

УДК 004.31 : 004.424

Модернизация канала компьютерного управления учебной моделью цифрового процессора

Ю.Л. Грабовская, С.Н. Рева

Харьковский национальный университет имени В.Н. Каразина, Украина

На примере компьютеризации учебной модели цифрового процессора описаны проблемы организации обмена в системах контроля и управления, построенных с использованием современных версий операционных систем семейства Windows. Изучена проблема взаимодействия новых операционных систем с аппаратными ресурсами компьютера. Предложен вариант построения канала обмена, принцип формирования протокола взаимодействия и структура управляющего программного обеспечения, которые могут обеспечить кроссплатформенность, аппаратную и программную совместимость с современными персональными компьютерами. Описаны примеры программной реализации и результаты их тестирования. Даны рекомендации о возможности применения предложенных методов для построения промышленных и научных установок с компьютерным управлением.

Ключевые слова: цифровой процессор, учебная модель, компьютерное управление, кроссплатформенность, протокол обмена.

Робота присвячена модернізації каналу зв'язку між комп'ютером та створеною раніше навчальною моделлю цифрового процесора. На прикладі комп'ютеризації навчальної моделі описані проблеми управління та програмного контролю обладнання, які можуть виникати при заміні комп'ютерів та операційних систем сучасними версіями. Акцентовано увагу на зміні зовнішніх портів комп'ютерів та суттєвому обмеженні доступу до їх апаратних регістрів. На основі проведеного аналізу запропоновано методи вирішення проблеми, серед яких слід відмітити заміну апаратних засобів та протоколів зв'язку між комп'ютером та обладнанням, впровадження мікропроцесорних інтерфейсних модулів для узгодження сигналів управління, а також розробку нового програмного забезпечення, побудованого на засадах кроссплатформності. Наведено конкретні приклади створення послідовних каналів зв'язку з використанням мікроконтролерних інтерфейсів, які призначені для зміни формату сигналів управління, описано структурну схему інтерфейсного модуля. Запропоновано структуру програмного забезпечення, що здатне працювати під різними операційними системами. Пакет програмних засобів створено з використанням мов програмування Java та JavaScript, для взаємодії з послідовним портом використано бібліотеку Java Simple Serial Connector. В роботі звернено увагу на особливості використання цієї бібліотеки, наведено уривок програмного коду. Під час досліджень проведено тестування створеної програмної структури з використанням апаратного емулятора для здійснення обміну даними в рамках розробленого протоколу, наведено результати тестування. Зроблено висновок про можливість використання запропонованих рішень для побудови промислових та наукових установок з комп'ютерним управлінням.

Ключеві слова: цифровой процессор, учебная модель, компьютерное управление, кроссплатформенность, протокол обмена.

The article describes the modernization of a communication channel between a computer and an educational model of the digital processor created a few years ago. An example of the computerization of a learning model describes the management and software control problems of an equipment that may occur when replacing computers and operating systems with modern versions. The focus is on changing external ports of computers and severely restricting access to their hardware registers. On the basis of the conducted analysis, the methods of solving the problem are proposed. They are the replacement of hardware and communication protocols between the computer and equipment, the introduction of microprocessor interface modules for the coordination of control signals, as well as the development of new cross-platform software. The examples of serial communication channels using microcontroller interfaces, which are intended to change the format of control signals, are given. The block diagram of the interface module is presented. The structure of software that can work with different operating systems has been offered. By using Java and JavaScript programming languages, the software package has been created. For interaction with the serial port the Java Simple Serial Connector library has been used. The paper focuses on the peculiarities of using this library. A piece of software code is given. A hardware emulator for data exchange based on the developed protocol has been used for testing. The results of testing are presented as well. The proposed solutions can be used for creating industrial and scientific devices with computer control.

Key words: digital processor, educational model, computer control, cross-platform, protocol of exchanging.

