

Контроллер соединяется с сервером по протоколу HTTP(s) и методом POST отправляет имеющиеся у него данные. Сервер в ответ на запрос присылает управляющие сообщения для контроллера.

Если сервер поддерживает ONLINE проверку доступа, то при поднесении карты делается запрос на сервер для проверки разрешения на проход. При этом все остальные функции (события, запись карт итд) работают.

При невозможности ONLINE проверки контроллер (сервер недоступен или возвращает ошибку) переходит в OFFLINE режим и работает с картами, записанными в его память. Периодически контроллер проверяет доступность сервера

## 1. Сообщения отправляемые контроллером

Все посылки имеют следующий вид:

```
{
  "type": "Z5RWEB",
  "sn": 50001,
  "messages": [
    {
      "id": 10,
      ...
    },
    {
      "id": 20,
      ...
    },
    ...
    {
      "id": N,
      ...
    }
  ]
}
```

type - тип контроллера

sn - серийный номер контроллера

messages - массив сообщений от контроллера

id - уникальный идентификатор сообщения

### 1.1 POWER\_ON

Посылается при первом соединении после питания контроллера и продолжает посылаться до тех пор, пока сервер не пришлет [SET\\_ACTIVE](#)

Запрос:

```
{
  "id": 123456789,
  "operation": "power_on",
  "fw": "1.0.1",
  "conn_fw": "2.0.2",
  "active": 0,
  "mode": 0,
  "controller_ip": "192.168.0.222"
}
```

operation - название операции

fw - версия прошивки контроллера

conn\_fw - версия прошивки модуля связи  
active - признак активированности контроллера  
mode - режим работы контроллера (смотри SET\_MODE)  
controller\_ip - IP адрес контроллера в локальной сети

**Ответ:**

Без ответа (смотри SET\_ACTIVE)

## 1.2 CHECK\_ACCESS

Посылается контроллером в режиме ONLINE проверки доступа при поднесении карты к считывателю

**Запрос:**

```
{
  "id": 123456789,
  "operation": "check_access",
  "card": "00B5009EC1A8",
  "reader": 1
}
```

card - номер карты в шестнадцатеричном виде (см. ПРИЛОЖЕНИЕ 2)

reader - считыватель, к которому приложена карта. 1- вход, 2 - выход.

**Ответ:**

```
{
  "id":123456789,
  "operation": "check_access",
  "granted":1
}
```

granted - 1 - проход разрешен, 0 - запрещен

## 1.3 PING

Посылается периодически при отсутствии событий. Интервал передачи настраивается в WEB интерфейсе.

**Запрос:**

```
{
  "id": 123456789,
  "operation": "ping",
  "active": 1,
  "mode": 0
}
```

active, mode - смотри POWER\_ON

**Ответ:**

Сообщения для контроллера.

## 1.4 EVENTS

Посылается при появлении новых событий в контроллере. При ответе "success" = N, N событий считаются обработанными. При ответе "success" = 0 или отсутствии ответа, повторяется отправка.

**Запрос:**

```
{
  "id": 123456789,
  "operation": "events",
  "events": [
```

```
{
  "event": 4,
  "card": "00B5009EC1A8",
  "time": "2015-06-25 16:36:01",
  "flag": 0
},
{
  "event": 16,
  "card": "00BA00FE32A2",
  "time": "2015-06-25 16:36:02",
  "flag": 0
}
]
```

events - массив событий

event - тип события (смотри ПРИЛОЖЕНИЕ 1)

card - номер карты в шестнадцатеричном виде (для событий с картой) (см. ПРИЛОЖЕНИЕ 2)

time - время события

flag - флаги события (для событий с флагами)

**Ответ:**

```
{
  "id":123456789,
  "operation": "events",
  "events_success":2
}
```

events\_success - количество успешно принятых событий

## 2. Сообщения отправляемые сервером

Посылки от сервера имеют следующий вид:

```
{
  "date": "2017-07-25 10:20:30",
  "interval": 10,
  "messages": [
    {
      "id": 10,
      ...
    },
    {
      "id": 20,
      ...
    },
    ...
    {
      "id": N,
      ...
    }
  ]
}
```

date - текущее время на сервере. время в контроллере устанавливается равным присланному времени

interval - период между соединениями установленный сервером

## 2.1 SET\_ACTIVE

Активирует / деактивирует работу контроллера с сервером. Не активированный контроллер не передаёт события и не принимает управляющие посылки.

Также сервер сообщает контроллеру, поддерживает ли он ONLINE проверку доступа

Запрос:

```
{
  "id":123456789,
  "operation":"set_active",
  "active":1,
  "online":1
}
```

active - 1 - активация контроллера, 0 - деактивация

online - 1 - сервер поддерживает режим ONLINE, 0 - не поддерживает

Ответ:

```
{
  "id":123456789,
  "success ":1
}
```

success - 1- команда принята, 0 - ошибка

## 2.2 OPEN\_DOOR

Вызывает срабатывание выходного каскада в заданном направлении

Запрос:

```
{
  "id":123456789,
  "operation":"open_door",
  "direction": 0
}
```

direction - 0 - вход, 1 - выход

Ответ:

```
{
  "id":123456789,
  "success ":1
}
```

## 2.3 SET\_MODE

Устанавливает режим работы контроллера (Норма, Свободный проход, Блокировка, Ожидание Свободного прохода)

