

# ОЗОН

Руководство  
по эксплуатации

## Логический контроллер

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>ТЕХНИЧЕСКИЕ ХАРАКТЕРИСТИКИ .....</b>	<b>3</b>
<b>УСТРОЙСТВО И ПРИНЦИП РАБОТЫ .....</b>	<b>4</b>
<b>РЕГИСТРАЦИЯ НОВЫХ ПУЛЬТОВ ДИСТАНЦИОННОГО УПРАВЛЕНИЯ.....</b>	<b>8</b>
<b>ПРОТОКОЛ ОБМЕНА ДАННЫМИ ПО ИНТЕРФЕЙСУ RS-485 .....</b>	<b>9</b>
<b>ПРОТОКОЛ ОБМЕНА WAKE16 .....</b>	<b>15</b>
Описание протокола .....	15
Структура пакета .....	15
<b>ПРОГРАММА WakeControl.....</b>	<b>19</b>
<b>ОБНОВЛЕНИЕ ВСТРОЕННОГО ПРОГРАММНОГО ОБЕСПЕЧЕНИЯ .....</b>	<b>25</b>

## ОПИСАНИЕ

Логический контроллер предназначен для управления боллардами (дорожными блокираторами). Он имеет три независимых канала управления исполнительными механизмами.

Поднятие/опускание боллардов осуществляется путём подачи управляющего сигнала на соответствующий вход, либо по командам от внешнего пульта дистанционного управления.

В каждом канале управления предусмотрено два входа для подключения концевых выключателей, по сигналам с которых происходит автоматическая остановка движения болларда в одном из крайних положений.

Контроллер имеет интерфейс RS-485, через который можно выполнять настройки, управлять выходными сигналами, а также осуществлять обновление встроенного программного обеспечения.

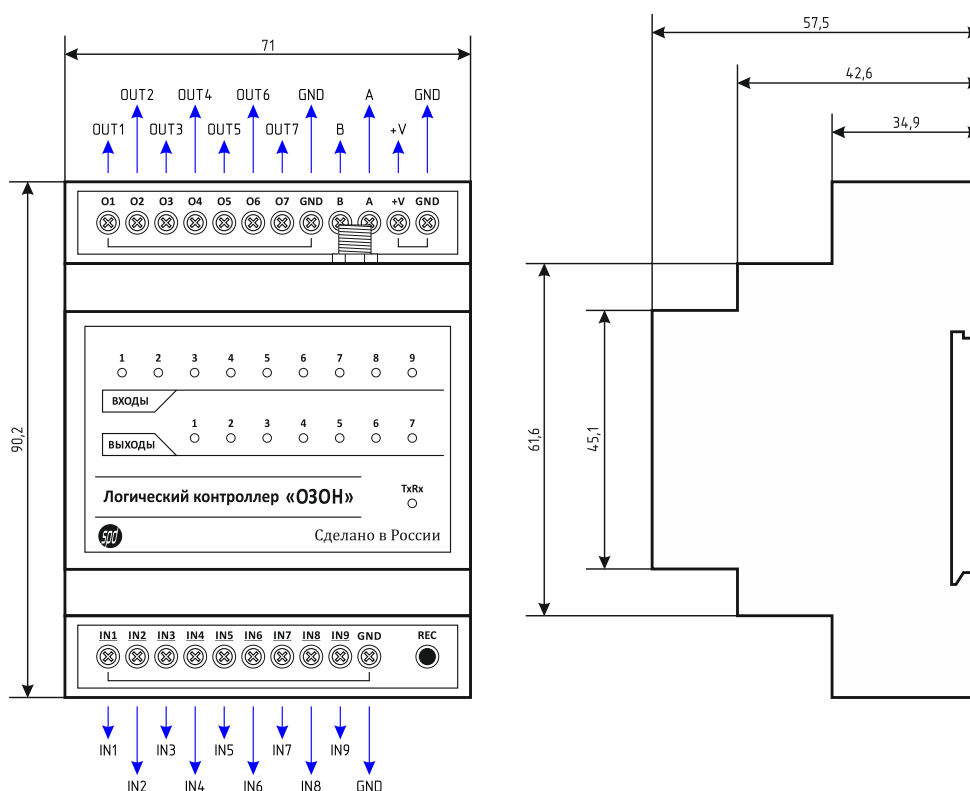


## ТЕХНИЧЕСКИЕ ХАРАКТЕРИСТИКИ

Напряжение питания .....	9 ÷ 40 В
Максимальный потребляемый ток .....	200 мА
Количество входов .....	9
Количество выходов .....	7
Максимальный ток выхода .....	1 А
Частота работы встроенного радиомодуля .....	433,92 МГц
Протокол передачи по радиоканалу .....	Keeloq
Максимальное количество пультов в памяти контроллера .....	7
Скорость обмена данными по интерфейсу RS-485 .....	115200 бит/сек
Протокол обмена данными по интерфейсу RS-485 .....	WAKE16
Степень защиты .....	IP20
Габаритные размеры .....	71 × 90 × 57 мм
Температурный диапазон работы .....	от -40°C до +85°C
Относительная влажность воздуха .....	не более 90% при +35°C

## УСТРОЙСТВО И ПРИНЦИП РАБОТЫ

Логический контроллер выпускается в пластиковом корпусе на DIN-рейку:



Подключение внешних цепей осуществляется при помощи винтовых клеммников.

