

# КОЛИБРИ 8×4

Руководство  
по эксплуатации

USB-реле с интерфейсом RS-485

ПРОМЫШЛЕННАЯ ЭЛЕКТРОНИКА



Студия разработки СпецПромДизайн

Разработка электроники и программного обеспечения ...это просто

Web: [www.spd.net.ru](http://www.spd.net.ru), E-mail: [info@spd.net.ru](mailto:info@spd.net.ru)

## СОДЕРЖАНИЕ

<b>ОПИСАНИЕ .....</b>	<b>3</b>
<b>ПРИМЕНЕНИЯ .....</b>	<b>3</b>
<b>ОСОБЕННОСТИ .....</b>	<b>3</b>
<b>ТЕХНИЧЕСКИЕ ХАРАКТЕРИСТИКИ .....</b>	<b>4</b>
<b>УСТРОЙСТВО И ПРИНЦИП РАБОТЫ .....</b>	<b>5</b>
<b>КОММУТАЦИЯ ИНДУКТИВНЫХ НАГРУЗОК .....</b>	<b>7</b>
<b>СТОРОЖЕВОЙ ТАЙМЕР .....</b>	<b>8</b>
<b>ПРОТОКОЛ ОБМЕНА ДАННЫМИ ПО USB .....</b>	<b>9</b>
<b>ПРОГРАММА USB-Relay .....</b>	<b>12</b>
<b>РЕЖИМ АВТОМАТИЧЕСКОГО СТОРОЖЕВОГО ТАЙМЕРА .....</b>	<b>15</b>
<b>ПРОТОКОЛ ОБМЕНА ДАННЫМИ ПО ИНТЕРФЕЙСУ RS-485 .....</b>	<b>16</b>
<b>ПРОТОКОЛ ОБМЕНА WAKE16 .....</b>	<b>22</b>
Описание протокола .....	22
Структура пакета .....	22
<b>ПРОГРАММА WakeControl .....</b>	<b>26</b>
<b>ОБНОВЛЕНИЕ ВСТРОЕННОГО ПРОГРАММНОГО ОБЕСПЕЧЕНИЯ .....</b>	<b>30</b>
<b>ПРОГРАММА «Test USB-Relay CS» .....</b>	<b>33</b>
<b>РАБОТА С USB-УСТРОЙСТВАМИ В ОПЕРАЦИОННОЙ СИСТЕМЕ LINUX .....</b>	<b>34</b>

## ОПИСАНИЕ

USB-реле предназначено для управления внешними электрическими цепями и нагрузками с персонального компьютера через USB-порт или через интерфейс RS-485. Устройство содержит восемь электромагнитных реле, позволяющих коммутировать нагрузку 250 В, 7 А и четыре входа для подключения внешних контактных датчиков, два из которых имеют оптронную развязку.



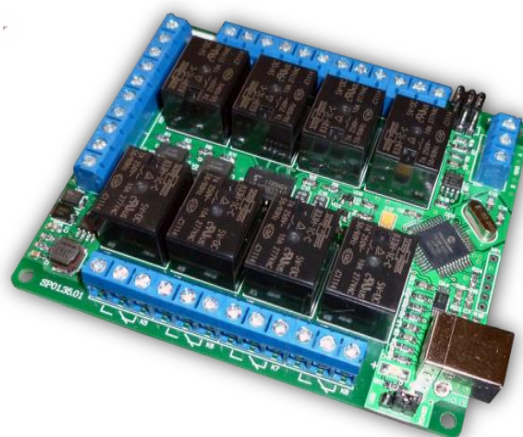
USB-реле также может выполнять функцию сторожевого таймера (Watchdog timer – WDT) для персонального компьютера. Поддерживаются операционные системы Windows и Linux.



***ВНИМАНИЕ!** Интерфейс USB обладает низкой помехозащищённостью, поэтому при использовании устройства в производственных, охранных и других ответственных системах управления объектами его следует подключать через интерфейс RS-485!*

## ПРИМЕНЕНИЯ

- Игровые терминалы
- Терминалы самообслуживания
- Системы «Умный дом»



## ОСОБЕННОСТИ

- Два интерфейса управления – USB и RS-485
- Два гальванически развязанных входа
- Малые габариты
- Функция сторожевого таймера
- Простой протокол передачи данных

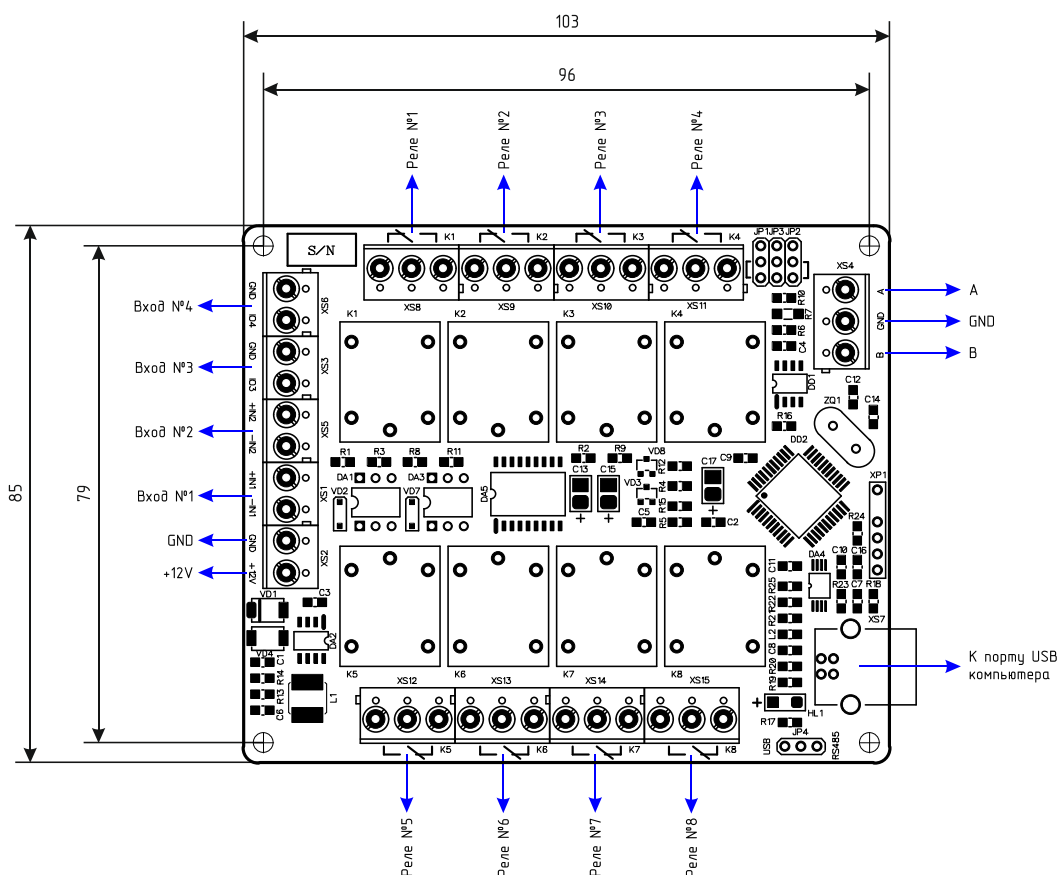
## ТЕХНИЧЕСКИЕ ХАРАКТЕРИСТИКИ

Напряжение питания от USB-порта .....	5 В ± 10%
Максимальный потребляемый ток от USB-порта .....	500 мА
Напряжение питания от внешнего источника .....	9 ÷ 18 В
Максимальный потребляемый ток от внешнего источника .....	300 мА
Количество входов без развязки (замыкание на землю) .....	2
Количество входов с оптронной развязкой .....	2
Количество электромагнитных реле .....	8
Параметры реле .....	250 В, 7А
Максимальное напряжение, подаваемое на оптронный вход .....	30 В
Скорость обмена данными по интерфейсу RS-485 .....	115200 бит/сек
Максимальное количество устройств на линии RS-485 .....	32
Гальваническая развязка интерфейса RS-485 .....	отсутствует
Протокол обмена по интерфейсу RS-485 .....	WAKE16
Подтягивающие резисторы на линиях А, В .....	620 Ом
Терминатор линии .....	120 Ом
Габаритные размеры .....	103 × 86 × 25 мм
Температурный диапазон работы .....	от –40°С до +85°С
Относительная влажность воздуха .....	не более 90% при +35°С

Для разработки собственного программного обеспечения для персонального компьютера, осуществляющего обмен данными по протоколу WAKE16, можно использовать динамическую библиотеку WAKE16.dll. Она содержит набор функций, позволяющих принимать и отправлять пакеты WAKE16. Кроме того, в ней имеются функции для непосредственной работы с USB-реле «КОЛИБРИ 8×4» или ПЛК «КСЕМИКС-1». Эти функции реализуют все поддерживаемые данными устройствами команды. Библиотека написана на языке Delphi в среде Embarcadero RAD Studio.

Подробное описание библиотеки приведено в документе DS16002 «Динамическая библиотека WAKE16.dll».

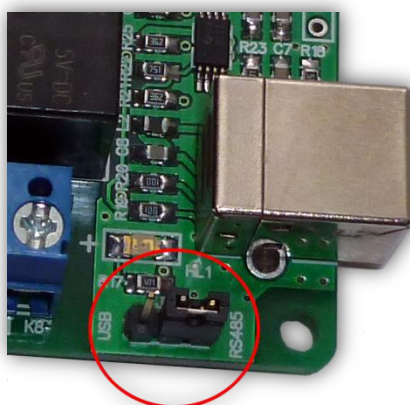
## УСТРОЙСТВО И ПРИНЦИП РАБОТЫ



Контакты реле и входов для подключения внешних датчиков выведены на соответствующие клеммники. Активным сигналом датчика для входов без оптронной развязки считается лог. 0 (замыкание входа на «землю» GND). Для входа с оптронной развязкой активным сигналом считается подача напряжения выше 3 В.

USB-реле может подключаться к персональному компьютеру либо через USB-порт, либо по интерфейсу RS-485 через специализированный преобразователь, например, «СОКЕТ-1» (RS-232 ⇔ RS-485), «СОКЕТ-2» (USB ⇔ RS-485), «СОКЕТ-3» (Bluetooth ⇔ RS-485) или «СОКЕТ-4» (Ethernet ⇔ RS-485).

