовувати універсальні мови програмування: С, Pascal та інші, — але це досить трудомістке завдання і вимагає високої кваліфікації користувача. Щоб полегшити програмування вимірювальних пристроїв, були створені спеціалізовані системи, засновані на принципах візуального графічного програмування, що містять великий набір готових програмних модулів, для спрощеного програмування багатьох операцій та виконання специфічних задач автоматизації наукових вимірювань та експериментів. Система за замовчуванням сама виконує багато функцій, які зазвичай покладаються на програміста, звільняючи користувача від оперування низькорівневими подробицями роботи програми.

Однією з поширених систем такого роду є LabVIEW (Laboratory Virtual Instruments Engineering Workbench), розроблена фірмою National Instruments. Ця система широко використовується в навчальному процесі в багатьох провідних університетах світу (в області автоматизації експериментальних установок, в практикумах, для моделювання, створення комп'ютерних демонстрацій і т. д.). У фундаментальній науці вона використовується в передових наукових центрах, наприклад CERN, Lawrence Livermore, Oak ridge і багатьох інших, а також у високотехнологічних науково-промислових корпораціях (NASA, Jet propulsion Laboratory).

Для зв'язку середовища LabVIEW із зовнішніми пристроями використовуються системи збору даних National Instruments, наприклад USB-600X. В роботі [1] запропонований підхід щодо використання як системи збору даних та пристрою керування лабораторним устаткуванням апаратної платформи Arduino, що обмінюється з LabVIEW через драйвер послідовного порта NI-VISA. В такому випадку використовується бібліотека комунікаційного інтерфейсу з Arduino LIFA (LabVIEW Interface for Arduino), яка знайшла широке використання для розробки візуальних приладів [2—4]. Інструментальні блоки бібліотеки LIFA показано на рис. 1.

Незважаючи на дешевину та простоту

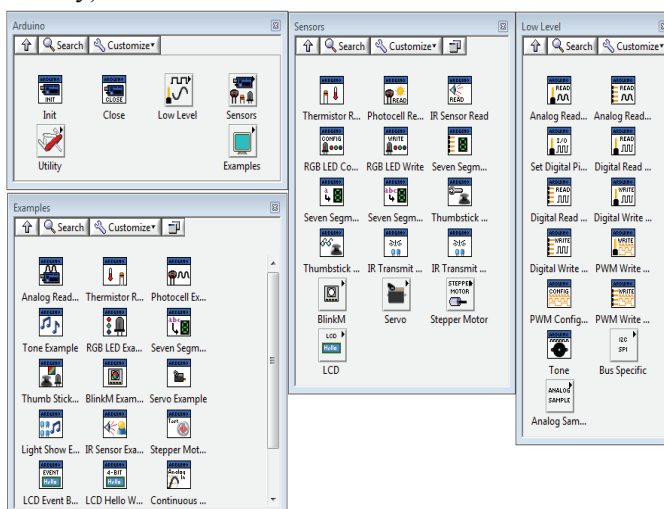


Рис. 1. Інструментальні блоки бібліотеки LIFA

побудови такий підхід не позбавлений низки недоліків:

1. Обмежений спектр підтримки апаратних засобів, інтерфейс LIFA підтримує тільки роботу з мікропроцесорною платформою Arduino (плати типу Uno та Mega);
2. Немає можливості використання апаратних засобів локального мікроконтролера, оскільки він працює в режимі плати вводу—виводу, і, як наслідок, суттєво знижуються динамічні властивості системи, обмежені швидкістю передачі даних комунікаційного інтерфейсу.
3. Інтерфейс LIFA підтримує доступ тільки до обмеженої кількості апаратних регістрів вводу виводу. Значення внутрішніх змінних або проміжних даних недоступні.
4. Відсутня можливість інтеграції інтерфейсу зв'язку LabVIEW у вже існуючі мікропроцесорні системи.