Назначение контактов клеммников следующее:

**+V, GND** – напряжение питания устройства;

**IN1...IN9** – входы (замыкание на «землю»);

**OUT1...OUT7** – выходные линии;

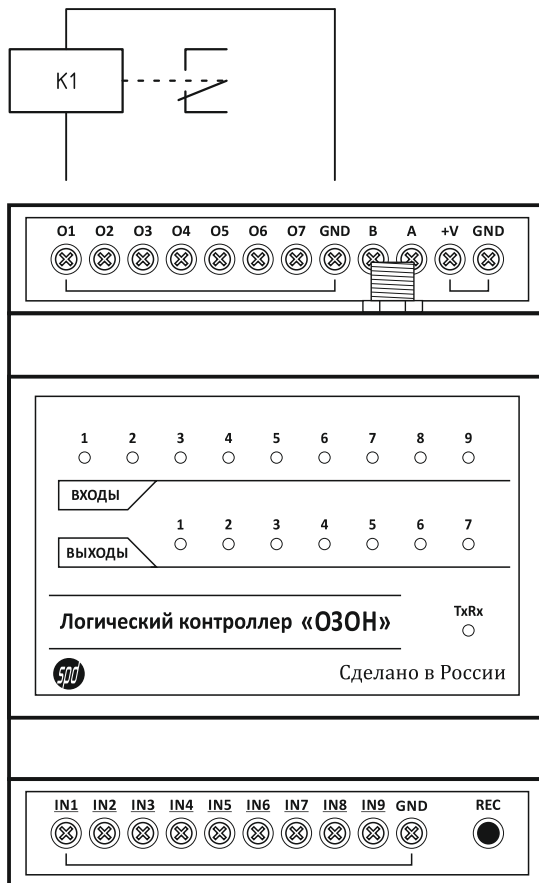
**A, B** – линии A и B интерфейса RS-485.

На лицевой панели расположено 16 светодиодов. Светодиоды с номерами 1...9 группы «Входы» и 1...7 в группе «Выходы» отображают состояния соответствующих входов для подключения внешних контактных датчиков и управляющих линий.

Светодиод «TxRx» индицирует приём данных встроенным радиомодулем – кратковременно вспыхивает при приёме очередной команды. При переводе контроллера в режим загрузчика (см. ниже) для обновления встроенного программного обеспечения этот светодиод постоянно мигает с частотой около 2 Гц.

Возле клемм входов расположена кнопка «REC», которая предназначена для регистрации в контроллере новых пультов дистанционного управления.

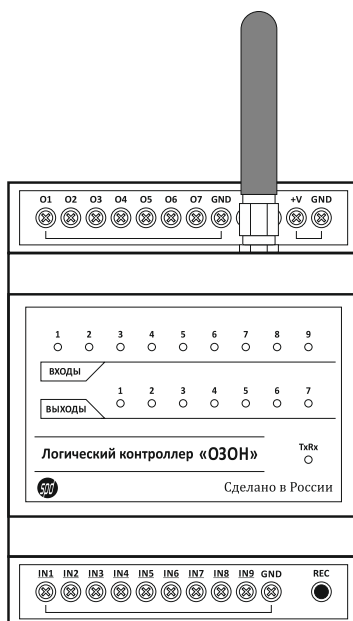
Выходные линии логического контроллера позволяют напрямую коммутировать нагрузку с током потребления до 1 А. В активном состоянии они выдают напряжение, равное напряжению питания контроллера, поэтому нагрузку следует подключать между выходом и «землёй»:



Все линии имеют защиту от короткого замыкания на «землю». После устранения причины замыкания работоспособность выхода автоматически восстанавливается.

Входы IN1...IN9 предназначены для подключения контактных датчиков с выходом «сухой контакт». Активным сигналом для входов является лог. 0 (замыкание входа на «землю» GND).

Контроллер имеет встроенный радиомодуль, работающий на частоте 433,92 МГц. Антенный вход радиомодуля выведен на разъём SMA, расположенный над клеммами А и В:



Для увеличения дальности приёма к данному разъёму можно подключить внешнюю антенну:



Логический контроллер может управлять одновременно тремя группами боллардов. При этом для каждой группы используются свои линии управления и входы для концевых выключателей.

Распределение выводов контроллера по группам приведено ниже:

***Первая группа***

OUT1 – ПОДЪЁМ

OUT2 – ОПУСКАНИЕ

IN1 – вход концевого выключателя на ПОДЪЁМ

IN2 – вход концевого выключателя на ОПУСКАНИЕ

***Вторая группа***

OUT3 – ПОДЪЁМ

OUT4 – ОПУСКАНИЕ

IN3 – вход концевого выключателя на ПОДЪЁМ

IN4 – вход концевого выключателя на ОПУСКАНИЕ

***Третья группа***

OUT5 – ПОДЪЁМ

OUT6 – ОПУСКАНИЕ

IN5 – вход концевого выключателя на ПОДЪЁМ

IN6 – вход концевого выключателя на ОПУСКАНИЕ

Входы IN7 и IN8 используются для подачи управляющих сигналов ПОДЪЁМ и ОПУСКАНИЕ. Активным сигналом для этих входов является импульс продолжительностью не менее 50 мс.

При одновременной подачи обоих сигналов контроллер переходит в режим СТОП, в котором все управляющие сигналы отключаются и движение боллардов прекращается.

После включения режима ПОДЪЁМА или ОПУСКАНИЯ управляющие сигналы будут удерживаться в активном состоянии до срабатывания соответствующего концевого выключателя. При этом в настройках контроллера задаётся максимально возможная продолжительность выдачи управляющих сигналов (тайм-аут), после превышения которой сигналы деактивированы. Это сделано для исключения поломок исполнительных устройств при неисправности концевых выключателей.

По умолчанию значение тайм-аута составляет 10 сек.

Если тайм-аут в какой-либо группе оказался превышен, то светодиоды, соответствующие этой группе, будут моргать с частотой 1 Гц.

Для управления ПОДЪЁМОМ и ОПУСКАНИЕМ можно также использовать пульт дистанционного управления на чипе HCS301:



Нажатия кнопок на пульте полностью имитируют подачу управляющих сигналов на входы IN7 (ПОДЪЁМ) и IN8 (ОПУСКАНИЕ).

## РЕГИСТРАЦИЯ НОВЫХ ПУЛЬТОВ ДИСТАНЦИОННОГО УПРАВЛЕНИЯ

Контроллер поддерживает одновременную работу с семью пультами дистанционного управления.

Для регистрации в контроллере нового пульта необходимо выполнить следующие действия:

1. Перевести контроллер в режим регистрации пультов кратковременным нажатием кнопки «REC».