Запрос:

```
{
  "id":123456789,
  "operation":"set_mode",
  "mode": 2
}
```

возможные режимы: 0 - норма, 1 - блок, 2 - свободный проход, 3 - ожидание свободного прохода

Ответ:

```
{
  "id":123456789,
  "success ":1
}
```

```
}
```

## 2.4 SET\_TIMEZONE

Устанавливает параметры временной зоны контроллера

Запрос:

```
{
  "id":123456789,
  "operation":"set_timezone",
  "zone": 0,
  "begin":"00:00",
  "end":"23:59",
  "days":"11111110"
}
```

zone - номер временной зоны (0 - 6)

begin - время начала действия зоны

end - время окончания действия зоны

days - маска дней недели для зоны (0 - зона выключена, 1 -включена), понедельник - 1-й

Ответ:

```
{
  "id":123456789,
  "success ":1
}
```

## 2.5 SET\_DOOR\_PARAMS

Устанавливает параметры открывания и контроля состояния двери

Запрос:

```
{
  "id":123456789,
  "operation":"set_door_params",
  "open":30,
  "open_control":50,
  "close_control":50
}
```

open - время подачи сигнала открывания замка (в 1/10 секунды)

open\_control - время контроля открытия двери (в 1/10 секунды)

close\_control - время контроля закрытия двери (в 1/10 секунды)

Ответ:

```
{
  "id":123456789,
  "success ":1
}
```

## 2.6 ADD\_CARDS

Добавляет карты в память контроллера. Если в памяти контроллера уже имеется карта с таким-же номером, для этой карты обновляются флаги и временные зоны.

Запрос:

```
{
  "id":123456789,
  "operation":"add_cards",
  "cards": [
    {
      "card": "00B5009EC1A8",
      "flags": 0,

```

```
    "tz": 255
  },
  {
    "card": "0000000FE32A2",
    "flags": 32,
    "tz": 255
  }
]
```

cards - массив карт для добавления

card - номер карты в шестнадцатеричном виде (см. ПРИЛОЖЕНИЕ 2)

flags - флаги для карты (8 - блокирующая карта, 32 - короткий код карты (три байта))

tz - временные зоны для карты

**Ответ:**

```
{
  "id":123456789,
  "success ":1
}
```

## 2.7 DEL\_CARDS

Удаляет карты из памяти контроллера

Запрос:

```
{
  "id":123456789,
  "operation":"del_cards",
  "cards": [
    {"card":"000000A2BA93"},
    {"card":"000000A2A18A"}
  ]
}
```

cards - массив карт для удаления, содержит номера карты в шестнадцатеричном виде (см. ПРИЛОЖЕНИЕ 2)

**Ответ:**

```
{
  "id":123456789,
  "success ":1
}
```

## 2.8 CLEAR\_CARDS

Удаляет все карты из памяти контроллера

Запрос:

```
{
  "id":123456789,
  "operation":"clear_cards"
}
```

**Ответ:**

```
{
  "id":123456789,
  "success ":1
}
```

## ПРИЛОЖЕНИЕ 1: Коды событий

Событие	Вход	Выход	Номер карты	Флаги
открыто кнопкой изнутри	0	1		
ключ не найден в банке ключей	2	3		
ключ найден, дверь открыта	4	5	Есть	
ключ найден, доступ не разрешен	6	7	Есть	
Открыто оператором по сети	8	9		
ключ найден, дверь заблокирована	10	11	Есть	
дверь взломана	12	13		
дверь оставлена открытой (timeout)	14	15		
Проход состоялся	16	17	Есть	
Перезагрузка контроллера	20			
Питание	21			0 - пропало 1 - появилось
Дверь открыта	32	33	Есть	
Дверь закрыта	34	35		
Переключение режимов работы (см Режим)	37			Флаги Режимов
Пожарные события (см Пожар)	38			Флаги Пожара
Охранные события (см Охрана)	39			Флаги Охраны
Проход не совершен за заданное время	40	41	Есть	
Совершен вход в шлюз	48	49	Есть	
Заблокирован вход в шлюз (занят)	50	51	Есть	
Разрешен вход в шлюз	52	53	Есть	
Заблокирован проход (Антипассбек)	54	55	Есть	

## ПРИЛОЖЕНИЕ 2: Преобразование кодов карт

При записи карты в контроллер и передачи событий от контроллера (а также в режиме online), код карты передаётся в виде строки, представляющей 6 байт кода в шестнадцатиричной системе. Количество значащих байт зависит от протокола считывателя подключённого к контроллеру.

Для протокола "iButton" передаются 6 значащих байт:

```
"card": "665544332211"
```

Для протокола "Wiegand" передаются 3 значащих байт:

```
"card": "000000332211"
```

Если у контроллера включен протокол "Wiegand", все записываемые в него карты обрезаются до 3х байт.

На картах и брелоках Em-Marine обычно пишется их номер в формате "187,01899".

Пример преобразования такого номера в шестнадцатиричную строку:

187 (dec) = 0x**BB** (hex)

1899 (dec) = 0x**076B** (hex)

Код карты:

```
"card": "000000BB076B"
```

Это не полный код карты Em-Marine. Полный код составляет 5 байт, 2 дополнительных байта на карте не напечатаны, но считыватель передаёт их в контроллер. Поэтому при записи такого кода в контроллер надо добавить признак короткого кода:

```
"flags": 32
```

На картах и брелоках Mifare номер пишется в виде "AB721582". это уже шестнадцатиричное представлени 4х байт номера карты. Для записи в контроллеры:

```
"card": "0000AB721582"
```