Выбор необходимого интерфейса для работы (USB или RS-485) осуществляется переключкой JP4:



Правое положение соответствует выбору интерфейса RS-485, левое положение – USB

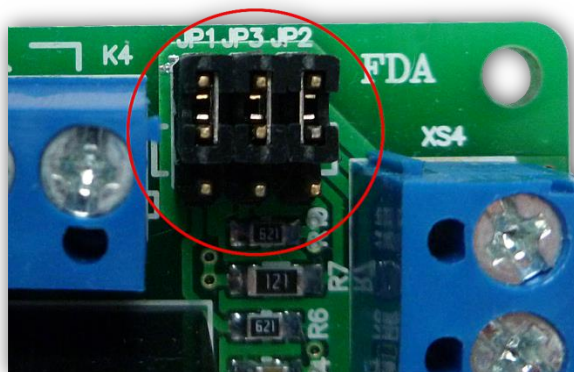
(на рисунке выше показан выбор интерфейса RS-485).

Подключение к порту USB персонального компьютера осуществляется посредством кабеля USB A-B. Питается устройство непосредственно от USB-порта. Если его нагрузочной способности будет недостаточно для включения всех электромагнитных реле устройства, то необходимо использовать внешний источник питания, подключаемый к клеммнику XS2.

После первого подключения к компьютеру операционная система запросит специализированный драйвер. Необходимо указать путь к папке с драйверами и далее следовать указаниям операционной системы.

После этого в операционной системе появится виртуальный COM-порт, номер которого можно уточнить в «Диспетчере устройств».

При работе с интерфейсом RS-485 линию связи необходимо подключить к клеммнику XS4. USB-реле имеет внутренние подтягивающие резисторы и резистор-терминатор. Если устройство находится в начале или конце линии связи, то необходимо установить три перемычки JP1, JP3 (подтягивающие резисторы) и JP2 (терминатор):



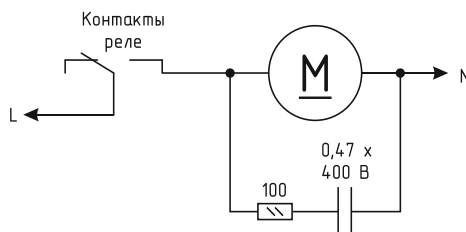
Включённому состоянию соответствует их нижнее положение (на рисунке выше показано выключенное состояние).



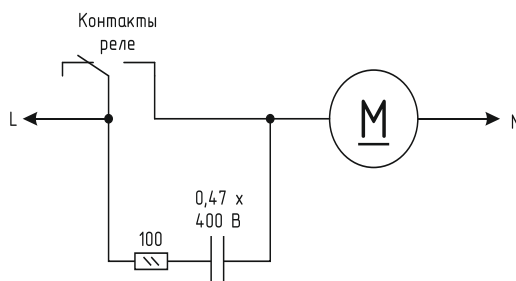
**ВНИМАНИЕ!** USB-реле не имеет гальванической развязки интерфейса RS-485, поэтому необходимо контролировать падение напряжения на линии связи во избежание проблем с передачей данных!

## КОММУТАЦИЯ ИНДУКТИВНЫХ НАГРУЗОК

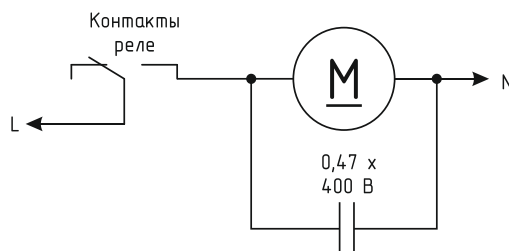
При коммутации индуктивных нагрузок (двигатели, электромагнитные клапаны и т.п.) в момент размыкания контактов реле может образовываться электрическая дуга, приводящая к возникновению сильных электромагнитных помех, способных привести к нестабильности работы устройства. Для подавления этих помех можно использовать внешние искрогасящие RC-цепочки, подключаемые параллельно нагрузке:



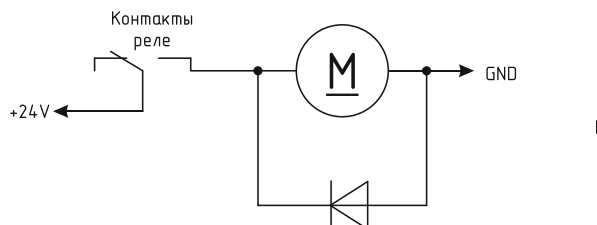
В случаях, когда затруднительно подключить такую цепь к обоим контактам нагрузки, можно подсоединить её параллельно самим контактам реле:



Иногда при малых мощностях нагрузки достаточным будет использование только одного конденсатора:



В случае если нагрузка будет питаться постоянным током, вместо RC-цепочки можно использовать просто диод для гашения ЭДС самоиндукции:



Во всех приведённых выше схема резистор должен иметь мощность не менее 0,25 Вт. Конденсатор желательно использовать металлоплёночный с рабочим напряжением не менее 400 В, например из серии К73-17. Диод подойдёт любой импульсный соответствующей мощности.

### СТОРОЖЕВОЙ ТАЙМЕР

Встроенное ПО USB-реле поддерживает режим сторожевого таймера, который можно использовать, например, для контроля за персональным компьютером.

Этот режим включается специальной командой при загрузке компьютера. Данной командой задаётся период сторожевого таймера в секундах. При этом устройство начинает отсчёт времени. Компьютер должен периодически сбрасывать внутренний счётчик сторожевого таймера специальной командой. Если же он не сделает этого в течение заданного периода (например, в случае сбоя или «зависания»), то устройство включит на 2 сек одно из реле, которое следует подключить к кнопке «RESET» персонального компьютера, что приведёт к его аппаратной перезагрузке. Сторожевой таймер при этом автоматически выключится.

Во время отсчёта времени светодиод, соответствующий выбранному реле, будет кратковременно вспыхивать с частотой 1 Гц.



## ПРОТОКОЛ ОБМЕНА ДАННЫМИ ПО USB

USB-реле поставляется со встроенным ПО, организующем только управление электромагнитными реле и считывание состояния входов по командам компьютера. Устройство работает по принципу «запрос-ответ». Данные передаются должны передаваться единым пакетом в 16-ричном текстовом виде (по два символа на один логический байт). При этом 16-битные значения передаются старшим байтом вперёд. Символьные и строковые данные передаются в кодировке Windows-1251.

Формат пакета данных приведён ниже:

: <cmd> <data0> <data1> ... <dataN> ;

Пакет всегда должен начинаться с двоеточия и заканчиваться точкой с запятой. Между этими двумя символами допустимы только 16-ричные цифры (0...9 и A...F). После двоеточия идёт код команды, а после него данные. Количество данных не передаётся и вычисляется приёмной стороной автоматически. Пример пакета показан ниже (пробелы между символами вставлены для удобства восприятия и в реальной команде должны быть исключены):

:01 01 00 3A 05;

Здесь код команды – 01 и четыре байта данных – 01, 00, 3A, 05.

Встроенное ПО устройства поддерживает шесть команд. При успешном выполнении принятой команды выдаётся ответ с кодом команды 0x33, при ошибке – с кодом 0x22.

-----  
***cmSetRelay*** – включение/отключение электромагнитных реле.

***Описание:*** задаёт индивидуальное состояние каждого электромагнитного реле. Включённому состоянию соответствует значение больше 0x00, выключенному – 0x00.

***Код команды:*** 0x01

***Данные:*** <relay1>, <relay2>, <relay3>, <relay4>, <relay5>, <relay6>, <relay7>, <relay8>

***Ответ:***

***Команда:*** 0x33 при успешном выполнении команды и 0x22 при ошибке

***Данные:*** нет  
-----

***cmGetInput*** – получение состояния входов для подключения внешних датчиков.

***Описание:*** считывает состояние входов внешних датчиков. Сработавшему датчику соответствует значение 0x01, выключенному – 0x00.

***Код команды:*** 0x02

***Данные:*** нет

***Ответ:***

Команда: 0x33

Данные: <i1>, <i2>, <i3>, <i4> – состояние соответствующего входа.

---

**cmGetInfo** – получение информации об устройстве.

Описание: возвращает информацию о названии устройства, количестве установленных электромагнитных реле и датчиков.

Код команды: 0x03

Данные: нет

Ответ:

Команда: 0x33

Данные:

0x08 – количество реле

0x04 – количество датчиков

<info\_len> – длина строки с названием устройства

<str<sub>0</sub>>...<str<sub>info\_len</sub>> – текстовая строка с названием устройства

---

**cmGetRelay** – получение текущего состояния электромагнитных реле.

Описание: считывает состояние электромагнитных реле. Включённому реле соответствует значение 0x01, выключенному – 0x00.

Код команды: 0x04

Данные: нет

Ответ:

Команда: 0x33

Данные: <relay1>, <relay2>, <relay3>, <relay4>, <relay5>, <relay6>, <relay7>, <relay8> – состояние соответствующего реле

---

**cmSetWDT** – запуск сторожевого таймера.

Описание: задаёт период сторожевого таймера. При значении периода больше нуля таймер запускается, при нулевом значении – останавливается.

Код команды: 0x5A

Данные:

<timeoutWDT<sub>16</sub>> – период сторожевого таймера в секундах

<relayWDT> – номер реле, которым будет управлять сторожевой таймер

Ответ:

Команда: 0x33

Данные: нет

***cmResetWDT*** – сброс сторожевого таймера.

Описание: сбрасывает внутренний счётчик сторожевого таймера в ноль. Отсчёт времени будет продолжаться.