После этого светодиод «TxRx» будет светиться постоянно, а светодиоды из группы «Выходы» будут индицировать количество записанных в память контроллера пультов.

Выход из этого режима осуществляется автоматически если никакие действия с контроллером не проводились в течение 10 сек.

2. Для регистрации нового пульта следует нажать на нём любую кнопку. Контроллер автоматически считает идентификатор пульта и запишет его в свою встроенную энергонезависимую память.

Если добавление произошло успешно, количество светящихся светодиодов в группе «Выходы» увеличится на один.

В случае, если такой пульт уже зарегистрирован в контроллере, то один из светодиодов в группе «Выходы», соответствующий порядковому номеру пульта в памяти контроллера, будет мигать в течение 1 сек.

3. Для удаления всех номеров пультов из памяти контроллера следует нажать и удерживать кнопку «REC» в течение 3 сек. После этого память будет очищена, а все светодиоды из группы «Выходы» погаснут.



## ПРОТОКОЛ ОБМЕНА ДАННЫМИ ПО ИНТЕРФЕЙСУ RS-485

Настройка контроллера может осуществляться через интерфейс RS-485 по протоколу WAKE16 на скорости 115200 бит/сек. Для управления используются 13 команд. Если контроллер примет пакет с неверным форматом команды, то он передаст ответ с командой 0x22 и пустыми данными.

По умолчанию адрес контроллера имеет значение 0x7FFF (32767). Если он один на линии RS-485, то может использоваться широковещательный адрес 0x0000 (0). Ответ от контроллера всегда содержит нулевое значение поля адреса, что позволяет всегда отличать его от остальных принимаемых пакетов данных.

Ниже приводится описание команд, реализованных в контроллере. 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.

---

***cmReadAddr*** – чтение адреса контроллера.

Описание: возвращает текущий адрес контроллера.

Команда: 0x0A

Данные: нет

Ответ:

Команда: 0x33

Данные: <addr<sub>16</sub>> – текущий адрес контроллера

---

***cmReset*** – сброс контроллера.

Описание: выполняет аппаратный сброс контроллера.

Команда: 0x08

Данные: нет

Ответ:

Команда: 0x33

Данные: нет

---

***cmRunBoot*** – переключение в режим загрузчика.

Описание: переключает контроллер в режим загрузчика для обновления встроенного программного обеспечения.

Команда: 0x02

Данные: нет

Ответ:Команда: 0x33Данные: нет

---

**cmSetOuts** – включение/отключение выходных линий.

Описание: задаёт индивидуальное состояние каждой выходной линии.

Команда: 0x61Данные: <outs\_mask >

Значение <outs\_mask> содержит маску включения выходных линий. Установленный бит в 0-ом разряде соответствует активному состоянию линии OUT1, в 1-ом разряде – OUT2, во 2-ом разряде – OUT3 и т.д.

Ответ:Команда: 0x33Данные: нет

---

**cmGetOutsInputs** – текущее состояние входов и выходных линий.

Описание: считывает текущее состояние входов и выходных линий.

Команда: 0x62Данные: нетОтвет:Команда: 0x33Данные: <inputs\_mask<sub>16</sub>>, <outs\_mask>

Значение <inputs\_mask> содержит маску входов. Установленный бит в 0-ом разряде соответствует активному состоянию входа IN1, в 1-ом разряде – IN2, во 2-ом разряде – IN3 и т.д.

Значение <outs\_mask> содержит маску выходных линий. Установленный бит в 0-ом разряде соответствует активному состоянию линии OUT1, в 1-ом разряде – OUT2, во 2-ом разряде – OUT3 и т.д.

---

**cmUp** – команда ПОДЪЁМ.

Описание: включает режим ПОДЪЁМА боллардов.

Код команды: 0x63Данные: нетОтвет:

Команда: 0x33

Данные: нет

---

**cmDown** – команда ОПУСКАНИЕ.

Описание: включает режим ОПУСКАНИЯ боллардов.

Код команды: 0x64

Данные: нет

Ответ:

Команда: 0x33

Данные: нет

---

**cmStop** – команда СТОП.

Описание: переводит контроллер в режим СТОП и отключает все выходные линии.

Код команды: 0x65

Данные: нет

Ответ:

Команда: 0x33

Данные: нет

---

**cmReadPults** – чтение информации о зарегистрированных пультах.

Описание: считывает информацию о количестве и идентификаторах зарегистрированных пультов.

Команда: 0x6A

Данные: нет

Ответ:

Команда: 0x33

Данные:

<cnt> – количество зарегистрированных пультов (1...7)

<pult\_1<sub>32</sub>>...<pult\_cnt<sub>32</sub>> – идентификаторы пультов

---

**cmWritePults** – запись информации о зарегистрированных пультах.

Описание: сохраняет в контроллере список пультов.

Команда: 0x6B

Данные:

<cnt> – количество пультов (1...7)

<pult\_1<sub>32</sub>>...<pult\_cnt<sub>32</sub>> – идентификаторы пультов

Ответ:Команда: 0x33Данные: нет**cmReadTimeout** – чтение тайм-аута.Описание: возвращает тайм-аут контроллера.Команда: 0x6CДанные: нетОтвет:Команда: 0x33Данные: <timeout<sub>16</sub>> – тайм-аут контроллера**cmWriteTimeout** – запись тайм-аута.Описание: записывает тайм-аут в контроллер.Команда: 0x6DДанные: <timeout<sub>16</sub>> – тайм-аут контроллераОтвет:Команда: 0x33Данные: нет**cmGetInfo** – получение информации об устройстве.Описание: считывает информацию о названии устройства, количестве входов, выходов и т.п..Команда: 0x71Данные: нетОтвет:Команда: 0x33Данные:

&lt;mode&gt; – код режима работы (0x11 – основной режим, 0x10 – режим загрузчика)

&lt;version&gt; – 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

0x0C – сигнатура микроконтроллера (PIC18F46K22)

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

0x04 – сигнатура блока данных (целое число)