1 Введение

С целью повышения уровня подготовки программистов в области электроники, ознакомления с принципами построения и функционирования процессоров на аппаратном уровне, а также для формирования представления о классической архитектуре микропроцессоров и языках программирования низкого уровня кафедрой электроники и управляющих систем Харьковского национального университета имени В.Н. Каразина при участии студентов старших курсов была разработана учебная модель цифрового процессора. Она построена на микросхемах малой степени интеграции, позволяет продемонстрировать процесс выполнения программы на уровне

отдельных логических элементов и обрабатываемых ими сигналов [1].

Чуть позже на кафедре был создан интерфейсный модуль, который позволил подключать макет процессора к персональному компьютеру для мониторинга его состояния и управления работой в реальном времени. Благодаря разработке этого интерфейса и сопутствующего программного обеспечения появилась возможность:

- загружать в оперативную память более объемные программы, полученные путем трансляции с языка ассемблер;
- отображать на экране монитора текущее состояние модели процессора, содержимое регистров, памяти, состояние шин;
- показывать ход выполнения отдельных команд;
- управлять работой процессора как на уровне выполнения отдельных машинных циклов, так и на уровне выполнения подпрограмм.

Все эти возможности существенно улучшают наглядность макета и позволяют более эффективно использовать его в учебном процессе.

Взаимодействие модели процессора с компьютером через созданный интерфейс осуществляется через параллельный восьмиразрядный LPT-порт [2]. Передача данных из компьютера в макет выполняется по параллельной восьмиразрядной шине, а чтение состояния функциональных узлов модели производится полубайтами с помощью системы мультиплексоров и с использованием служебных регистров LPT-порта. При организации обмена, программа, непосредственно, управляет работой как регистров данных, так и входных и выходных служебных регистров в составе порта. Параллельная передача данных обеспечивает высокое быстродействие системы в целом.

2 Проблематика

На протяжении последних лет модель процессора успешно применялась в учебной практике факультета компьютерных наук. Работа с моделью осуществлялась при использовании операционных систем, предоставляющих возможность доступа к аппаратным ресурсам компьютера (из серии операционных систем Windows работа была возможна с версиями Windows 98, Windows Vista, Windows XP).

Но на сегодняшний день все перечисленные выше операционные системы не поддерживаются компанией Майкрософт и постепенно вытесняются из использования. На смену им приходят более новые версии, такие как Windows 7, Windows 8, Windows 10. Во всех этих системах с целью обеспечения безопасности и их устойчивости к вредоносным программам максимально ограничен доступ к аппаратным ресурсам компьютера. Это делает невозможным использование ранее созданной программы по управлению моделью процессора, так как в существующей версии передача данных осуществляется через служебные регистры внешнего порта компьютера. Вероятней всего, последующие операционные системы будут еще больше ограничивать возможности прямого управления портами без использования специально разрабатываемых драйверов, предусматривающих соответствующие правила доступа и ограничения полномочий.

Современная тенденция развития персональных компьютеров определена в первую очередь коммерческими интересами производителей и ориентирована на удовлетворение потребностей наиболее широкого контингента пользователей. Это, в первую очередь, — реализация коммуникативных, мультимедийных и игровых функций. По этой причине в современных компьютерах «исчезли» параллельные порты ввода-вывода. Передача данных в основном осуществляется через USB-порты, а также через порты, в основу работы которых положен радиочастотный принцип передачи информации (технологии Bluetooth и Wi-Fi). В результате созданное ранее оборудование невозможно подключить к современным ноутбукам и стационарным персональным компьютерам.

В связи с этим возникла необходимость для дальнейшего успешного использования модели процессора разработать новый модуль аппаратного интерфейса и новое программное обеспечение, которое позволило бы управлять аппаратурой при работе с современными компьютерами в операционных системах Windows 7 ... Windows 10.

3 Методы решения

С целью поиска приемлемого решения был проведен системный анализ задачи, в результате

которого определены методы построения аппаратно-программного комплекса, позволяющие успешно использовать современные компьютеры и операционные системы для управления внешним оборудованием.