Код команды: 0x5B

Данные: нет

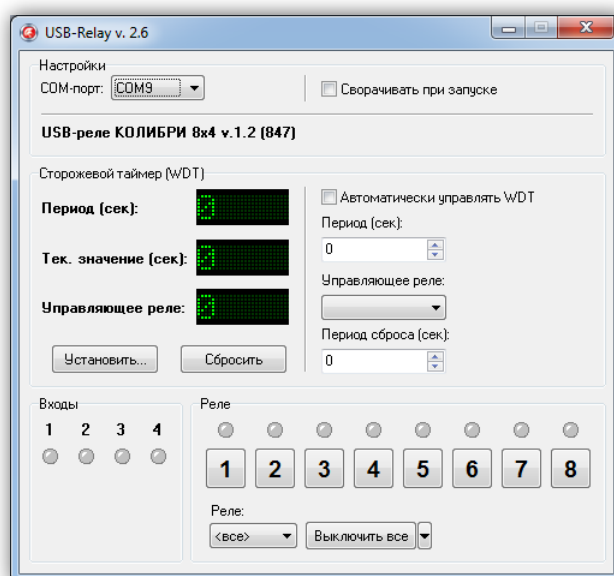
Ответ:

Команда: 0x33

Данные: нет

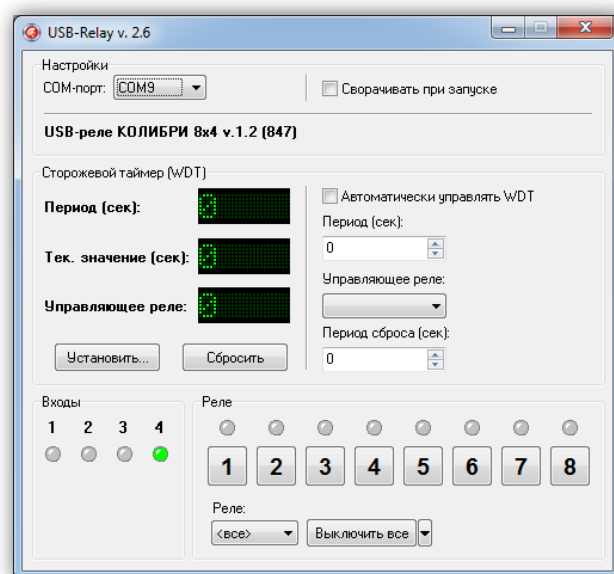
## ПРОГРАММА USB-Relay

Программа USB-Relay предназначена для управления USB-реле через USB-порт. Внешний вид программы показан на рисунке ниже:



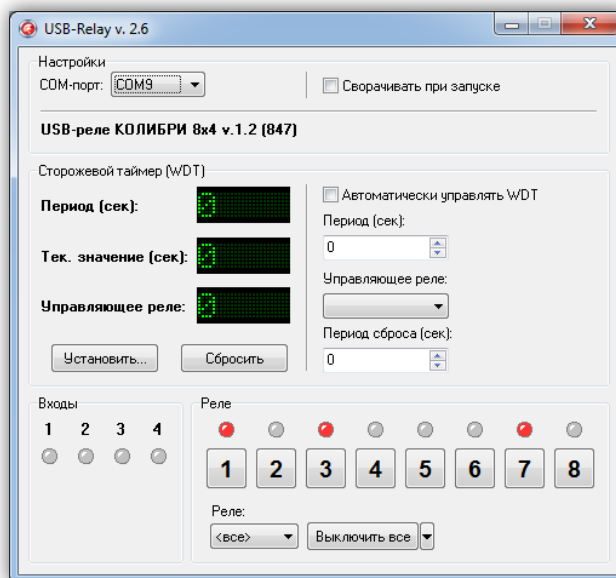
После запуска необходимо выбрать COM-порт, соответствующий USB-реле. Если порт был выбран верно и устройство исправно, программа будет отображать текстовое название устройства. В скобках после номера версии указывается адрес устройства, используемый при управлении им по протоколу WAKE16).


Программа будет постоянно опрашивать состояние входов для подключения внешних датчиков и отображать их состояние в поле «Входы». Сработавшему датчику будет соответствовать зелёный цвет индикатора:

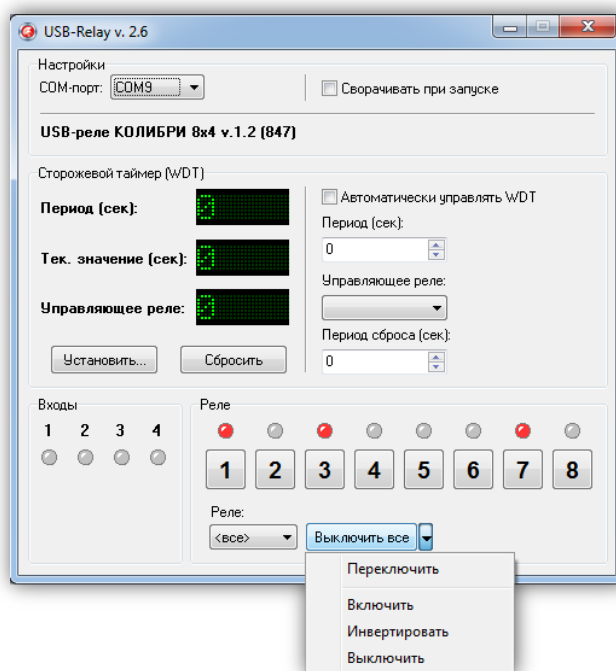


Управление электромагнитными реле может осуществляться двумя способами. Можно просто нажать соответствующую кнопку ( 1 2 3 4 5 6 7 8 ), при этом первое нажатие включит реле, второе – выключит. Включённому состоянию соответствует красный

цвет индикатора, расположенного над кнопкой:



При втором способе управления необходимо отметить нужное реле в списке «Реле», а затем выбрать требуемую команду в выпадающем меню кнопки «Выключить все» (нажать кнопку со стрелкой ):



Следует отметить, что в списке «Реле» можно выбрать пункт «<все>», в этом случае действие соответствующей команды будет распространяться одновременно на все реле.

Ниже приведено описание команд для управления реле:

«Перекл. (все)» – выполняет «перезапуск» реле – состояние выбранного реле изменится на противоположное на время 1,5 сек, после чего реле вернётся в прежнее состояние. В течение времени выполнения данной команды выбранное из списка реле будет недоступно для ручного управления.

«Включить» – включает выбранное реле.

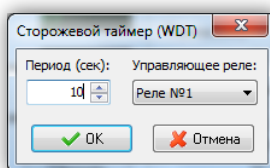
«Инвертировать» – изменяет состояние выбранного реле на противоположное.

«Выключить» – выключает выбранное реле.

При нажатии на самую кнопку «Выключить все» будет произведено выключение всех реле независимо от того, какой пункт выбран в списке «Реле».

Элементы поля «Сторожевой таймер (WDT)» позволяют осуществлять ручное управление сторожевым таймером.

При нажатии кнопки «Установить...» откроется следующее диалоговое окно:



Здесь задаётся период сторожевого таймера и выбирается реле, которым он будет управлять. После нажатия кнопки «ОК» таймер будет запущен и в главном окне программы в строке «Тек. значение (сек)» будет отображаться текущее значение внутреннего счётчика. Кнопка, соответствующая управляющему реле, будет недоступна на время работа сторожевого таймера:



Как только его значение достигнет значения периода («Период (сек)»), будет включено выбранное реле на время 2 сек. После этого сторожевой таймер автоматически выключится.

Если в процессе отсчёта периодически нажимать кнопку «Сбросить», то будет осуществляться сброс внутреннего счётчика и включения реле не произойдёт.



*Во время работы сторожевого таймера закрыть программу невозможно! Вначале нужно остановить таймер путём задания нулевого периода.*

## РЕЖИМ АВТОМАТИЧЕСКОГО СТОРОЖЕВОГО ТАЙМЕРА

Программа USB-Relay позволяет реализовать функцию защиты компьютера от зависания с использованием функции сторожевого таймера, аппаратно реализованного в USB-реле.

Для настройки этого режима необходимо выполнить следующие действия.

1. Снять отметку пункта «Автоматически управлять WDT». При этом станут доступны поля «Период (сек)», «Управляющее реле» и «Период сброса (сек)».

2. В поле «Период (сек)» следует задать интервал времени, отсчитываемый сторожевым таймером, в течение которого необходимо хотя бы раз выполнить его сброс.

3. В поле «Управляющее реле» выбирается реле, которым будет управлять сторожевой таймер.

4. В поле «Период сброса (сек)» необходимо задать интервал времени, через который сторожевой таймер будет сбрасываться программой USB-Relay. Этот интервал рекомендуется задавать равным примерно 10...20% от значения «Период (сек)».

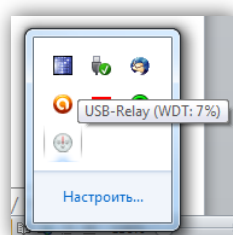
5. Установить отметку пункта «Автоматически управлять WDT».

6. Перезапустить программу.

После этого автоматически запустится отсчёт времени сторожевого таймера, и также будет выполняться его сброс с заданным периодом.

В случае зависания компьютера сброса не произойдёт, сторожевой таймер переполнится и при помощи заданного реле осуществит перезапуск компьютера.

При использовании данной функции программы рекомендуется отметить пункт «Сворачивать при запуске». В этом случае программа будет запускаться в свёрнутом виде. Значок программы будет отображаться в панели иконок, рядом с часами. Открытие окна программы можно осуществить двойным щелчком мыши на этом значке:



При наведении на значок программы курсора мыши будет выведено всплывающее сообщение, в котором будет отображаться отсчёт времени сторожевого таймера в процентах. На рисунке выше внутренний счётчик сторожевого таймера отсчитал уже 7% от общего периода. Если до достижения 100% сторожевой таймер не будет сброшен, то USB-реле аппаратно перезагрузит компьютер.

## ПРОТОКОЛ ОБМЕНА ДАННЫМИ ПО ИНТЕРФЕЙСУ RS-485

Обмен данными с USB-реле осуществляется по протоколу WAKE16 на скорости 115200 бит/сек. Для управления используются пять команд. Если USB-реле примет пакет с неверным форматом команды, то оно передаст ответ с командой 0x22 и пустыми данными.

Каждое USB-реле имеет собственный адрес, который можно сменить через программу WakeControl. По умолчанию адрес всех USB-реле 0x7FFF (32767). Этот адрес используется при обмене данными с конкретным устройством на линии RS-485. Ответ от устройства всегда содержит нулевое значение поля адреса, что позволяет всегда отличать его от остальных принимаемых пакетов данных.

Ниже приводится описание команд, реализованных в USB-реле. Данные размерностью более одного байта помечаются числом битов в нижнем индексе (например, <номер\_блока<sub>16</sub>>). Многобайтные значения передаются старшими байтами вперёд. Строки передаются в формате языка Си – с завершающим нулём. Символьные и строковые данные передаются в кодировке Windows-1251.

---

***cmSetRelay*** – включение/отключение электромагнитных реле.

Описание: задаёт индивидуальное состояние каждого электромагнитного реле.

Команда: 0x51

Данные: <relay\_mask >

Значение <relay\_mask> содержит маску включения реле. Установленный бит в 0-ом разряде соответствует включённому состоянию реле №1, в 1-ом разряде – реле №2, во 2-ом разряде – реле №3 и т.д.