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

0x02 – размер данных

<outs\_cnt<sub>16</sub>> – количество выходов

0x04 – сигнатура блока данных (целое число)

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

0x02 – размер данных

<inputs\_cnt<sub>16</sub>> – количество входов

0x04 – сигнатура блока данных (целое число)

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

0x02 – размер данных

<outs\_state<sub>16</sub>> – текущее состояние выходов

0x04 – сигнатура блока данных (целое число)

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

0x02 – размер данных

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

0x01 – сигнатура блока данных (строка)

<str<sub>0</sub>>, <str<sub>1</sub>> ... <0x00> – название блока «Напряжение (В)»

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

<str<sub>0</sub>> ... <str<sub>size-1</sub>>, <0x00> – текстовая строка с текущим напряжением питания

0x07 – сигнатура блока данных (32-битное 16-ричное число)

<str<sub>0</sub>>, <str<sub>1</sub>> ... <0x00> – название блока «Pult SN»

0x04 – размер данных

<pult\_sn<sub>32</sub>> – идентификатор пульта

0x04 – сигнатура блока данных (целое число)

<str<sub>0</sub>>, <str<sub>1</sub>> ... <0x00> – название блока «Pult Button»

0x02 – размер данных

<buttons\_state<sub>16</sub>> – состояние кнопок пульта (0-й бит – кнопка «ПОДНЯТЬ»,

1-й бит – кнопка «ОПУСТИТЬ»

0x06 – сигнатура блока данных (16-битное 16-ричное число)

<str<sub>0</sub>>, <str<sub>1</sub>> ... <0x00> – название блока «Privod State»

0x02 – размер данных

<privods\_state<sub>16</sub>> – состояние групп управления (биты 0...3 – первая группа, биты 4...7 – вторая группа, биты 8...11 – третья группа).

Коды состояний следующие:

- 0 – действие выполнено,
- 1 – идёт ПОДЪЁМ,
- 2 – идёт ОПУСКАНИЕ,
- 3 – нарушение тайм-аута,
- 4 – состояние СТОП