Во-первых, для организации связи с аппаратурой необходимо ориентироваться на применение наиболее используемых на сегодняшний день в компьютерах и защищенных от воздействия электромагнитных помех USB-портов. Но у них есть недостатки:

короткое расстояние передачи данных;

передача через USB-порт осуществляется в последовательном коде, следовательно, для управления макетом необходимо устройство декодирования, обеспечивающее запись данных в параллельные регистры и осуществляющее обратное преобразование данных.

Для устранения первого недостатка в качестве наиболее удобной технологии выбрано использование виртуального COM-порта [3], который организуется при подключении адаптера COM-USB. Это дает нам возможность удалить устройство на расстояние до нескольких десятков, а то и сотен метров, хотя это приведет к определенному снижению скорости обмена информацией. А в будущем, при соответствующей доработке аппаратного интерфейса, можно перейти на протокол беспроводной связи и использовать один из радиоканалов, например, Bluetooth или Wi-Fi.

В качестве устройства для преобразования данных из последовательного формата в параллельный и обратно можно применить микроконтроллер или же разработать микропрограммный автомат на основе программируемой логической матрицы.

Во-вторых, необходимо разработать протокол взаимодействия компьютера с управляемым устройством и систему команд, которые бы не требовали непосредственного доступа к аппаратным регистрам порта, а основывались на передаче данных с использованием унифицированных протоколов обмена, поддерживаемых операционными системами и встроенными системными драйверами.

В-третьих, при выборе технологии разработки программного обеспечения следует ориентироваться на внедрение принципов кроссплатформенности, чтобы обеспечить работоспособность управляющей программы под различными операционными системами.

4. Структура интерфейсного модуля

Принцип построения двунаправленного канала передачи данных, который связывает компьютер с моделью цифрового процессора, показан на рис. 1.

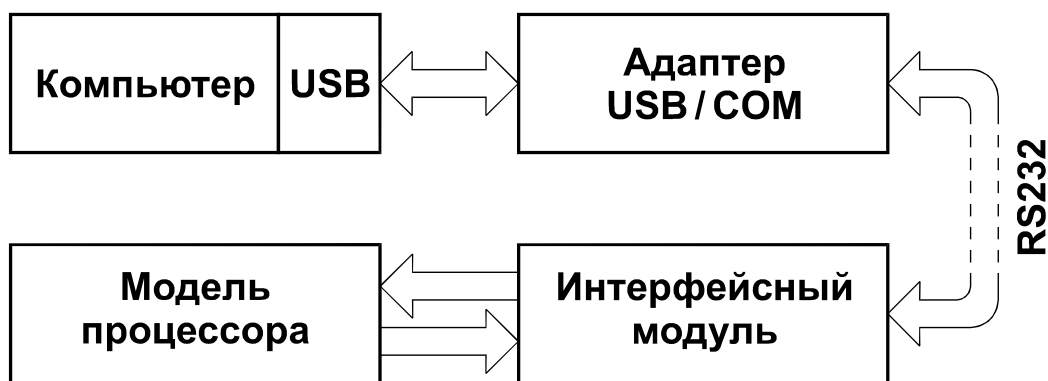


Рис.1. Структура канала передачи данных

Компьютер выполняет обмен информацией с внешним устройством посредством одного из встроенных USB-портов, к которому подсоединен адаптер USB/COM. Последовательная линия передачи данных, связывающая адаптер с интерфейсным модулем, выполнена в стандарте RS232. Сам интерфейсный модуль непосредственно присоединяется к модели процессора через предусмотренный конструкцией модели разъем.

Взаимодействие модуля с моделью процессора осуществляется параллельными шинами и отдельными дискретными сигналами. Для управления и контроля состояния модели необходимо обеспечить формирование пяти управляющих сигналов, чтение четырех однобайтных двоичных

слов, а также организовать формирование двунаправленных восьмиразрядных шин для чтения/записи адреса и данных при взаимодействии с оперативной памятью. Обобщенная структурная схема интерфейсного модуля показана на рис. 2.