Ответ:

Команда: 0x33

Данные: нет

Пример: включение реле №2 на устройстве с адресом 24:

0xC0 0x80 0x18 0x51 0x00 0x01 0x02 0x10 0xD5

0xC0 – начало пакета

0x80, 0x18 – адрес USB-реле (0x0018) с установленным старшим битом

0x51 – код команды

0x00, 0x01 – один байт данных

0x02 – маска реле (0b00000010 – включено реле №2)

0x10, 0xD5 – CRC16

Ответ:

0xC0 0x33 0x00 0x00 0x50 0xF9

0xC0 – начало пакета



0x33 – код команды (пакет данных имеет верный формат)  
0x00, 0x00 – нет данных  
0x50, 0xF9 – CRC16

---

***cmGetRelayInput*** – текущее состояние входов и электромагнитных реле.

Описание: считывает текущее состояние электромагнитных реле и входов для подключения внешних датчиков.

Команда: 0x52

Данные: нет

Ответ:

Команда: 0x33

Данные: <input\_mask>, <relay\_mask>

Значение <input\_mask> содержит маску датчиков. Установленный бит в 0-ом разряде соответствует сработавшему датчику №1, в 1-ом разряде – датчику №2, во 2-ом разряде – датчику №3, в 3-ем разряде – датчику №4.

Значение <relay\_mask> содержит маску реле. Установленный бит в 0-ом разряде соответствует включённому состоянию реле №1, в 1-ом разряде – реле №2, во 2-ом разряде – реле №3 и т.д.

Пример: считывание текущего состояния электромагнитных реле и входов для подключения внешних датчиков на устройстве с адресом 24:

0xC0 0x80 0x18 0x52 0x00 0x00 0xAA 0xFF

0xC0 – начало пакета

0x80, 0x18 – адрес USB-реле (0x0018) с установленным старшим битом

0x52 – код команды

0x00, 0x00 – нет данных

0xAA, 0xFF – CRC16

Ответ:

0xC0 0x33 0x00 0x02 0x03 0x02 0x45 0x57

0xC0 – начало пакета

0x33 – код команды (пакет данных имеет верный формат)

0x00, 0x02 – два байта данных

0x03 – состояние входов (на 1-ом и 2-ом активный уровень)

0x02 – состояние электромагнитных реле (реле №2 включено)

0x45, 0x57 – CRC16

---

***cmSetWDT*** – запуск сторожевого таймера.

Описание: задаёт период сторожевого таймера. При значении периода больше нуля таймер запускается, при нулевом значении – останавливается.

Код команды: 0x5A

Данные: <timeoutWDT<sub>16</sub>> – период сторожевого таймера в секундах

<relayWDT> – номер реле, которым будет управлять сторожевой таймер

Ответ:

Команда: 0x33

Данные: нет

Пример: запуск сторожевого таймера с периодом в 5 секунд с исполнительным реле №2 на устройстве с адресом 32767:

0xC0 0xFF 0xFF 0x5A 0x00 0x03 0x00 0x05 0x02 0x11 0x08

0xC0 – начало пакета

0xFF, 0xFF – адрес USB-реле (0x7FFF) с установленным старшим битом

0x5A – код команды

0x00, 0x03 – три байта данных

0x00, 0x05 – период сторожевого таймера (5 секунд)

0x02 – исполнительное реле №2

0x11, 0x08 – CRC16

Ответ:

0xC0 0x33 0x00 0x00 0x50 0xF9

0xC0 – начало пакета

0x33 – код команды (пакет данных имеет верный формат)

0x00, 0x00 – нет данных

0x50, 0xF9 – CRC16

---

***cmResetWDT*** – сброс сторожевого таймера.

Описание: сбрасывает внутренний счётчик сторожевого таймера в ноль. Отсчёт времени будет продолжаться.

Команда: 0x5B

Данные: нет

Ответ:

Команда: 0x33

Данные: нет

Пример: сброс сторожевого таймера на устройстве с адресом 32767:

0xC0 0xFF 0xFF 0x5B 0x00 0x00 0xAA 0x45

0xC0 – начало пакета

0xFF, 0xFF – адрес ПЛК (0x7FFF) с установленным старшим битом

0x5B – код команды

0x00, 0x00 – нет данных

0xAA, 0x45 – CRC16

Ответ:

0xC0 0x33 0x00 0x00 0x50 0xF9

0xC0 – начало пакета

0x33 – код команды (пакет данных имеет верный формат)

0x00, 0x00 – нет данных

0x50, 0xF9 – CRC16

**cmGetInfo** – получение информации об устройстве.

Описание: считывает информацию о названии устройства, количестве установленных электромагнитных реле и датчиков.

Команда: 0x71

Данные: нет

Ответ:

Команда: 0x33

Данные:

<mode> – код режима работы (0x11 – основной режим, 0x10 – режим загрузчика)

<version> – номер версии программного обеспечения

<build<sub>16</sub>> – номер сборки программного обеспечения

<str<sub>0</sub>>, <str<sub>1</sub>> ... <0x00> – текстовая строка с названием устройства

0x09 – сигнатура микроконтроллера (PIC18F4550)

0x00, 0x00, 0x00 – объем внешней памяти (0 байт)

0x04 – сигнатура блока данных (целое число)

<str<sub>0</sub>>, <str<sub>1</sub>> ... <0x00> – название блока «Кол-во реле»

0x02 – размер данных

<relay\_cnt<sub>16</sub>> – количество электромагнитных реле

0x04 – сигнатура блока данных (целое число)

<str<sub>0</sub>>, <str<sub>1</sub>> ... <0x00> – название блока «Кол-во входов»

0x02 – размер данных

<input\_cnt<sub>16</sub>> – количество входов для подключения внешних датчиков

0x04 – сигнатура блока данных (целое число)

<str<sub>0</sub>>, <str<sub>1</sub>> ... <0x00> – название блока «Сост-е реле»

0x02 – размер данных

<relay\_state<sub>16</sub>> – текущее состояние электромагнитных реле

0x04 – сигнатура блока данных (целое число)

<str<sub>0</sub>>, <str<sub>1</sub>> ... <0x00> – название блока «Сост-е входов»

0x02 – размер данных

<input\_state<sub>16</sub>> – текущее состояние входов

для подключения внешних датчиков

0x01 – сигнатура блока данных (строка)

<str<sub>0</sub>>, <str<sub>1</sub>> ... <0x00> – название блока «DateTime FW»

<size> – размер данных (длина строки + 1)

<str<sub>0</sub>> ... <str<sub>size - 1</sub>>, <0x00> – текстовая строка даты и времени

компиляции встроенного программного обеспечения

микроконтроллера

Пример: считывание информации об устройстве с адресом 24:

0xC0 0x80 0x18 0x71 0x00 0x00 0x46 0xA0

0xC0 – начало пакета

0x80, 0x18 – адрес USB-реле (0x0018) с установленным старшим битом

0x71 – код команды

0x00, 0x00 – нет данных

0x46, 0xA0 – CRC16

Ответ:

0xC0 0x33 0x00 0x02 0x03 0x02 0x45 0x57

0xC0 – начало пакета

0x33 – код команды (пакет данных имеет верный формат)

0x00, 0x84 – 132 байта данных

0x11 – основной режим работы

0x10 – версии программного обеспечения 1.0

0x00, 0x25 – номер сборки программного обеспечения (37)

0x55, 0x53, 0x42, 0x2D, 0xF0, 0xE5, 0xEB, 0xE5, 0x20, 0xCA,

0xCE, 0xCB, 0xC8, 0xC1, 0xD0, 0xC8, 0x20, 0x38, 0x78,

0x34, 0x00 – название устройства USB-реле «КОЛИБРИ 8х4»

0x09 – сигнатура микроконтроллера (PIC18F4550)  
0x00, 0x00, 0x00 – объем внешней памяти (0 байт)

0x04 – сигнатура блока данных (целое число)  
0xCA, 0xEE, 0xEB, 0x2D, 0xE2,  
0xEE, 0x20, 0xF0, 0xE5, 0xEB,  
0xE5, 0x00 – название блока «Кол-во реле»

0x02 – размер блока данных 2 байта  
0x00, 0x08 – восемь электромагнитных реле

0x04 – сигнатура блока данных (целое число)  
0xCA, 0xEE, 0xEB, 0x2D, 0xE2, 0xEE,  
0x20, 0xE2, 0xF5, 0xEE, 0xE4, 0xEE,  
0xE2, 0x00 – название блока «Кол-во входов»

0x02 – размер блока данных 2 байта  
0x00, 0x04 – четыре входа для подключения внешних датчиков

0x04 – сигнатура блока данных (целое число)  
0xD1, 0xEE, 0xF1, 0xF2, 0x2D,  
0xE5, 0x20, 0xF0, 0xE5, 0xEB  
0xE5, 0x00 – название блока «Сост-е реле»

0x02 – размер блока данных 2 байта  
0x00, 0x00 – все реле выключены

0x04 – сигнатура блока данных (целое число)  
0xD1, 0xEE, 0xF1, 0xF2, 0x2D, 0xE5,  
0x20, 0xE2, 0xF5, 0xEE, 0xE4, 0xEE,  
0xE2, 0x00 – название блока «Сост-е входов»

0x02 – размер блока данных 2 байта  
0x00, 0x00 – сработавших датчиков нет

0x01 – сигнатура блока данных (текстовая строка)  
0x44, 0x61, 0x74, 0x65, 0x54,  
0x69, 0x6D, 0x65, 0x20, 0x46,  
0x57, 0x00 – название блока «DateTime FW»

0x14 – размер блока данных 20 байт  
0x30, 0x37, 0x2E, 0x30,  
0x31, 0x2E, 0x32, 0x30, 0x31,  
0x32, 0x20, 0x31, 0x35, 0x3A,  
0x31, 0x33, 0x3A, 0x30,  
0x34, 0x00 – текстовая строка «07.01.2012 15:13:04»  
0xE5, 0xA6 – CRC16

## ПРОТОКОЛ ОБМЕНА WAKE16

### Описание протокола

Протокол WAKE16 является логическим уровнем интерфейса управления оборудованием с помощью асинхронного последовательного канала. Физический уровень интерфейса протоколом не определяется. В качестве него может использоваться, например, RS-232 или RS-485. Протокол позволяет производить обмен пакетами данных длиной до  $2^{16}$  байт с адресуемыми устройствами, которых может быть до  $2^{15} - 1$ . Последовательный канал должен быть сконфигурирован следующим образом:

- 1) число бит в посылке – 8;
- 2) количество стоп-бит – 1;
- 3) бит чётности – нет;
- 4) скорость обмена – 1200...921600 бит/сек;
- 5) использование линий управления модемом – произвольное.

