

# РАЗРАБОТКА ПРОГРАММНОГО ИНТЕРФЕЙСА ДЛЯ ИЗМЕРИТЕЛЬНОГО КОМПЛЕКСА КОНТРОЛЯ ПАРАМЕТРОВ РЕЛЕ КИПРО1

**И. Ш. Невлюдов**

Доктор технических наук, профессор, заведующий кафедрой  
Кафедра ТАПР\*  
Контактный тел.: (057) 397-98-86

**Б. А. Шостак**

Кандидат технических наук, доцент  
Кафедра ТАПР\*  
Контактный тел.: (057) 364-45-53

**А. О. Беликов**

Кафедра ПЕЕА\*  
Контактный тел.: (057) 372-69-92  
\*Харьковский национальный университет радиозлектроники  
пр. Ленина, 14, г. Харьков, Украина, 61166

*У статті приведені результати досліджень впливу типу ОС при формуванні програмного забезпечення вимірювального комплексу для контролю параметрів реле. Виконано порівняння продуктивності комплексу при тестуванні різних типів реле у різноманітних ОС*

*Ключові слова: АЦП, LPT-порт, операційна система*

*В статье приведены результаты исследований влияния типа ОС при формировании программного обеспечения измерительного комплекса для контроля параметров реле. Выполнено сравнение производительности комплекса при тестировании разных типов реле в различных ОС*

*Ключевые слова: АЦП, LPT-порт, операционная система*

*In article there are findings of investigations to influence of the type OS by forming software of the measuring complex for parameters of relay monitoring. Comparison of complex productivity by testing various types of relay in different OS is performed.*

*Key words: ADC, LPT-port, operating system*

## 1. Введение

Для контроля электрических и временных параметров электромагнитных реле постоянного тока был разработан измерительный комплекс КИПРО1. Он обеспечивает измерение следующих параметров реле: ток срабатывания, ток отпускания, сопротивление обмотки, время срабатывания, время отпускания, время дребезга контакта при срабатывании, время дребезга контакта при отпуске, сопротивление контактов и обмотки и т.д.

Аппаратная часть измерительного комплекса состоит из следующих устройств: низковольтный блок, высоковольтный блок, блок высокоскоростного АЦП, коммутационная панель, предназначенная для подключения проверяемого реле к измерительному комплексу.

## 2. Постановка задачи

Необходимо исследовать влияние типа операционной системы на быстродействие и качество работы программного обеспечения для измерительного комплекса КИПРО1.

Программное обеспечение для измерительного комплекса должно обеспечить функционирование в следующих режимах: создание шаблона параметров, редактирование шаблона параметров, автоматическая и полуавтоматическая проверка низковольтных параметров (время замыкания контактов при срабатывании, время дребезга при срабатывании, время отпускания контактов при отпуске, время дребезга при отпуске, сопротивление обмотки, сопротивление замкнутых контактов, ток срабатывания, ток отпускания), автоматическая проверка высоковольтных пара-

метров (пробой изоляции, сопротивление изоляции), настройка и поверка измерительной системы.

### 3. Решение задачи

Для управления функционированием аппаратной части измерительного комплекса используем двунаправленный LPT-порт и интерфейс Centronics.

Адаптер параллельного интерфейса представляет собой набор регистров, расположенных в пространстве ввода/вывода. Регистры порта адресуются относительно базового адреса порта, стандартными значениями которого являются 3BCh, 378h и 278h. Порт может использовать линию запроса аппаратного прерывания, обычно IRQ7 или IRQ5. Порт имеет внешнюю 8-битную шину данных, 5-битную шину сигналов состояния и 4-битную шину управляющих сигналов.

BIOS поддерживает до четырех (иногда до трех) LPT-портов (LPT1-LPT4) своим сервисом – прерыванием INT 17h, обеспечивающим через них связь с принтерами по интерфейсу Centronics.

Параллельный порт для связи с принтером (или другим устройством) имеет базовый адрес &H378 (LPT1), &H278 (LPT2), &H3BC (LPT3). В данной статье мы будем рассматривать только LPT1. Адресное пространство данного порта занимает диапазон &H378-&H37A. Адрес &H378 называется базовым и служит для записи (чтения, но об этом попозже) данных в порт, на линии D0-D7. Адрес &H379 (базовый+1) предназначен для чтения битов состояния с устройства, подключенного к LPT-порту (принтер, сканер и т.д.). Адрес &H37A (базовый+2) служит для записи битов управления устройством, подключенного к LPT-порту (принтер, сканер и т.д.).