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

## 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;

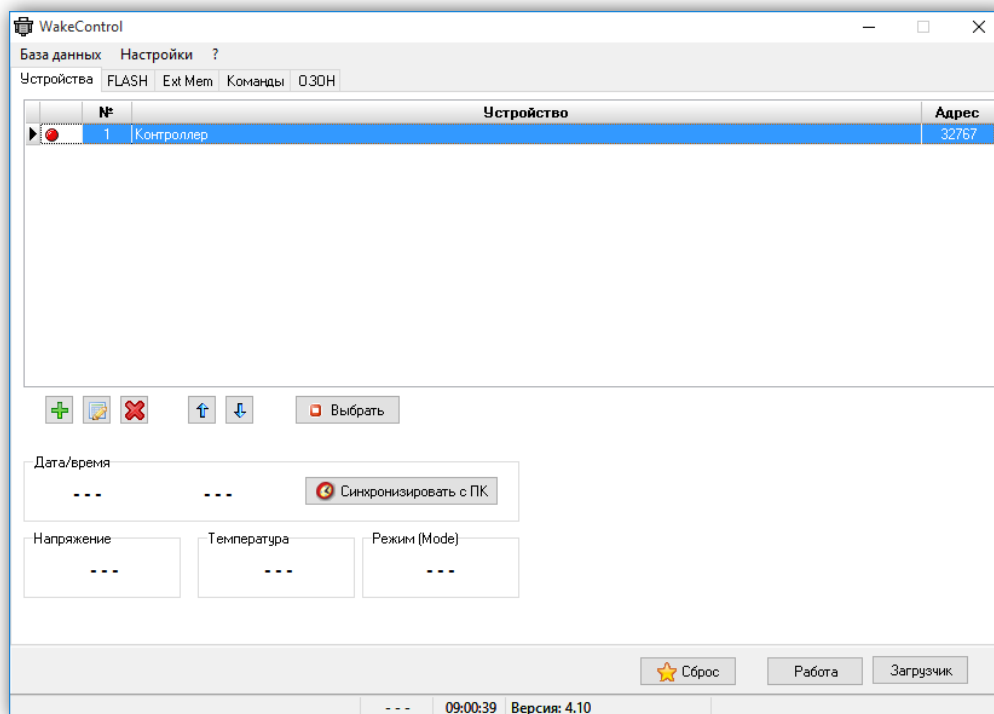
    for (i = 8; i > 0; i--)
        if (CRC16 & 1)
            CRC16 = (CRC16 >> 1) ^ 0x8408;
        else
            CRC16 >>= 1;
}
```

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 контроллером по интерфейсу RS-485, изменения адреса устройства и обновления встроенного программного обеспечения.

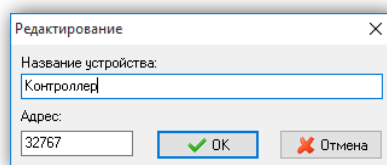
Внешний вид главного окна программы показан ниже:



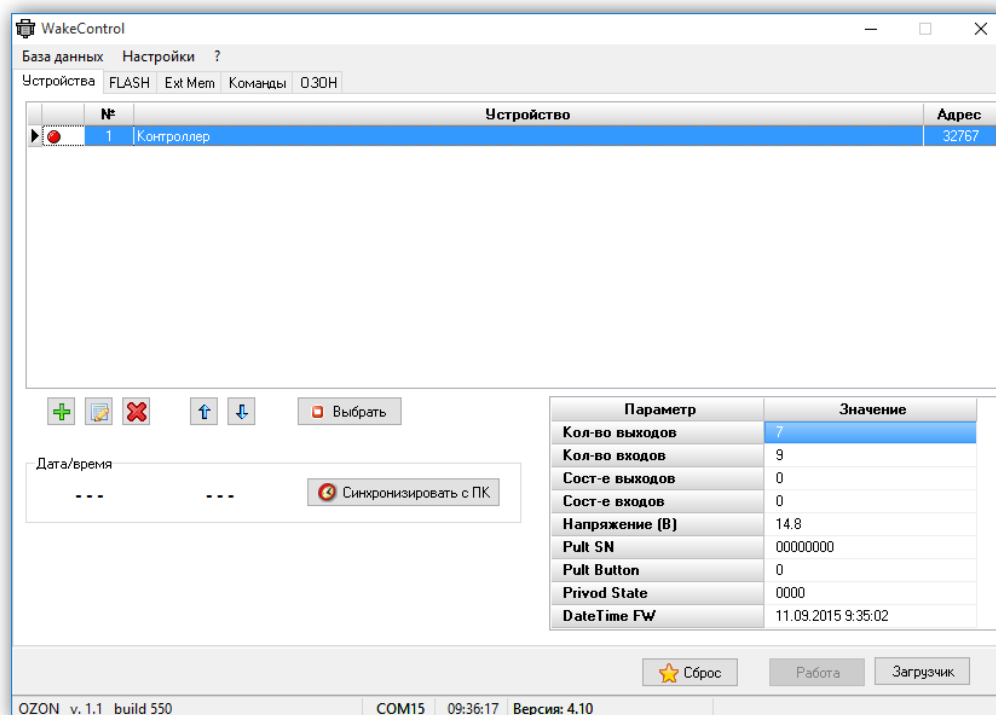
На вкладке «Устройства» отображается список зарегистрированных контроллеров и их адреса. При помощи соответствующих кнопок можно добавлять, изменять и удалять устройства.

Каждый контроллер имеет собственный адрес, который можно сменить через программу WakeControl. По умолчанию адрес всех контроллеров 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 (OZON) и версия его встроенного ПО:

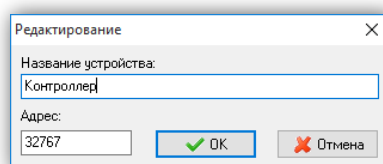


Если вместо названия устройства в строке статуса будет написано «Bootloader», то это означает, что устройство находится в режиме загрузчика. В зависимости от режима работы устройства (рабочий или загрузчик) будут доступны те или иные функции программы WakeControl.