Основой протокола WAKE16 является протокол SLIP (UNIX™ Serial Link Interface Protocol). Передача данных осуществляется в двоичном виде, то есть используются все возможные значения байта (0x00...0xFF). Для передачи служебной информации зарезервированы два кода: FEND = 0xC0 (Frame End) и FESC = 0xDB (Frame Escape). Управляющий код FEND служит для обозначения начала посылки, а код FESC служит для передачи ESC-последовательностей. Если в потоке данных встречаются байты, значения которых совпадают с управляющими кодами, производится подмена этих байт ESC-последовательностями. Этот механизм называют байт-стаффингом. Код FEND заменяется последовательностью <FESC>, <TFEND>, а код FESC – последовательностью <FESC>, <TFESC>, где TFEND = 0xDC (Transposed FEND), TFESC = 0xDD (Transposed FESC). Коды TFEND и TFESC являются управляющими только в ESC-последовательностях, поэтому при передаче данных они в подмене не нуждаются.

### Структура пакета

Пакет всегда начинается управляющим кодом FEND (0xC0). Затем следует необязательный байт адреса, после которого идёт байт команды. За ним следует байт количества данных и собственно байты данных. Завершает пакет байт контрольной суммы CRC16:

FEND	ADDR	CMD	N	Data <sub>1</sub>	...	Data <sub>N</sub>	CRC16
------	------	-----	---	-------------------	-----	-------------------	-------

**FEND:** управляющий код FEND (C0h) является признаком начала пакета. Благодаря стаффингу, этот код больше нигде в потоке данных не встречается, что позволяет в любой ситуации однозначно определять начало пакета.

**ADDR:** 16-битное слово адреса используется для адресации отдельных устройств. На

практике распространена ситуация, когда управление осуществляется только одним устройством. В таком случае слово адреса не требуется, и его можно не передавать. Вместо него сразу за кодом FEND передаётся байт команды CMD.

Часто встречается ситуация, когда в сети имеется всего одно устройство, которое может инициировать обмен данными (мастер-устройство), и при этом обмен ведётся по принципу «запрос – ответ». Примером могут являться сети, построенные на основе интерфейса RS-485. В этом случае пакет данных от мастера содержит адрес устройства, которому этот пакет предназначен. Однако в ответном пакете указывать адрес мастера нет необходимости, так как он единственный в сети.

Для того чтобы можно было однозначно установить, к адресу или команде относится второй байт пакета, введены некоторые ограничения. Для адресации используется 15 бит, а старший бит, передаваемый вместе с адресом, должен всегда быть установлен в единицу:

	D15	...	D1	D0
ADDR =	1	...	A1	A0

Иногда возникает необходимость передать какую-то команду или данные сразу всем устройствам. Для этого предусмотрен коллективный вызов, который осуществляется путём передачи нулевого адреса (учитывая единичный старший бит, в этом случае передаваемый адрес равен 0x8000). Нужно отметить, что передача в пакете нулевого адреса полностью аналогична передаче пакета без адреса. Поэтому при реализации протокола можно автоматически исключать нулевой адрес из пакета. Учитывая разрядность адреса и один зарезервированный адрес для коллективного вызова, максимальное количество адресуемых устройств составляет  $2^{15} - 1$ .

Если возникает необходимость передать значение адреса, содержащие значения 0x40 или 0x5B (передаваемый байт в этом случае будет равен 0xC0h или 0xDB), то производится байт-стаффинг. Поэтому следует учитывать, что устройства с такими адресами требуют большей длины пакета. Это может быть заметно в тех случаях, когда используются короткие пакеты. В таких случаях следует избегать назначения устройствам названных адресов.

**CMD:** байт команды всегда должен иметь нулевой старший бит:

	D7	D6	D5	D4	D3	D2	D1	D0
CMD =	0	C6	C5	C4	C3	C2	C1	C0

Таким образом, код команды занимает 7 бит, что позволяет передавать до 128 различных команд. Коды команд выбираются произвольно в зависимости от нужд приложения. Рекомендуется использовать несколько стандартных кодов команд:

Код	Описание команды
0x 01	Передача управления основной программе центрального контроллера
0x 02	Передача управления загрузчику центрального контроллера
0x 03	Чтение блока FLASH-памяти центрального контроллера
0x 04	Запись блока FLASH-памяти центрального контроллера
0x 08	Сброс устройства
0x 71	Запрос информации об устройстве
0x7A	Изменение адреса устройства

Команды обычно имеют несколько параметров, которые передаются далее в виде пакета данных.

Поскольку код команды всегда имеет нулевой старший бит, этот код никогда не совпадает с управляющими кодами. Поэтому при передаче команды байт-стаффинг никогда не производится.

**N:** имеет значение, равное числу передаваемых байт данных. Под величину N отводится два байта:

	D15	...	D1	D0
N =	0	...	N1	N0

Таким образом, один пакет может содержать до  $2^{16}$  байт данных. Значение N не учитывает служебные байты пакета FEND, ADDR, CMD, N и CRC16. В результате байт-стаффинга фактическая длина пакета может возрасти. Значение N не учитывает этот факт и отражает количество полезных байт данных (то есть значение N всегда таково, как будто байт-стаффинг не осуществляется). Если передаваемая команда не имеет параметров, то передаётся  $N = 0x0000$  и байты данных опускаются.

Если возникает необходимость передать значение N, равное  $0xC0$  или  $0xDB$ , то производится байт-стаффинг. Однако при таких больших значениях N длина пакета столь велика, что его удлинение ещё на один байт практически незаметно.

**Data<sub>1</sub>...Data<sub>N</sub>:** байты данных, количество которых определяется значением N. При  $N = 0$  байты данных отсутствуют. Байты данных могут иметь любое значение, кроме FEND ( $0xC0$ ) и FESC ( $0xDB$ ). Если возникает необходимость передать одно из этих значений, то производится байт-стаффинг.

**CRC:** 16-битное слово контрольной суммы CRC16. Контрольная сумма рассчитывается перед операцией байт-стаффинга для всего пакета, начиная с байтов адреса (или байта команды, если адреса нет) и заканчивая последним байтом данных. При вычислениях используется значение адреса вместе со старшим битом.

Для расчёта контрольной суммы используется полином  $CRC16 = X^{16} + X^{12} + X^5 + 1$



([1] 0001 0000 0010 0001 = 0x1021). Значение CRC16 перед вычислением инициализируется числом 0xFFFF. Если CRC16 содержит значения байтов 0xC0 и 0xDB, то они заменяются ESC-последовательностями.

Функция для вычисления CRC16 показана ниже (используется «зеркальный» полином 1000 0100 0000 1000 [1] = 0x8408):

```
void UpdateCRC16 (char b)
{
    char i;

    CRC16 ^= b;

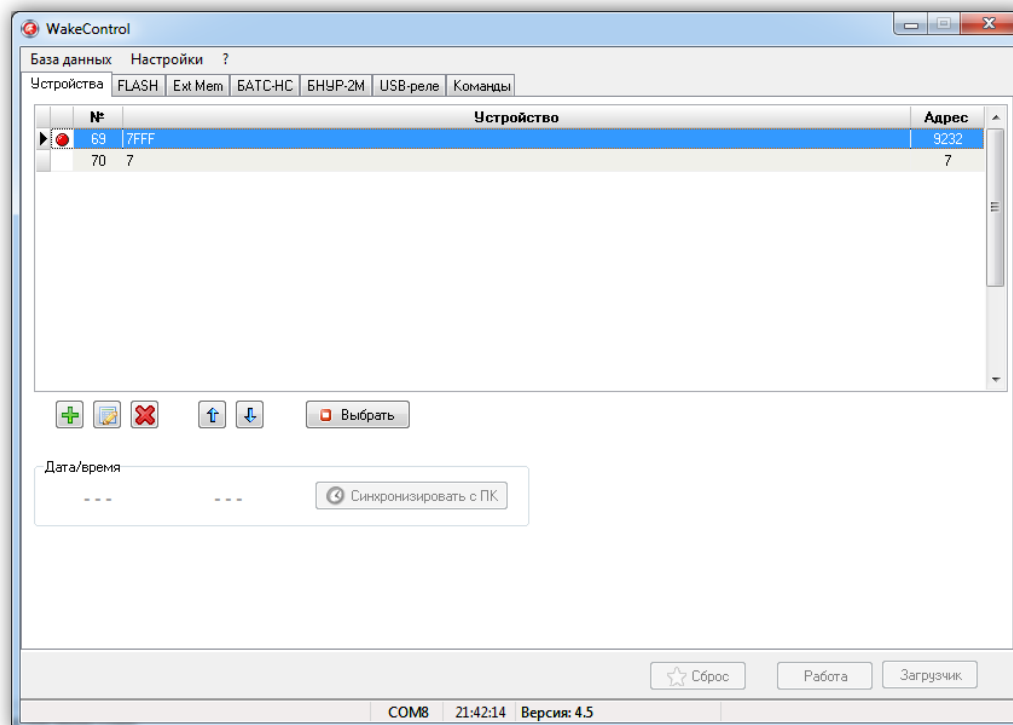
    for (i = 8; i > 0; i--)
        if (CRC16 & 1)
            CRC16 = (CRC16 >> 1) ^ 0x8408;
        else
            CRC16 >>= 1;
}
```

Эту функцию следует вызвать для каждого байта, в результате чего переменная CRC16 будет содержать вычисленное значение контрольной суммы.

## ПРОГРАММА WakeControl

Программа WakeControl предназначена для управления USB-реле по интерфейсу RS-485, изменения адреса устройства и обновления встроенного программного обеспечения.