Управление LPT-портом зависит от ОС.

В операционной системе Windows можно разработать драйвер устройства или использовать следующие библиотеки vbio32.dll, inpout32.dll, dlportio.dll и т.д., которые работают на большинстве компьютеров с операционными системами семейства Windows. Из них только vbio32.dll и inpout32.dll не будут работать под Win2000. Используя одну из них (например DlportIO.dll), можно обращаться к порту и не использовать временные файлы, а обращаться только к оперативной памяти, что значительно ускоряет работу. Чтение байта (8 бит) из порта производим с помощью функции DlPortReadPortUchar(port). Запись в порт производим с помощью процедуры DlPortReadPortUchar(port, x). Время обращения к порту составляет около 100 мкс. В самой системе Windows существует два стандартных драйвера, обеспечивающих работу параллельных портов и устройств, подключенных к этим портам – parclass.sys и parport.sys. На самом деле, в системе могут быть и другие драйвера, обеспечивающие работу аппаратуры, реализующей параллельные порты, но их работа прозрачна для остальных компонентов системы. Драйвер parclass.sys используется драйверами устройств более высокого уровня, например драйвером сканера, или приложениями для прямого доступа к параллельному порту (row access). Parport.sys предназначен для управления самим параллельным портом. Этот драйвер не позволяет напрямую обеспечивать доступ через порты ввода-вывода к регистрам параллельного порта.

Он поддерживает PnP, регулирует доступ к портам, их захват и освобождение, контролирует режим работы каждого из присутствующих портов. Библиотека DlportIO.dll через несколько промежуточных драйверов находит данный драйвер и через него получает доступ к регистрам параллельного порта. Недостатком того, что Windows (особенно XP), позволяет косвенно работать с портами ввода-вывода, является расход значительной части времени.

В операционной системе Linux нет необходимости искать драйвера и библиотеки для работы с внешним устройством [5]. Получив специальные права, можно получить прямой доступ ко всей системе, даже к ядру. Это значительно повышает быстродействие при обмене информацией с портом. Процедуры доступа к портам ввода/вывода находятся в /usr/include/asm/io.h (или в linux/include/asm-i386/io.h в некоторых дистрибутивах ядра). Процедуры представляют собой встроенные (inline) макроопределения, и достаточно просто добавить #include <asm/io.h> в зависимости от места расположения [4].

Перед тем, как получить доступ к какому-нибудь порту, нужно дать программе права на это. Это выполняется, при помощи функции ioperm(from, num, turn\_on) (определенной в unistd.h и находящейся в ядре), где from это первый порт, а num это количество подряд идущих портов, которым нужно дать доступ. Например, ioperm(0x378, 3, 1) дает доступ к порту с 0x378 по 0x37A (адресное пространство порта LPT1). Последний аргумент - это двоичное значение, определяющее, дать ли доступ к портам (истина (1)) или запретить его (ложь (0)). Для включения портов, идущих не подряд, нужно вызывать ioperm() несколько раз. Для вызова ioperm() необходимо иметь права root; таким образом, запускать программу от пользователя root или установить на файл флаг setuid. После того, как определен доступ к портам, права суперпользователя не нужны. После окончания работы программы порты освобождаются автоматически. Выполнение setuid() для переключения на другого пользователя не отключает доступ к портам, данный ioperm(), но это происходит при fork (наследованный процесс теряет доступ, когда как у порождающего процесса он остается).

ioperm() может дать доступ только к портам с 0x000 по 0x3ff. Для других портов используется iopl() (который дает доступ ко всем портам сразу). Уровень 3 (iopl(3)) дает доступ программы ко всем портам ввода/вывода (доступ к неправильным портам может испортить конфигурацию системы).

Чтобы считать байт (8 бит) из порта, используем функцию inb(port), возвращающую считанный байт. Чтобы вывести байт в порт, вызываем процедуру outb(value, port). Чтобы считать компьютерное слово (16 бит) из портов x и x+1 (по одному байту из каждого образуют слово), вызываем функцию inw(x). Чтобы вывести слово в два порта, используйте outw(value, x). Все функции, работающие с портами, требуют, как минимум, около микросекунды для выполнения.

