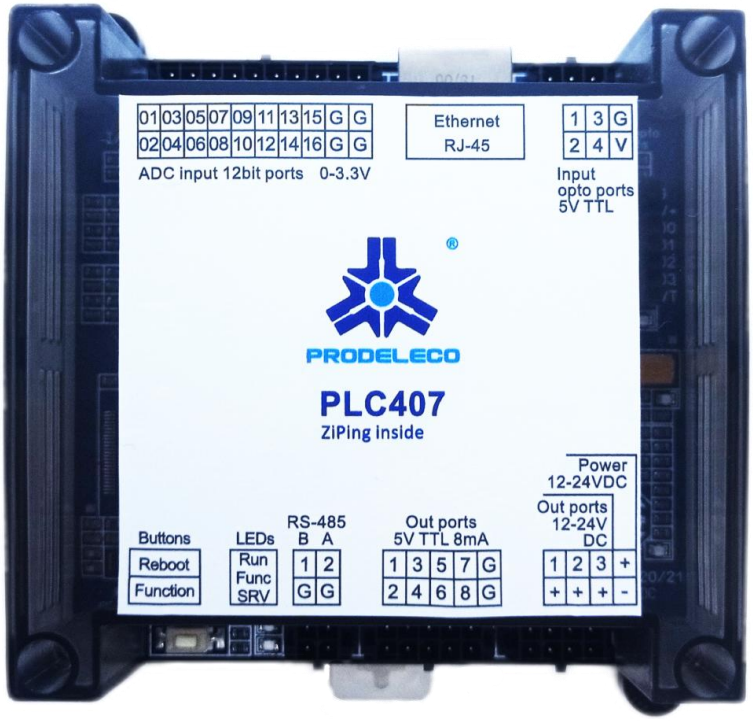
Руководство по эксплуатации.
Контроллер PLC407



Оглавление

Описание

Программируемый логический контроллер PLC407 Рис.1 (далее по тексту ПЛК) предназначен для автоматизации процессов, диспетчеризации узлов учета, мониторинга состояния систем.

Для удобства и простоты настройки, на контроллере предусмотрен WEB интерфейс, что позволяет подготавливать контроллер к работе, не обладая специальными инженерными навыками, а также без дополнительного программного обеспечения и оборудования.

Контроллер разработан для монтажа в электрический шкаф на DIN рейку посредством пластикового корпуса с зажимом, также в корпусе предусмотрено отверстие для крепления без DIN рейки.

Настоящее Руководство по эксплуатации (далее по тексту Руководство) описывает порядок хранения, монтажа, эксплуатации и обслуживания контроллера.

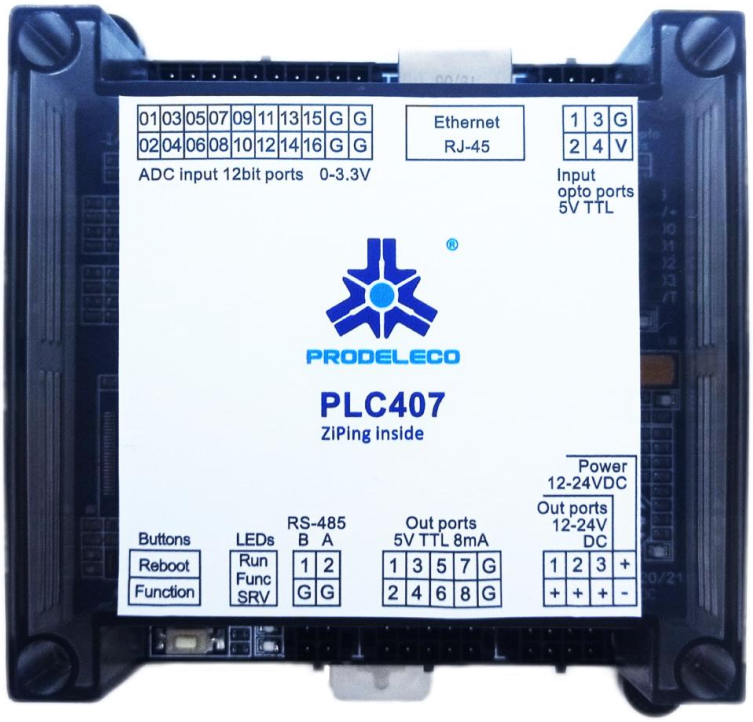


Рис.1

Предосторожности и рекомендации

Перед использованием ПЛК внимательно прочитайте данное руководство. Строго соблюдайте требования техники безопасности.