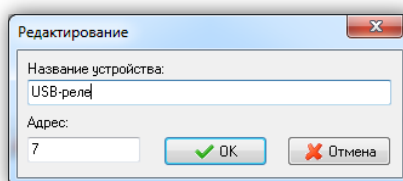
Внешний вид главного окна программы показан ниже:



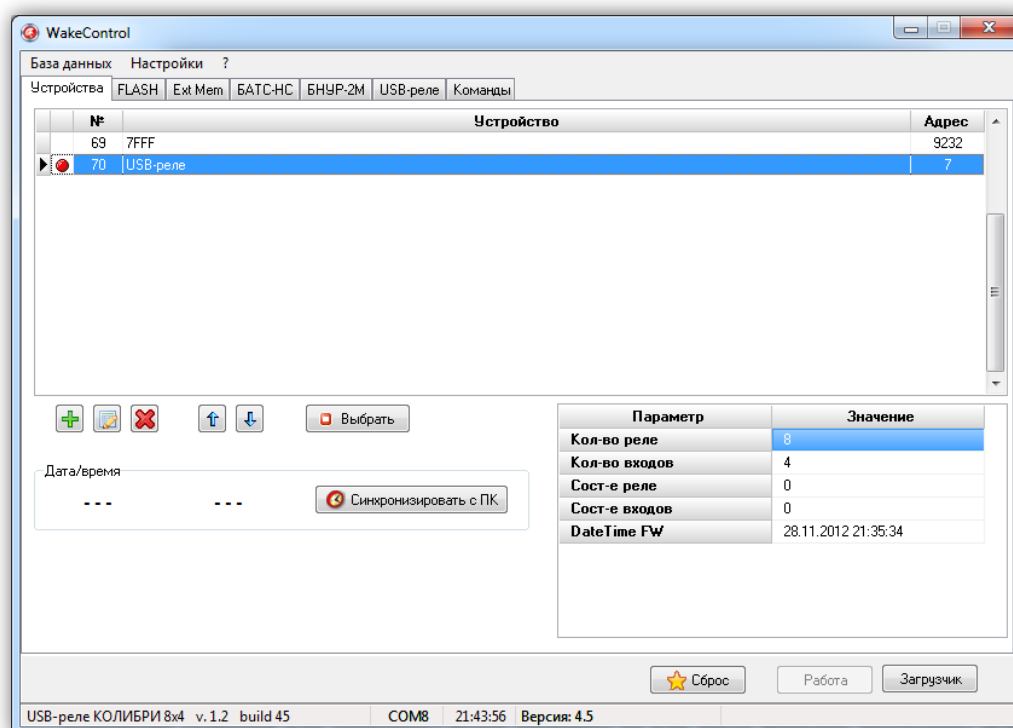
На вкладке «Устройства» отображается список зарегистрированных USB-реле и их адреса. При помощи соответствующих кнопок можно добавлять, изменять и удалять устройства.

Каждое USB-реле имеет собственный адрес, который можно сменить через программу WakeControl. По умолчанию адрес всех USB-реле 0x7FFF (32767).

Новое устройство необходимо зарегистрировать. Для этого следует нажать кнопку «Добавить» и в появившемся окне задать название устройства и его адрес:



Чтобы установить связь необходимо выбрать нужное устройство из списка и нажать кнопку «Выбрать». При успешном установлении связи в строке статуса программы появится название устройства (USB-реле «КОЛИБРИ 8х4») и версия его встроенного ПО:

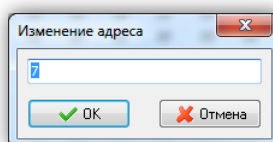


Если вместо названия устройства в строке статуса будет написано «Bootloader», то это означает, что устройство находится в режиме загрузчика. В зависимости от режима работы устройства (рабочий или загрузчик) будут доступны те или иные функции программы WakeControl.

В режиме загрузчика можно только считывать и записывать содержимое FLASH-памяти устройства (для обновления встроенного ПО), а также изменять адрес. В рабочем режиме можно управлять электромагнитными реле и считывать состояние входов.

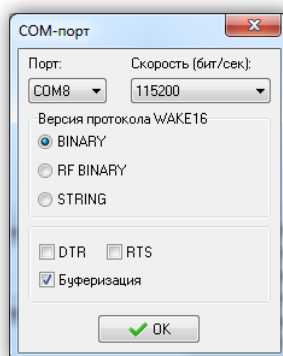
USB-реле можно перевести из рабочего режима в режим загрузчика и обратно при помощи кнопок «Загрузчик» и «Работа». Также можно выполнить общий сброс устройства нажатием кнопки «Сброс».

Чтобы изменить адрес устройства необходимо перевести его в режим загрузчика, а затем в меню «Настройки» выбрать пункт «Сменить адрес...». В появившемся окне необходимо ввести новый адрес и нажать кнопку «ОК»:



Устройство будет отвечать по новому адресу только после выполнения общего сброса нажатием кнопки «Сброс» или отключением и повторной подачей питания.

Если после запуска программы WakeControl и выбора соответствующего устройства связь не установилась, то необходимо выбрать пункт «COM-порт...» меню «Настройка»:

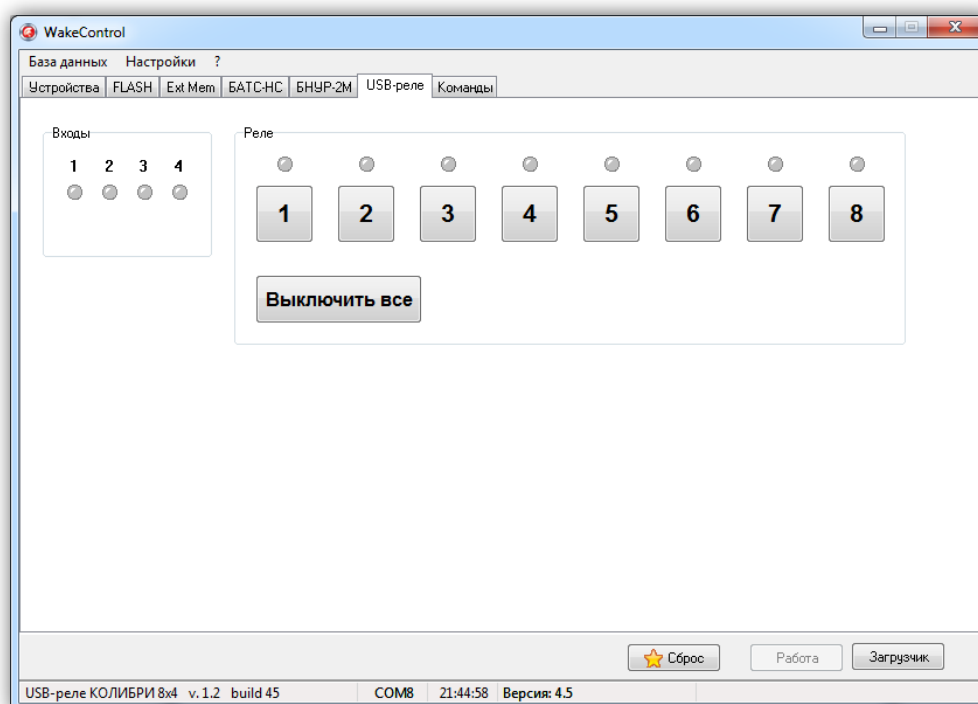


В данном окне следует указать номер COM-порта, к которому подключено устройство, скорость передачи данных задать равной 115200, а в качестве версии протокола WAKE16 выбрать BINARY.

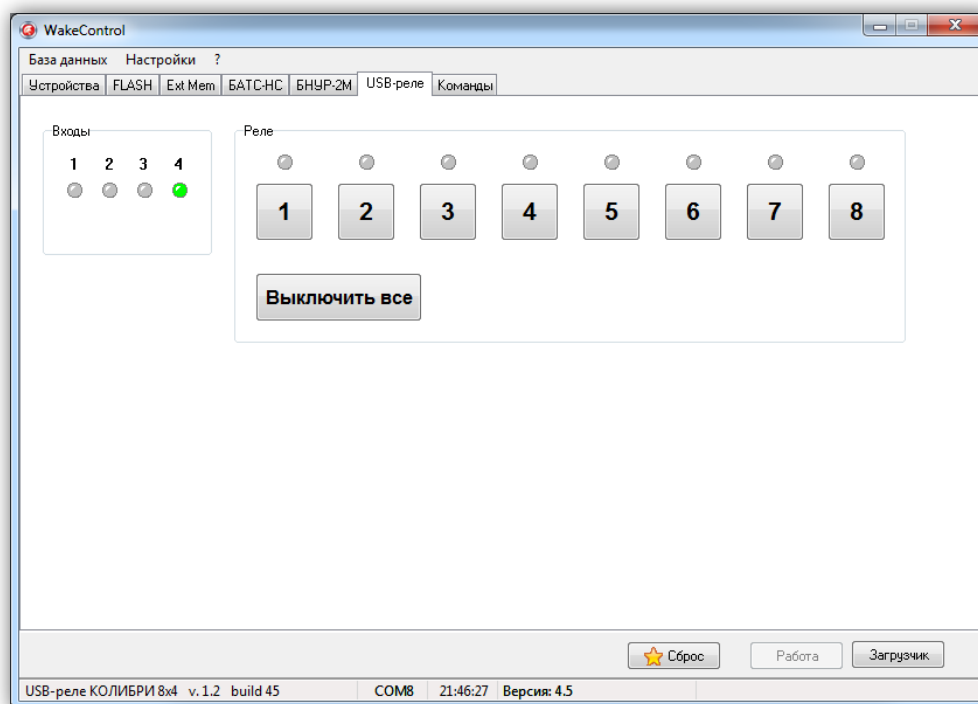










Если после выполнения всех настроек связь с устройством так и не появилась, то необходимо проверить целостность соединительного кабеля и исправность COM-порта персонального компьютера.