Метою роботи є розробка комунікативного кросплатформенного інтерфейсу LabVIEW для мікропроцесорних систем, який би інтегрувався в основну програму мікроконтролера та забезпечував динамічну передачу необхідної кількості даних без впливу на основну програму функціонування мікропроцесорної системи.

Результати дослідження

Запропоновано алгоритм функціонування мікропроцесорної системи з інтегрованою підпрограмою обміну даними з середовищем LabVIEW, що виконується в одному циклі з основною програмою. Алгоритм представлений на рис. 2 передбачає зчитування команди та числових значень з комунікаційного порта та виконання відповідної функції кожної команди в циклі. Очевидно, що ці функції можна поділити на функції зчитування даних з середовища LabVIEW, та запису в порт значень, отриманих з мікропроцесорної (МП) системи.

Як мікропроцесорний пристрій використано апаратну платформу Arduino. Обмін даними здійснюється через послідовний інтерфейс RS232 модуля UART, який входить в стандартний набір інструментів будь-якої сучасної мікропроцесорної системи. На рис. 3 показано фрагмент лістингу програми зчитування команди та даних в середовищі розробки Arduino IDE. Варто зауважити, що перший елемент вхідних знакових даних — це команда у вигляді одного символу, яка записується в змінну `in_comand`. Наступні значення це дані в форматі string, які записуються в змінну даних `com_data`.

```

//*****
//          Підпрограма зчитування та обробки даних порта
//*****
int readSerial() {
    String inData = "";
    if (Serial.available() > 0) {
        delay(5);
        int h = Serial.available();
        in_comand = (byte)Serial.read();
        for (int i = 1; i < h; i++) {
            inData += (char)Serial.read();
            com_data = inData.toInt();
        }
    }
}

```

Рис. 3. Фрагмент лістингу програми зчитування команди та даних в середовищі розробки Arduino IDE

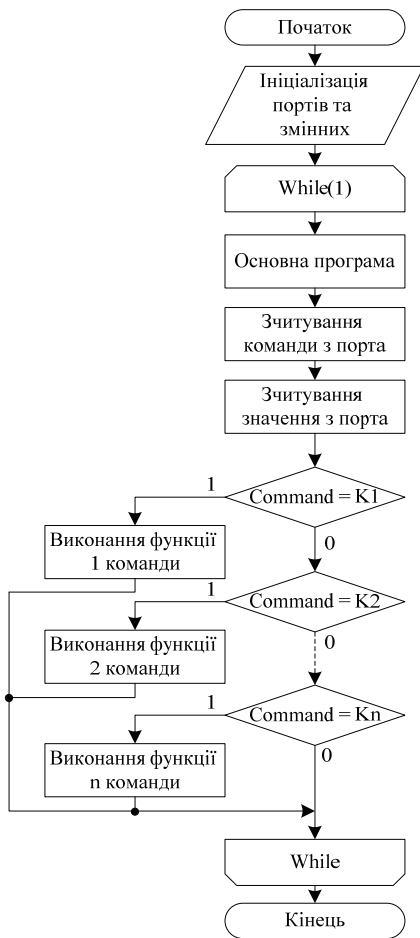


Рис. 2. Алгоритм функціонування мікропроцесорної системи з інтегрованою підпрограмою обміну даними з середовищем LabVIEW

Для тестування обміну проведемо запис двох змінних в регістри керування ШІМ виходами, та зчитування двох аналогових та одного цифрового входів.

На рис. 4 показано фрагмент лістингу програми запису двох змінних в регістри керування 8-розрядними ШІМ виходами, як бачимо, використовуються виводи з номерами 10 та 3, яким відповідають символічні команди К та М. За командою зчитуються дані, що визначають ширину імпульсу на виході, тому рядок даних, отриманих з LabVIEW матиме, наприклад, такий вигляд K123 M25, це означає, що коефіцієнт заповнення сигналу на виводах 10 та 3 становитимуть:

$$D_{10} = \frac{k_{10} \cdot 100 \%}{2^n - 1}, \quad D_3 = \frac{k_3 \cdot 100 \%}{2^n - 1}, \quad (1)$$

де D_{10}, D_3 — значення коефіцієнта заповнення сигналу на виводах 10 та 3, відповідно, %; k_{10}, k_3 — значення керувальних змінних що визначають ширину імпульсу на виводах 10 та 3 ($k_{10} = 123, k_3 = 25$); n — розрядність ШІМ ($n = 8$).