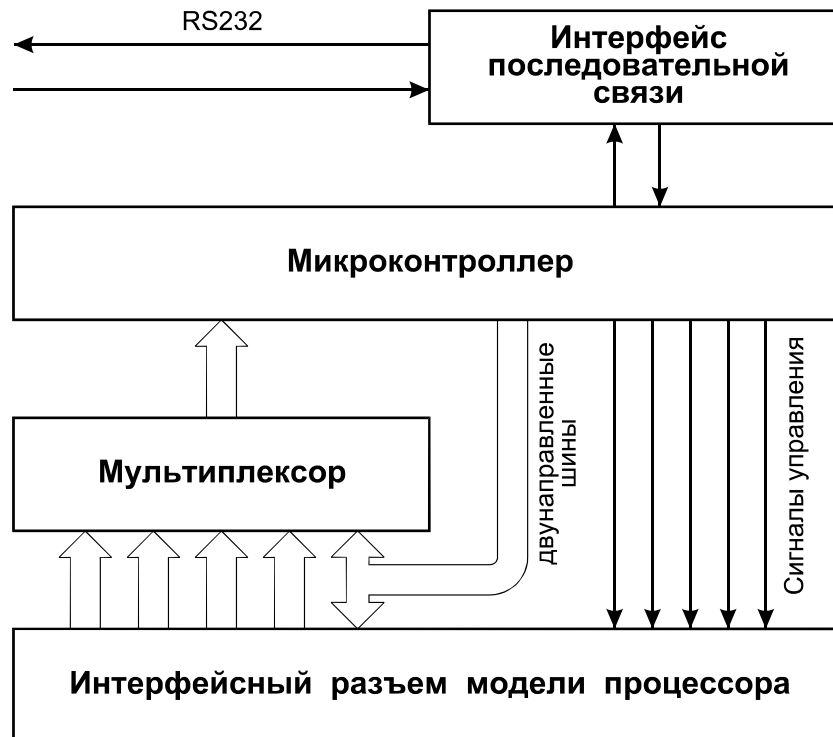


Рис. 2. Структурная схема интерфейсного модуля

Интерфейс позволяет через мультиплексор считывать состояние регистров А и В, состояние регистра команд, а также состояние отдельных сигналов управления, которые вырабатываются генератором сигналов машинного цикла. Мультиплексор также обеспечивает доступ по чтению шины адреса и шины данных.

При работе с модулем памяти контроль шины адреса и шины данных осуществляется непосредственно с портов микроконтроллера. Отдельными дискретными сигналами обеспечено формирование сигналов чтения и записи, а также сигнала Reset и импульсов тактовой частоты. Часть сигналов микроконтроллера используется для управления мультиплексором.

5. Протокол взаимодействия компьютера с моделью цифрового процессора

Для передачи информации с компьютера на модель процессора используется текстовый формат команд. Это исключает появление в передаваемых пакетах специальных символов, которые могут вызвать непредвиденную работу драйверов и системных служб Windows при попытке передачи бинарных значений. Примером такого взаимодействия является передача данных на принтер в виде ASCII кодов при использовании встроенных шрифтов (символьно-ориентированный или, так называемый, телетайпный протокол обмена).

Таким образом, при переходе к передаче текстовых файлов исчезает необходимость в доступе к управляющим регистрам, а передача данных может осуществляться по последовательному каналу с использованием только двух сигнальных проводов TXD и RXD — с использованием Z-модемного соединения. Прием и передачу информации осуществляет микроконтроллер интерфейсного модуля, он же выполняет преобразование команд в параллельный код для непосредственного взаимодействия с регистрами модели процессора.

В соответствии с принятой тенденцией решения задачи был разработан подробный протокол взаимодействия между компьютером и микропроцессорным устройством, на базе которого и спроектирован интерфейсный модуль.

6. Структура программы

Для обеспечения кроссплатформенности программного пакета при его построении были использованы технологии web-программирования, а в качестве основных языков программирования выбраны Java и JavaScript, — языки, которые поддерживаются всеми распространенными браузерами и вне зависимости от типа операционной системы будут работать на любом компьютере. Но задача является не стандартной, так как данные языки ориентированы в первую очередь для решения задач передачи и обработки информации в рамках интернет-технологий, а использование их для управления аппаратными ресурсами является весьма редким случаем.