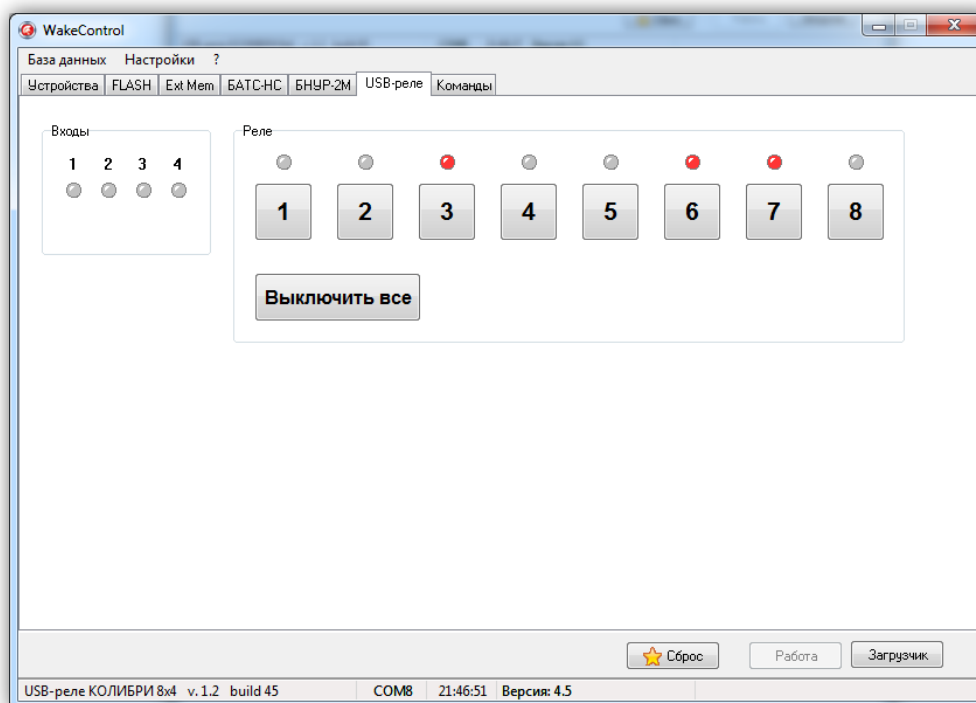
После настройки связи с устройством можно перейти на вкладку «USB-реле» и проверить работоспособность устройства:



Программа WakeControl постоянно опрашивает состояние входов для подключения внешних датчиков и отображает их состояние в поле «Входы». Сработавшему датчику будет соответствовать зелёный цвет индикатора:



Управление электромагнитными реле осуществляется нажатием на соответствующую кнопку (         ). Первое нажатие включит реле, второе – выключит. Включённому состоянию соответствует красный цвет индикатора, расположенного над кнопкой:

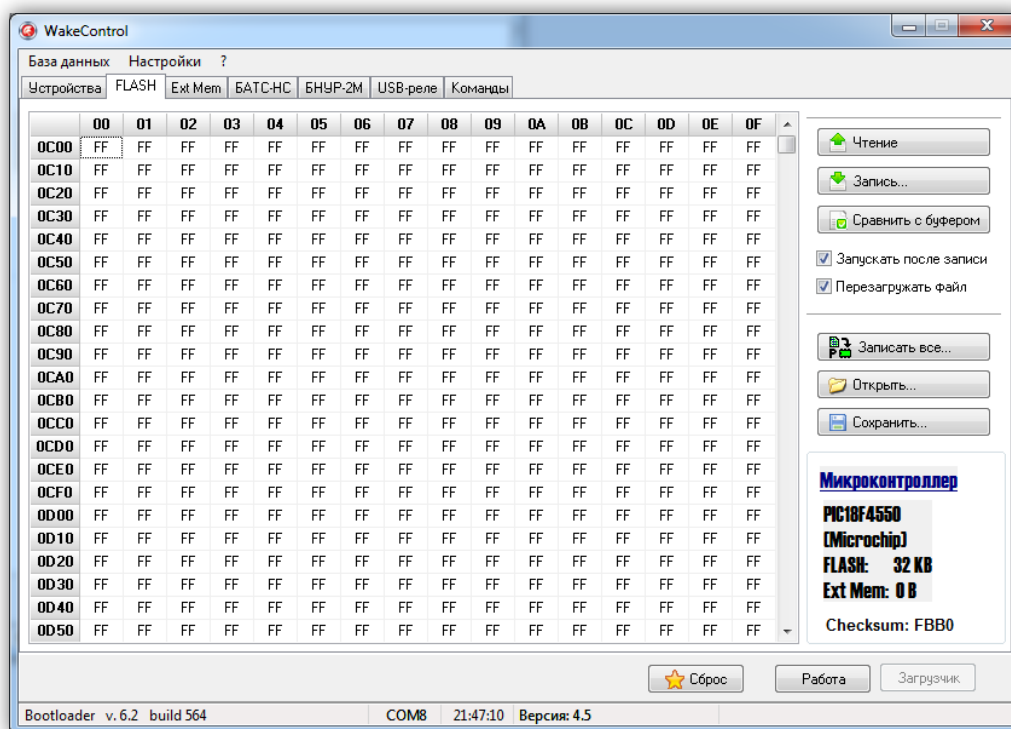


Нажатие на кнопку «Выключить все» приводит к выключению всех электромагнитных реле.

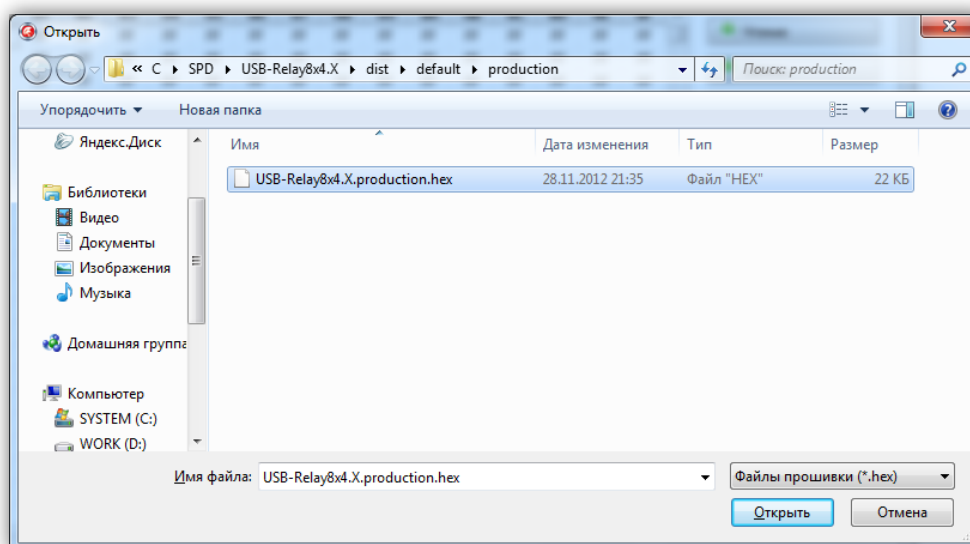
## ОБНОВЛЕНИЕ ВСТРОЕННОГО ПРОГРАММНОГО ОБЕСПЕЧЕНИЯ

Для загрузки нового программного обеспечения в USB-реле необходимо выполнить следующие действия:

- 1) Запустить программу WakeControl и перевести устройство в режим загрузчика.
- 2) Перейти на вкладку FLASH и нажать кнопку «Открыть...»:

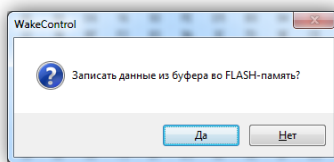


В появившемся окне нужно выбрать соответствующий файл формата Intel HEX:

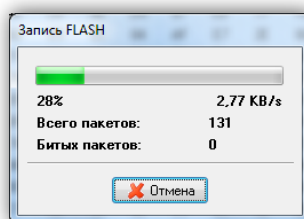


После выбора файла и нажатия кнопки «Открыть» файл будет загружен, а его содержимое в шестнадцатеричном виде будет отображено на панели.

Для записи нового программного обеспечения в устройство необходимо нажать кнопку «Запись...», а затем кнопку «Да» в появившемся диалоговом окне:



При этом начнётся процесс записи:

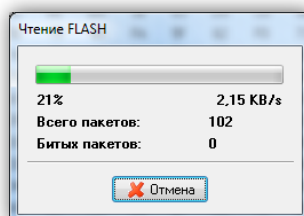


После окончания записи следует нажать кнопку «Работа» для перевода устройства в рабочий режим. Если всё прошло нормально, то в строке статуса программы будет выведена строка с новой версией программного обеспечения.

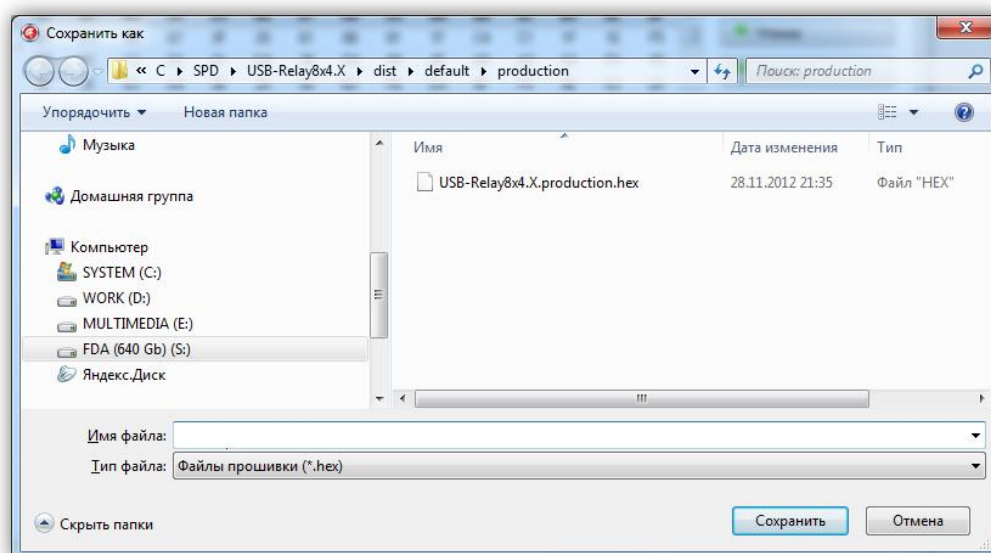


*Перед обновлением встроенного программного обеспечения рекомендуется сохранить предыдущую версию!*

Действия по сохранению текущей версии встроенного программного обеспечения аналогичны действиям при записи, но после перевода устройства в режим загрузчика следует нажать кнопку «Чтение...». При этом начнётся процесс считывания данных из устройства:



После окончания чтения нужно нажать кнопку «Сохранить...» и в появившемся окне задать имя файла:



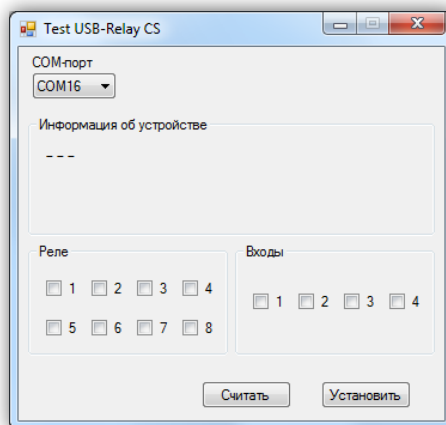
Далее необходимо нажать кнопку «Сохранить», после чего текущая версия ПО будет сохранена в заданном файле.