Отже, згідно з формулою (1) коефіцієнт заповнення сигналу на виводах 10 та 3 становитиме 48,2 % та 9,8 %, відповідно.

На рис. 5 показано фрагмент лістингу програми зчитування стану цифрового входу та двох аналогових сигналів на виводах A0 та A1 з пересиланням цих значень по протоколу RS232 за командами R, T, A, відповідно.

З боку середовища LabVIEW використано драйвер NI-VISA (Virtual Instrument Software Architecture) — широко використовуваний стандартизований [5] інтерфейс вводу-виводу в області тестування і вимірювань для управління приладами з персонального комп'ютера. Підтримує інтерфейси IEEE-488 (GPIB), VXI, RS-232, а також USB для вимірювальних пристроїв.

Блок-схема тестової програми в середовищі LabVIEW зображена на рис. 6. Основними компонентами головного циклу While є блоки зчитування та запису COM READ та COM WRITE. Блоки зчитування та запису можуть розташовуватись в будь-якому порядку та з'єднуються послідовно однією шиною. Розглянемо детальніше структуру цих блоків.

На рис. 7 показано структурну схему блока зчитування COM READ в середовищі LabVIEW. Вхідними сигналами блоку є входи послідовної шини VISA Refnum in та error in, write buffer, в який записується знакова команда та константа milliseconds to wait, яка визначає час очікування даних з МП системи. Вихідними сигналами є виходи послідовної шини VISA Refnum out, error out та змінна типу double Numeric, яка містить зчитане

```

//*****
//          Запис змінних зчитаних з порта
//*****
if(in_comand == 'K')
{
    analogWrite(10, com_data);
}
if(in_comand == 'M')
{
    analogWrite(3, com_data);
}
    
```

Рис. 4 Фрагмент лістингу програми запису двох змінних в регістри керування 8-розрядним ШІМ виходами

```

//*****
//          Запис змінних в порт зчитаних з цифрових та аналогових датчиків
//*****
if(in_comand == 'R')
{
    int buttonState = digitalRead(pushButton);
    Serial.println(buttonState);
    in_comand = ' ';
}
if(in_comand == 'T')
{
    int sensorValue = analogRead(A0);
    Serial.println(sensorValue);
    in_comand = ' ';
}
if(in_comand == 'A')
{
    int sensorValue1 = analogRead(A1);
    Serial.println(sensorValue1);
    in_comand = ' ';
}
    
```

Рис. 5. Фрагмент лістингу програми зчитування стану цифрового входу та двох аналогових сигналів на виводах A0 та A1