Учитывая специфику данных языков программирования, была разработана следующая структурная схема программного пакета (рис. 3).

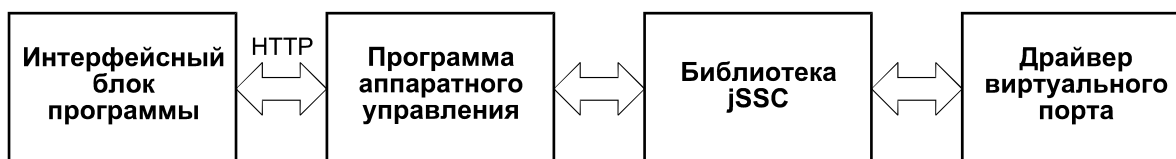


Рис. 3. Схема программного пакета

Интерфейсная часть — программа, написанная на JavaScript, и предназначена для взаимодействия пользователя с системой. Это интерфейс, с помощью которого пользователь имеет возможность отправлять команды в модель цифрового процессора, а также наблюдать результаты визуализации полученных ответов.

Программа аппаратного управления реализована на Java. Она взаимодействует с интерфейсным блоком с использованием протокола HTTP — протокола прикладного уровня передачи данных. Этот программный блок выполняет задачу формирования команд для модели цифрового процессора, а также задачу синтаксического анализа полученных от модели ответов с последующей передачей результатов анализа в интерфейсный блок для визуализации результатов выполнения команды.

Для организации обмена данными между программой аппаратного управления и коммуникационным портом была выбрана библиотека jSSC — Java-библиотека для простой работы с COM-портом (Java Simple Serial Connector) [4]. Своё официальное публичное начало она берёт в 2010 году. Именно тогда было решено поделиться ей с разработчиками на Java (распространяется под лицензией LGPL). К написанию библиотеки привёл факт отсутствия адекватных инструментов для работы с последовательными портами. На то время уже существовали `java.comm` и `gxtx`, но в этих библиотеках есть ряд своих недостатков. Библиотека `java.comm 3.0` не предназначена для работы в операционных системах семейства Windows, кроме того, сложно найти её качественную лицензионную версию. Библиотека `gxtx` не подошла для решения задачи из-за своей нестабильности.

В самой библиотеке jSSC можно выделить несколько основных частей:

- `SerialNativeInterface` — класс, который предоставляет доступ ко всем «нативным» методам библиотеки jSSC;
- `SerialPort` — класс, с помощью которого будет непосредственно производиться работа с выбранным портом;
- `SerialPortEventListener` — интерфейс, который необходимо реализовать, для того чтобы получать данные с порта.

Далее приведен фрагмент кода для команды очистки программной памяти модели процессора. С помощью данной подпрограммы выполняется открытие и настройка порта, передача команды и ожидание приема ответа с его последующей обработкой.

```
public class Test {
    private static SerialPort serialPort;

    public static void main(String[] args) {
        //Передаём в конструктор имя порта
        serialPort = new SerialPort("COM1");
        try {
            //Открываем порт
            serialPort.openPort();

            //Выставляем параметры
            serialPort.setParams(
                SerialPort.BAUDRATE_9600,
                SerialPort.DATABITS_8,    SerialPort.STOPBITS_1,
                SerialPort.PARITY_NONE);

            //Включаем аппаратное управление потоком
            serialPort.setFlowControlMode(
                SerialPort.FLOWCONTROL_RTSCCTS_IN |
                SerialPort.FLOWCONTROL_RTSCCTS_OUT);

            //Устанавливаем слушатель и маску
            serialPort.addEventListener(
                new PortReader(), SerialPort.MASK_RXCHAR);

            //Отправляем запрос устройству
            serialPort.writeString("MCLR;");
        } catch (SerialPortException ex) {
            System.out.println(ex);
        }
    }

    private static class PortReader    implements
        SerialPortEventListener {
        public void serialEvent(SerialPortEvent event) {
            if (event.isRXCHAR() && event.getEventValue() > 0) {
                try {
                    //Получаем ответ от устройства, обрабатываем данные и т.д.
                    String data = serialPort.readString(
                        event.getEventValue());
                } catch (SerialPortException ex) {
                    System.out.println(ex);
                }
            }
        }
    }
}
```