Макросы inb\_p(), outb\_p(), inw\_p() и outw\_p() работают аналогично вышеуказанным, но добавляют короткую (около 1 микросекунды) задержку после доступа к порту. Можно установить задержку около четырех микросекунд при помощи #define REAL\_LY\_SLOW\_IO перед #include <asm/io.h>. Обычно эти

макросы (за исключением случаев с `#define SLOW_IO_BY_JUMPING`, при которых выполнение становится менее точным) используют вывод в порт 0x80 в качестве задержки.

При написании программ, работающих под Linux воспользуемся Qt [2]. Qt – это библиотека классов C++ и набор инструментального программного обеспечения, предназначенных для построения многоплатформенных приложений с графическим интерфейсом. Qt представляет собой единую платформу для приложений, которые могут работать под управлением Windows 95/98/Me/2000/XP, Mac OS X, Linux, Solaris, HP-UX и других версий Unix [3]. Преимуществом такого программного обеспечения является то, что оно способно работать под Windows и Linux. При запуске программа сама определяет, из какой среды она запущена, и, либо с помощью макроопределений обращается к порту (Linux), либо подгружает библиотеку DlportIO.dll (Windows).

Таким образом, мы можем сравнить быстрдействие работы нашей программы, при работе ее в Windows и Linux с различными типами реле. Гистограмма зависимости времени измерения всех параметров реле от операционной системы представлена на рис. 1.

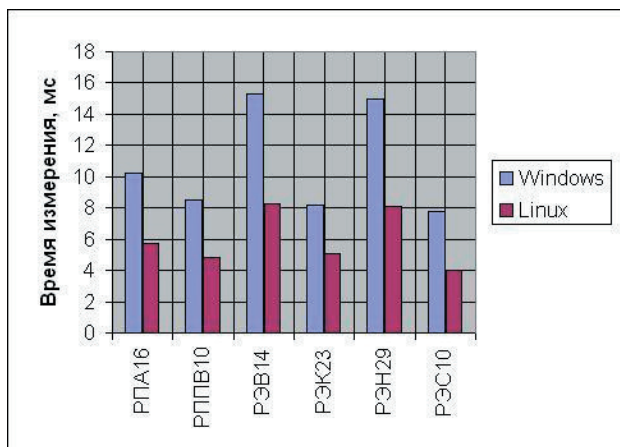


Рис. 1. Зависимость времени измерения параметров различных типов реле от типа ОС

#### 4. Выводы

Разработанное в процессе проведенных исследований программное обеспечение обладает рядом преимуществ.

Во-первых, оно написано на языке Qt, что позволяет не менять кода программы, а только компилятор, работать в таких операционных системах, как Windows 95/98/Me/2000/XP, Mac OS X, Linux, Solaris, HP-UX и других версий Unix.

Во-вторых, работа с ним в операционной системе Linux, позволяет получить прямой доступ ко всей системе и даже к ядру, следовательно, отпадает необходимость в написании драйвера для работы с внешним устройством. Это приблизительно на два порядка повышает быстрдействие при обмене информации с портом по сравнению с Windows XP (в зависимости от типа драйвера в Windows).

В-третьих, использование собственных драйверов могут восприниматься в Windows, как вредоносные программы, что может привести к блокированию работы порта, в то время как, работая в Linux, мы заранее предупреждаем систему о том, с каким адресным пространством порта мы будем работать.

#### Литература

1. Горчаков В.С. Цифровой измерительный комплекс. – М.: Радио и связь, из серии "Научно-популярное издание", 1986 г. – 55 с.
2. Шлее М. Qt4. Профессиональное программирование на C++. – СПб.: БХВ-Петербург, 2007. – 880 с.
3. Земсков Ю.В. Qt4 на примерах. – СПб.: БХВ-петербург, 2008. – 608 с.
4. Огих П. Использование LPT-порта ПК для ввода-вывода информации. – СПб.: НТ Пресс, 2006. – 160 с.
5. Колисниченко Д. Mandriva Linux 2008 Руководство пользователя. – СПб.: Питер, 2009. – 272 с.