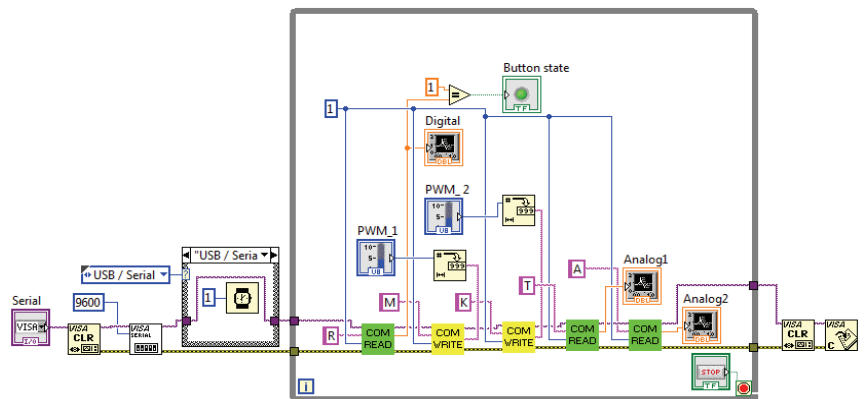


Рис. 6. Блок-схема тестової програми в середовищі LabVIEW

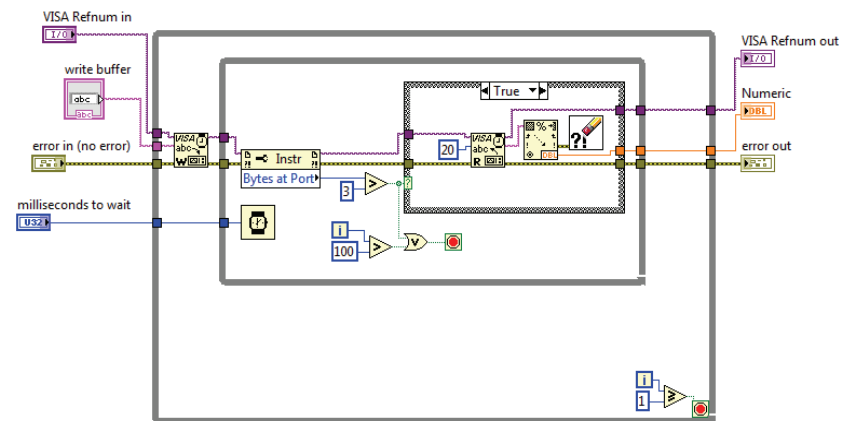


Рис. 7. Структурна схема блока зчитування COM READ в середовищі LabVIEW

значення. Як видно зі структурної схеми, спочатку передається команда, а потім здійснюється очікування прийому чотирьох байт даних, що відповідають чотирьом знаковим змінним, далі ці значення зчитуються та перетворюються в числовий формат на виводі Numeric який можна використовувати в середовищі LabVIEW.

На рис. 8 показано структурну схему блока запису даних в МП систему COM WRITE в середовищі LabVIEW. Вхідними сигналами блока є входи послідовної шини VISA Refnum in та error in, command, який відтворює знакову команду та char_data, який визначає дані, що записуються в порт та константа millisecond to wait, яка визначає час очікування запису. Вихідними сигналами є виходи послідовної шини VISA Refnum out та error out.

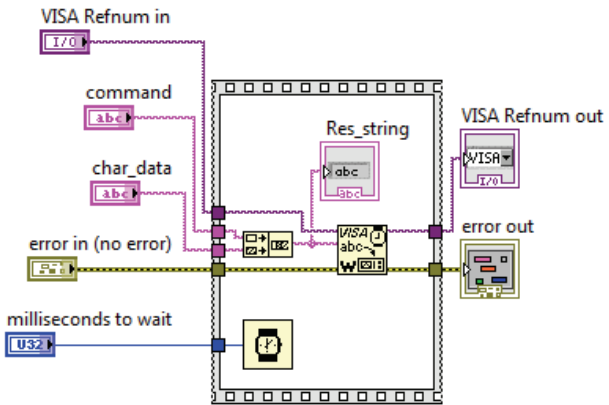


Рис. 8. Структурна схема блока запису даних COM WRITE в середовищі LabVIEW

Лицьова панель розробленої системи зображена на рис. 9. На ній відображаються графіки аналогових сигналів Analog1 та Analog2, що відтворюють значення напруги на виводах A0 та A1, до яких підключено потенціометр та генератор синусоїдального сигналу, відповідно. Графік Digital відображає стан кнопки на виводі мікроконтролера. В меню Serial вибирається активний послідовний порт для обміну між МП системою та середовищем LabVIEW. Ползунки PWM_1, та PWM_2 задають значення ширини імпульсів сигналів ШІМ на виводах 3 та 10, відповідно. Як видно з графіків, дані зчитуються та відображаються адекватно.