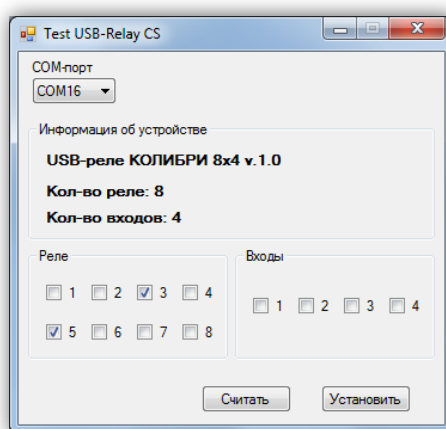
### ПРОГРАММА «Test USB-Relay CS»

Программа Test USB-Relay CS демонстрирует реализацию протокола управления USB-реле по интерфейсу USB. Программа написана на языке C# в среде Visual Studio 2012, представлена в исходных кодах и не имеет никаких ограничений по модификации и распространению.

Внешний вид главного окна программы представлен ниже:



Для начала работы с USB-реле необходимо выбрать COM-порт, к которому оно подключено. После этого следует нажать кнопку «Считать». Если порт был выбран верно и устройство исправно, программа выведет следующую информацию:



В поле «Реле» отмечены те номера электромагнитных реле, которые в данный момент включены. В поле «Входы» отмечены сработавшие в данный момент датчики.

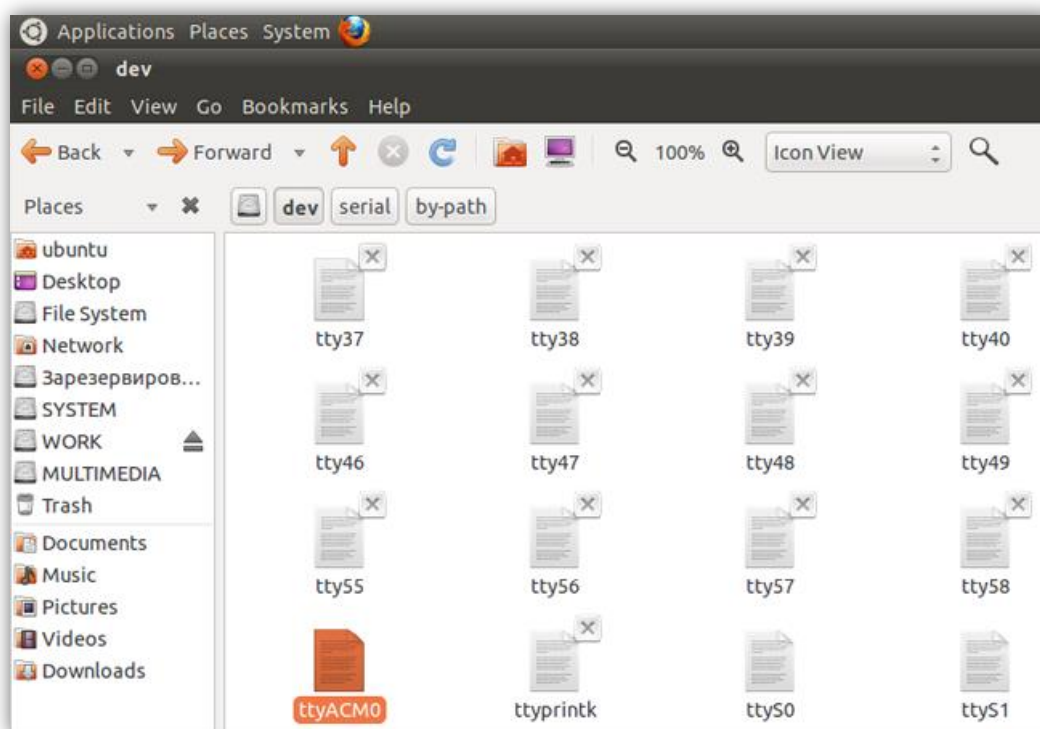
Для изменения состояния нужных реле нужно вручную отметить их, а затем нажать кнопку «Установить». В результате выбранные реле будут включены.

## РАБОТА С USB-УСТРОЙСТВАМИ В ОПЕРАЦИОННОЙ СИСТЕМЕ LINUX

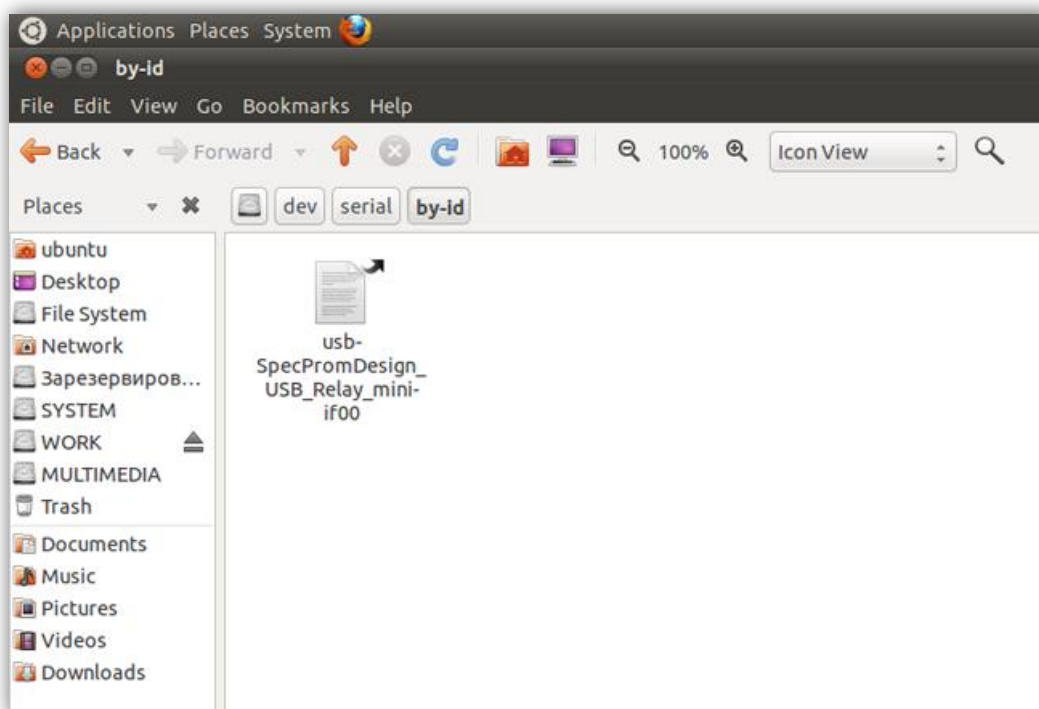
Для работы с USB-устройством в операционной системе Linux не требуется установка каких-либо драйверов и конфигурационных файлов. После подключения устройства к разъёму USB операционная система автоматически его обнаружит и пропишет в системе.

Необходимо только узнать под каким символическим именем USB-устройство существует в системе. По этому имени в дальнейшем необходимо будет обращаться к устройству при написании программ.

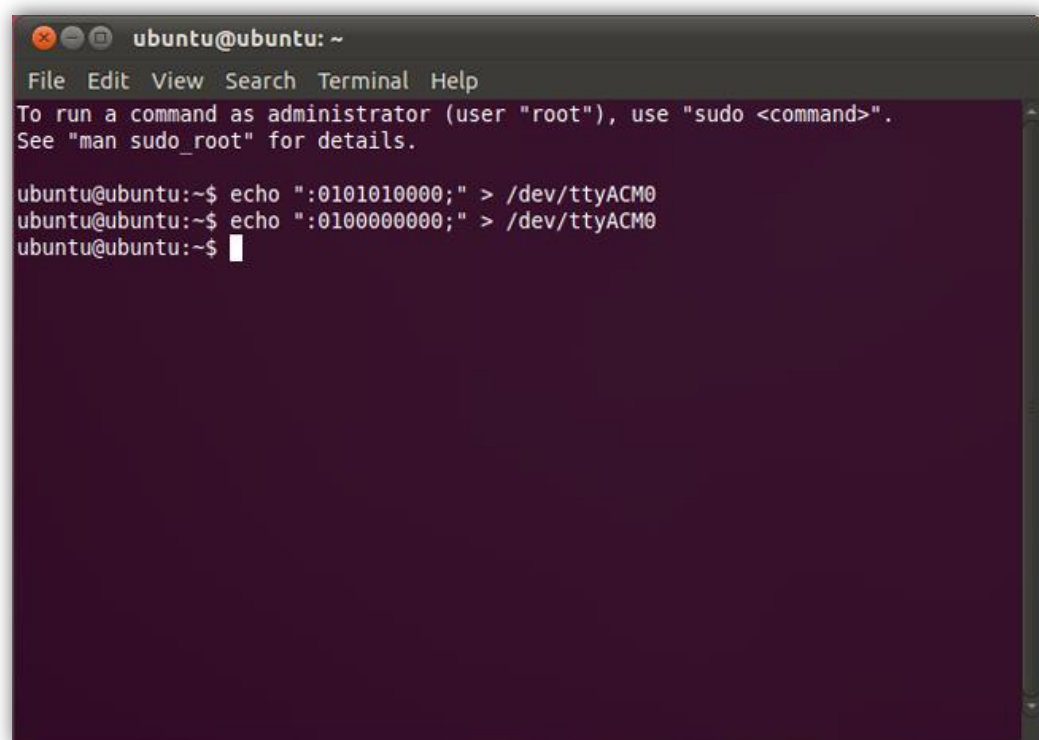
Для этого нужно посмотреть содержимое директории /dev. Наиболее вероятно, что устройству будет присвоено имя ttyACMx, либо ttyUSBx, где x – целое число. Рекомендуется отключить устройство и посмотреть директорию /dev. Затем снова подключить и ещё раз посмотреть директорию. Таким образом можно обнаружить появление нового устройства в списке:



Также можно просмотреть директорию `/dev/serial/by-id`:



После определения символического имени устройства можно проверить его работоспособность. Для этого необходимо запустить *Terminal* и передать данные устройству при помощи команды *echo*:



В данном примере показана работа с USB-реле «КОЛИБРИ». Вначале подаётся команда на включение первого и второго реле, а потом на их выключение.