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.

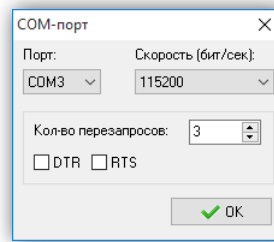
Контроллер можно перевести из рабочего режима в режим загрузчика и обратно при помощи кнопок «Загрузчик» и «Работа». Также можно выполнить общий сброс устройства нажатием кнопки «Сброс».

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»:



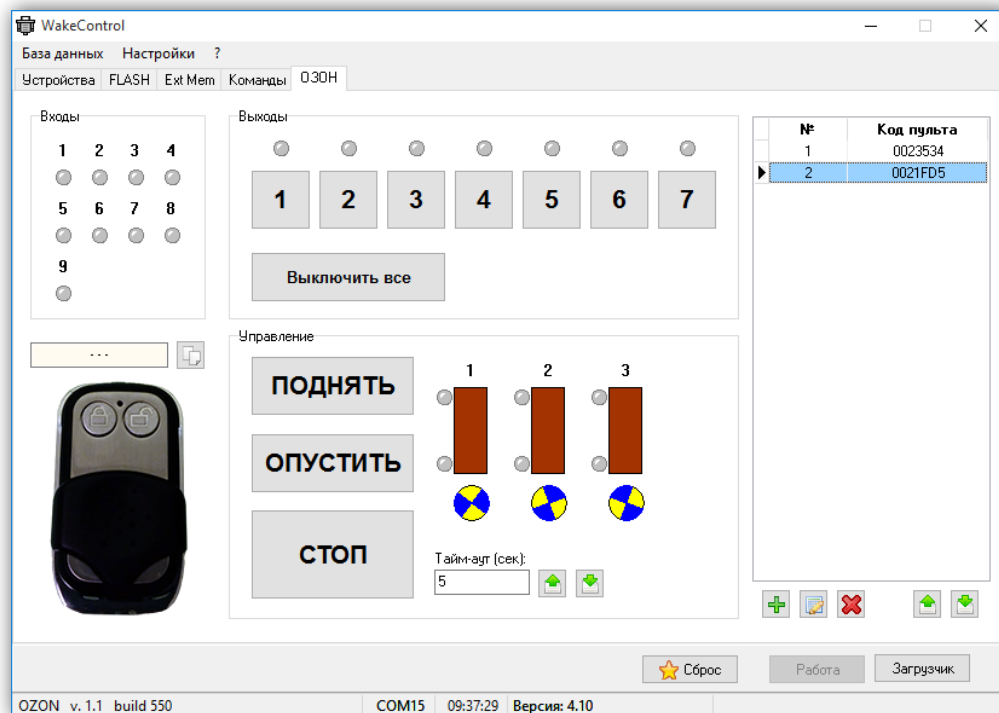
Устройство будет отвечать по новому адресу только после выполнения общего сброса нажатием кнопки «Сброс» или отключением и повторной подачей питания.

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

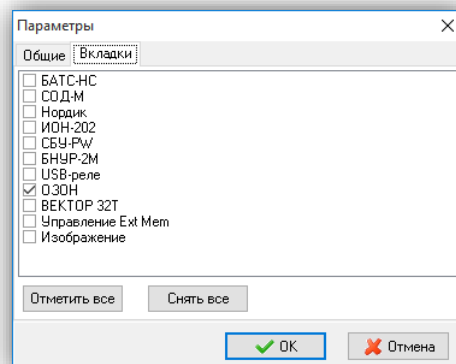


В данном окне следует указать номер СОМ-порта, к которому подключено устройство, а скорость передачи данных задать равной 115200.

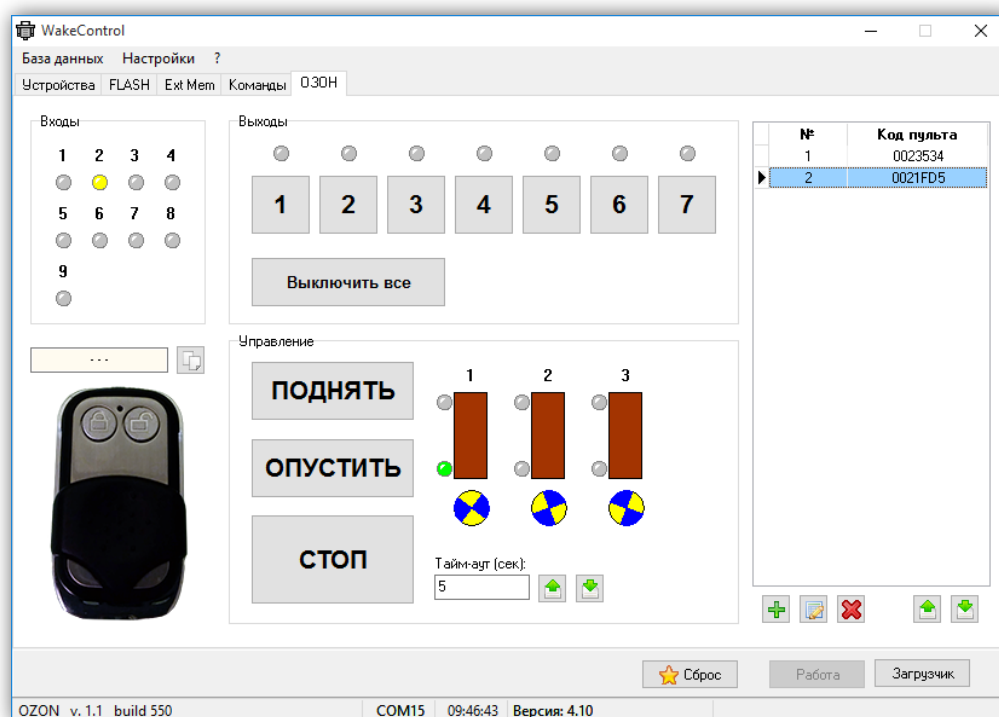
После настройки связи с устройством можно перейти на вкладку «ОЗОН» и проверить работоспособность устройства:



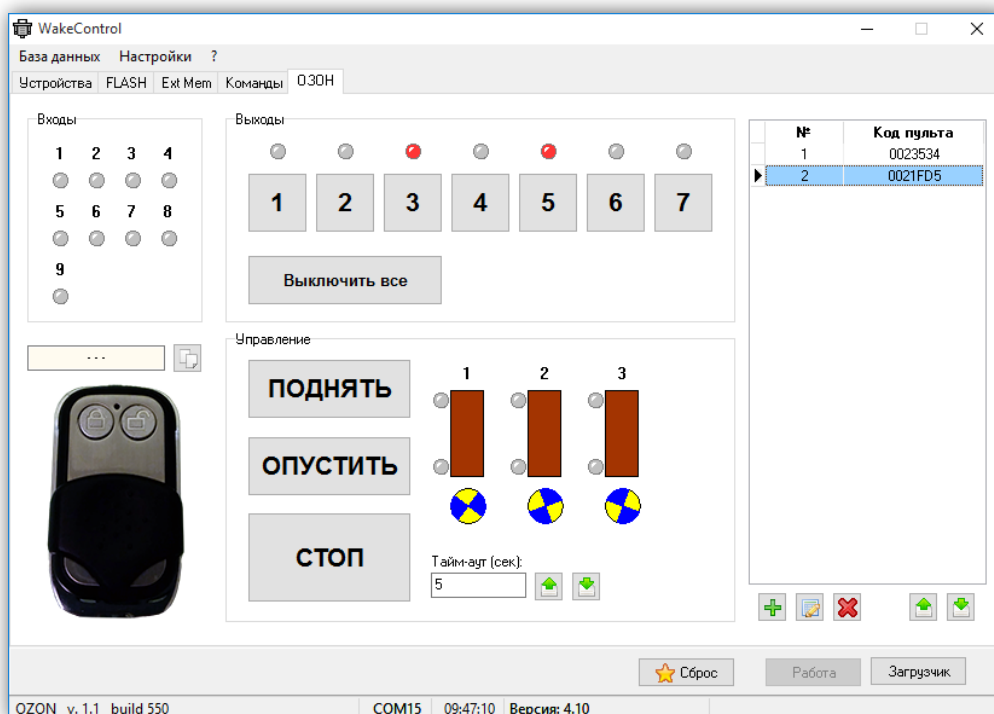
Если вкладка «ОЗОН» не отображается, то следует войти в меню «Настройки» → «Параметры» и выбрать необходимые вкладки для отображения:



Программа WakeControl постоянно опрашивает состояние входов и отображает их состояние в поле «Входы». Активному состоянию входа будет соответствовать жёлтый цвет индикатора:



Управление выходами осуществляется нажатием на соответствующую кнопку ( **1 2 3 4 5 6 7** ). Первое нажатие задаёт активное состояние, второе – выключенное. Активному состоянию соответствует красный цвет индикатора, расположенного над кнопкой:

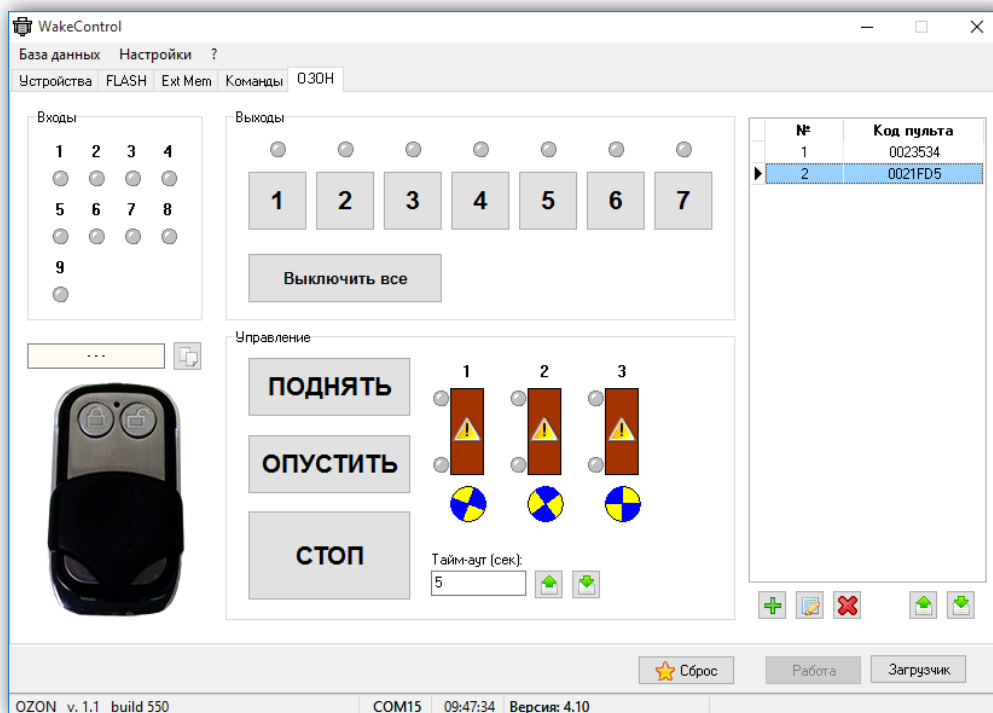




Нажатие на кнопку «Выключить все» приводит к выключению всех электромагнитных реле.

При помощи кнопок «ПОДЪЁМ», «ОПУСКАНИЕ» и «СТОП» можно послать контроллеру соответствующую команду. При этом в процессе выполнения команд ПОДЪЁМ и ОПУСКАНИЕ

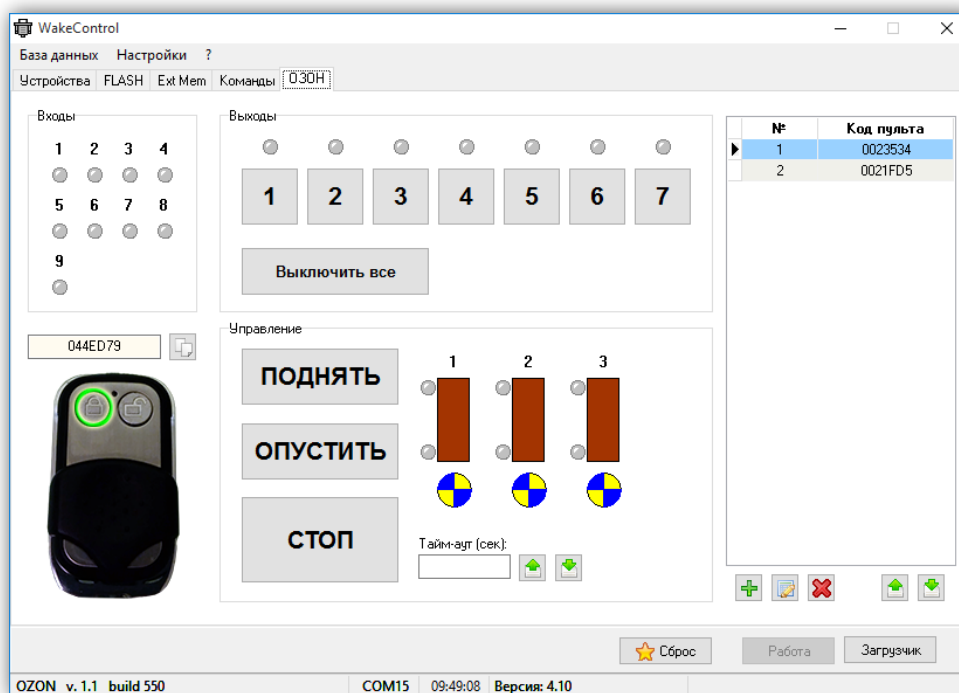
под каждой из условно изображённых групп управления будет вращаться значок насоса.

В случае превышения тайм-аута выполнения команды на соответствующей группе будет выведен значок предупреждения:





Само значение тайм-аута можно считать из контроллера нажатием кнопки (  ), затем отредактировать и записать обратно нажатием кнопки (  ).

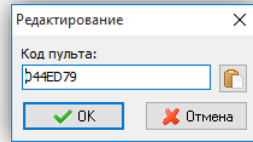
С левой стороны окна находится условное изображение пульта управления. При нажатии кнопок на реальном пульте соответствующие кнопки на изображении подсвечиваются:



В поле сверху изображения выводится код пульта. Нажатием кнопки справа от поля

ввода (  ) можно скопировать этот код в буфер обмена.