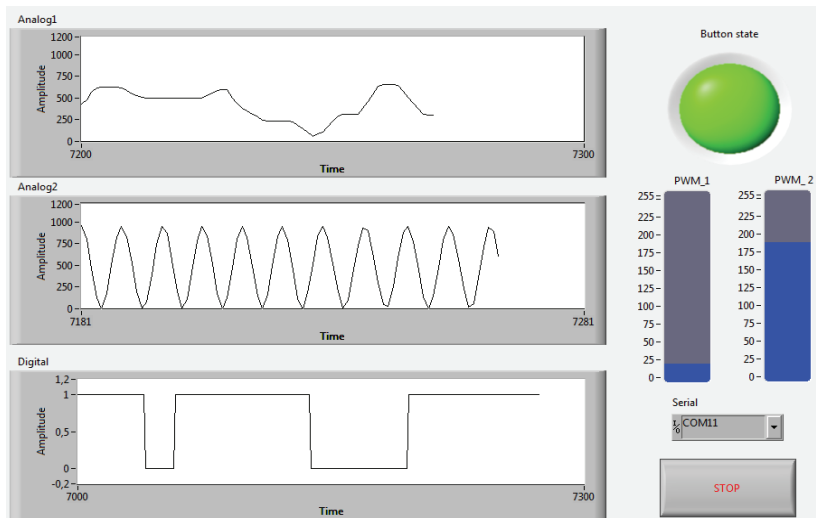


Рис. 9. Лицьова панель розробленої системи обміну даними в середовищі LabVIEW

Вигляд вихідних сигналів зафіксовано, використовуючи двоканальний осцилограф Instrustar ISDS205B, осцилограму сигналів PWM_1 та PWM_2, яким відповідають перший та другий вимірювальний канал, показано на рис. 10. Як видно з графіків, коефіцієнт заповнення для сигналів PWM_1 та PWM_2 становить 7% та 73%, відповідно, що узгоджується з заданими значеннями на лицьовій панелі.

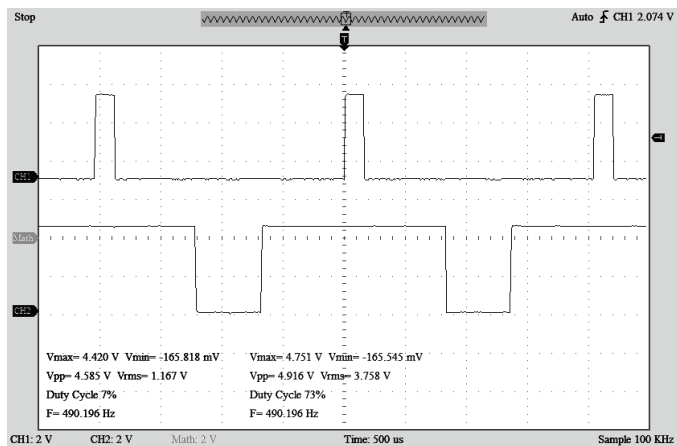


Рис. 10. Осцилограма вихідних сигналів ШІМ виводів

Висновки

Отже, розроблений інтерфейс зв'язку середовища LabVIEW з мікропроцесорними системами дозволяє використовувати його для широкого спектру апаратних засобів завдяки використанню стандартного послідовного зв'язку RS232. Запропоновано алгоритмічно-програмне забезпечення для інтеграції інтерфейсу зв'язку середовища LabVIEW з МП системами, практична реалізація якого на базі апаратної платформи Arduino підтвердила його працездатність. Розроблений крос-платформенний інтерфейс забезпечує динамічну передачу необхідної кількості даних без впливу на основну програму функціонування мікропроцесорної системи, що дає можливість додавання інтерфейсу зв'язку LabVIEW у вже існуючі мікропроцесорні системи.