7. Результаты тестирования

Созданный программный пакет прошел несколько этапов тестирования с использованием аппаратного эмулятора, который вместо модели процессора выполнял функции обмена в соответствии с разработанным протоколом и системой команд. Эмулятор реализован на микроконтроллере ADuC831 и позволяет поддерживать как режим штатного выполнения протокола, так и режим моделирования ошибок в канале передачи данных.

Многочасовое тестирование программного комплекса на различных компьютерах с операционными системами Windows XP, Windows 7 и Windows 10 подтвердило кроссплатформенные качества созданного программного обеспечения, а также показало

высокую устойчивость программного пакета к возникновению нештатных ситуаций и нарушениям обмена.

8. Заключение

Разработанный метод организации обмена между управляющим компьютером и ведомым аппаратным оборудованием позволил решить проблему взаимодействия модели цифрового процессора с современными компьютерами широкого применения и новыми операционными системами. По мнению разработчиков метод может успешно применяться для построения промышленных компьютеризированных систем автоматического контроля и управления, которые не предъявляют высоких требований к скорости обмена данными.

Предложенная схема обмена предполагает возможность использования беспроводных каналов передачи информации для управления оборудованием в случае замены аппаратного драйвера последовательной связи и при соответствующей адаптации программного обеспечения.

Подтвержденная тестированием мультиплатформенность созданного программного пакета дает основание предполагать сохранение работоспособности аппаратно-программного комплекса в случае перехода на новые (создаваемые) версии операционных систем.

ЛИТЕРАТУРА

1. Ю. А. Журавель Модель цифрового процессора [Текст] / Ю. А. Журавель, С. Н. Рева // Вісник Харківського національного університету Серія «Математичне моделювання. Інформаційні технології. Автоматизовані системи управління» №987, 2011, с. 5-18.
2. А. А. Белуха Некоторые нюансы разъемов принтеров типа „Centronics» [Текст] / А. А. Белуха РадиоАматор, № 10 2002, с. 22.
3. Последовательный интерфейс RS-232: [Электронный источник] / ООО Рынок микроэлектроники 1998-2016 – URL: <http://www.gaw.ru/html.cgi/txt/interface/rs232/> (Дата обращения 20.09.2018).
4. Работаем с COM-портом из Java при помощи jSSC [Электронный источник] / ТМ 2006-2018 – URL <https://habr.com/post/133766/> (Дата обращения 05.05.2018).

REFERENCIES

1. Y. A. Zhuravel. «The model of the digital processor» *Bulletin of V. Karazin Kharkiv National University*, vol. 987, pp. 5-18, 2011
2. A. A. Belucha. «Some nuances of connectors of printers of the «Centronics» type» *RadioAmator*, vol. 10, p. 22, 2002
3. «Serial interface RS-232.» Internet: <http://www.gaw.ru/html.cgi/txt/interface/rs232/>, 2016 [Sep. 20, 2018]
4. S. Sokolov «We work with COM-port from Java with jSSC.» Internet: <https://habr.com/post/133766/>, Des. 1, 2011 [Apr. 05, 2018].

Грабовская Юлия Леонидовна – студентка факультета компьютерных наук; Харьковский национальный университет имени В.Н. Каразина, г. Харьков-22, площадь Свободы, 4, 61022; e-mail: julia070796@gmail.com; ORCID: 0000-0001-7117-0806.

Рева Сергей Николаевич – кандидат технических наук, доцент кафедры ЭУС; Харьковский национальный университет имени В.Н. Каразина, г. Харьков-22, площадь Свободы, 4, 61022; e-mail: ieclab@karazin.ua; ORCID: 0000-0002-2615-9226.