В списке с правой стороны окна выводятся коды зарегистрированных пультов управления. Для обновления списка следует нажать кнопку (  ). Список можно редактировать вручную:



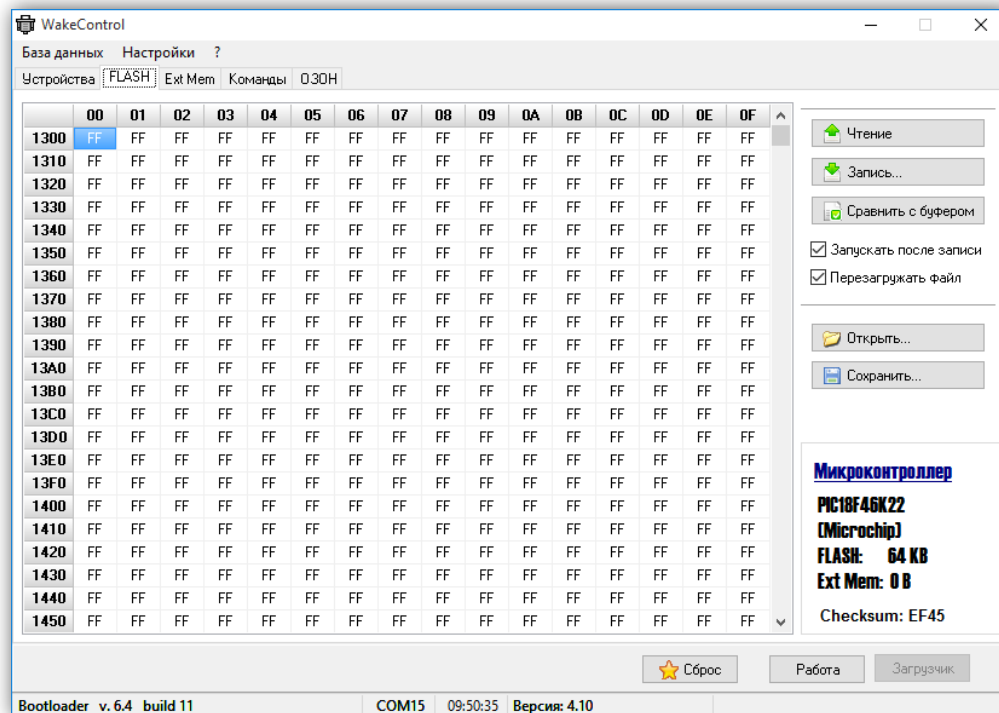
Таким образом можно добавлять новые пульты прямо с компьютера.



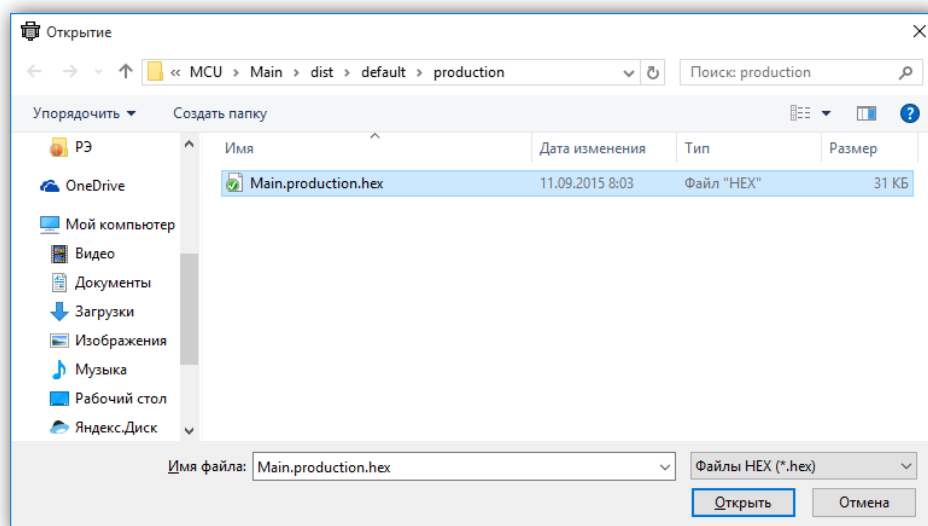
## ОБНОВЛЕНИЕ ВСТРОЕННОГО ПРОГРАММНОГО ОБЕСПЕЧЕНИЯ

Для загрузки нового программного обеспечения в контроллер необходимо выполнить следующие действия:

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

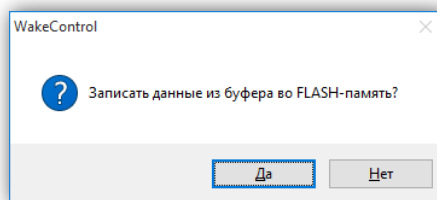


В появившемся окне нужно выбрать соответствующий файл формата Intel HEX:

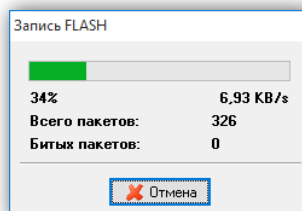


После выбора файла и нажатия кнопки «Открыть» файл будет загружен, а его содержимое в шестнадцатеричном виде будет отображено на панели.

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



При этом начнётся процесс записи:

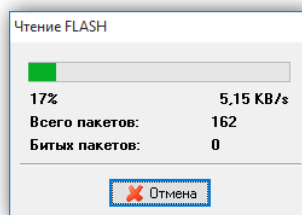


После окончания записи следует нажать кнопку «Работа» для перевода устройства в рабочий режим. Если всё прошло нормально, то в строке статуса программы будет выведена строка с новой версией программного обеспечения.

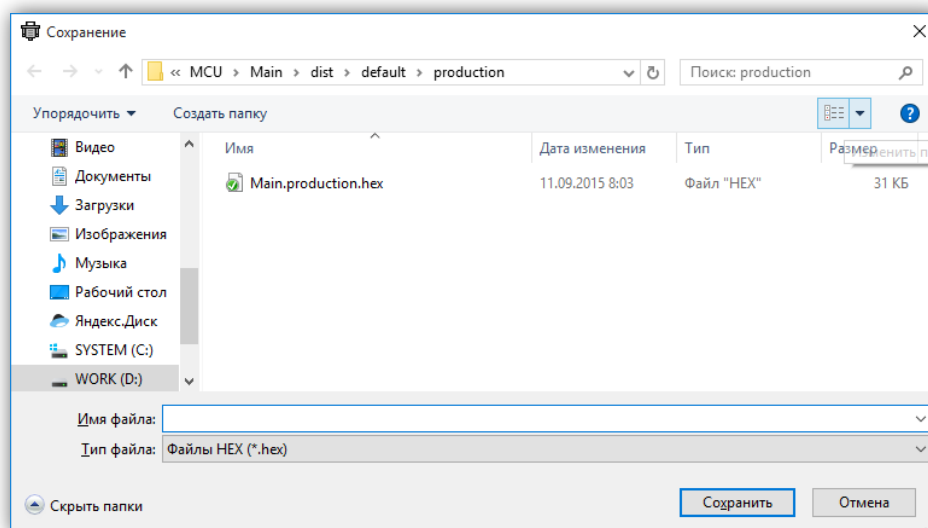


*Перед обновлением встроенного программного обеспечения рекомендуется сохранить предыдущую версию!*

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.