СПИСОК ВИКОРИСТАНОЇ ЛІТЕРАТУРИ

1. Naveenkumar R. Low Cost Data Acquisition and Control using Arduino Prototyping Platform and LabVIEW / R. Naveenkumar, Prasad Krishna // International Journal of Science and Research (IJSR). — 2013. — № 2. — С. 366—369.
2. Sathiyabama G. Controller Design for Quadcopter Using Labview With Image Processing Techniques / G. Sathiyabama, R. Praveenkumar, A. V. K Viswanath, R. Visunpriyan // International Journal for Research and Development in Engineering (IJRDE). — 2014. — No 1. — P. 452—455.
3. The Wearable Electronic Rescue System for Home Alone Elderly- Labview & Arduino Evaluation / [J. Kumar Roy, B. Deb, D. Chakraborty, S. Mahanta, N. Banik] // IOSR Journal of Electronics and Communication Engineering (IOSR-JECE). — 2013. — № 8. — P. 50 — 55.
4. Kantharia P. Design of Sensor Fault detection and Remote monitoring system for temperature measurement / P. Kantharia, T. Patel, M. Thakker // International Journal of Current Engineering and Technology. — 2014. — № 2. — P. 504—508.
5. Загидуллин Р. Ш. LabView в исследованиях и разработках / Р. Ш. Загидуллин — М. : Горячая линия–Телеком, 2005. — 352 с.

Рекомендована кафедрою метрології та промислової автоматики ВНТУ

Стаття надійшла до редакції 12.09.2014

Проценко Дмитро Петрович — канд. техн. наук, доцент кафедри електромеханічних систем автоматизації в промисловості і на транспорті, e-mail: procenkod@rambler.ru;

Шевчук Юрій Володимирович — канд. техн. наук, старший викладач кафедри електромеханічних систем автоматизації в промисловості і на транспорті, e-mail: yuriy.shevchuck@gmail.com;

Дворніцький Руслан Володимирович — студент Інституту електроенергетики та електромеханіки.

Вінницький національний технічний університет, Вінниця

D. P. Protsenko¹
Yu. V. Shevchuk¹
R. V. Dvornytskyi¹

Cross-platform interface LABVIEW for microprocessor systems

¹Vinnitsia National Technical University

Development and approbation of cross-platform interface of LabVIEW for microprocessor systems are conducted, that provides the dynamic transmission of necessary amount of data without influence on the main program of functioning of microprocessor system that gives an opportunity of integration of interface of connection with LabVIEW in already existent microprocessor system.

Keywords: interface, LabVIEW, Arduino, RS232, measuring operations.

Protsenko Dmytro P. — Cand. Sc. (Eng.), Assistant Professor of the Chair of Electromechanical Systems of Automation in Industry and on Transport, e-mail: procenkod@rambler.ru;

Shevchuk Yuriy V. — Cand. Sc. (Eng.), Senior Lecturer of the Chair of Electromechanical Systems of Automation in Industry and on Transport, e-mail: yuriy.shevchuck@gmail.com;

Dvornitskyi Ruslan V. — Student of the Institute of Electricity and Electromechanics

Д. П. Проценко¹
Ю. В. Шевчук¹
Р. В. Дворницкий¹

Кроссплатформенный интерфейс LABVIEW для микропроцессорных систем

¹Винницкий национальный технический университет

Проведена разработка и апробация кроссплатформенного интерфейса LabVIEW для микропроцессорных систем, обеспечивающего динамическую передачу необходимого количества данных без влияния на основную программу функционирования микропроцессорной системы, что дает возможность интегрирования интерфейса связи с LabVIEW в уже существующие микропроцессорные системы.

Ключевые слова: интерфейс, LabVIEW, Arduino, RS232, измерения.

Проценко Дмитрий Петрович — канд. техн. наук, доцент кафедры электромеханических систем автоматизации в промышленности и на транспорте, e-mail: procenkod@gambler.ru;

Шевчук Юрий Владимирович — канд. техн. наук, ст. преп. кафедры электромеханических систем автоматизации в промышленности и на транспорте, e-mail: yuriy.shevchuck@gmail.com;

Дворницкий Руслан Владимирович — студент Института электроэнергетики и электромеханики