ПРЕДУПРЕЖДЕНИЯ!

- **При любых монтажных работах, связанных с подключением и отключением проводов с входов и выходов контроллера, а также интерфейсных кабелей, контроллер должен быть обесточен.**
- **Не подавайте на входы и транзисторные выходы контроллера повышенное и переменное напряжение даже ошибочно, так как это приведет к разрушению устройства.**
- **Перед выполнением монтажных работ, после отключения напряжения питания подождите 1 минуту до полной разрядки конденсаторов.**
- **На печатной плате контроллера расположены чувствительные к статическому электричеству электронные компоненты. Во избежание повреждения элементов или цепей на печатной плате не следует касаться их голыми руками, либо металлическими предметами.**
- **В случае попадания посторонних (особенно электропроводящих) предметов внутрь ПЛК отключите напряжение сети и извлеките их.**
- **ПЛК является электрическим оборудованием, предназначенным для установки в шкафы управления или аналогичные закрытые рабочие пространства со степенью защиты, обеспечивающей требуемые условия эксплуатации.**
- **Невыполнение требований, изложенных в настоящем Руководстве, может привести к отказам, вплоть до выхода ПЛК из строя.**

При невыполнении потребителем требований и рекомендаций настоящего Руководства, Производитель снимает с себя гарантийные обязательства по бесплатному ремонту отказавшего контроллера! Производитель также не несёт гарантийной ответственности по ремонту при несанкционированной модификации ПЛК.

Технические характеристики

ПЛК PLC407 (рис.2) имеет в составе:

- 4 дискретных входов DI 5/12/24V
- 16 аналоговых входов AI 0-5В, 0-10В, 4-20мА
- 8 дискретных выходов DO 5V 8mA, 3 DO 12/24v 2A
- Modbus-RTU RS-485 1 порт
- Ethernet 1 порт
- корпус на DIN рейку
- http клиент отправки post запросов на сервер

ПЛК запитывается постоянным током до 1 Ампера и напряжением 12/24v Вольт.

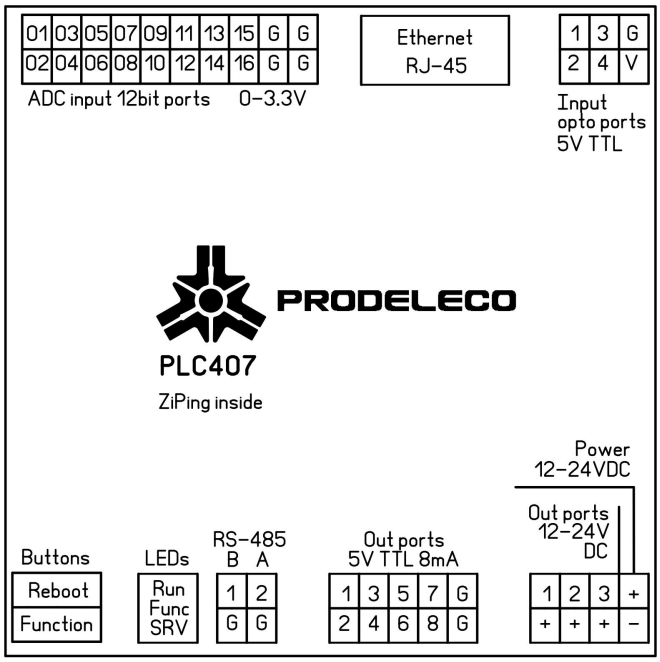


Рис.2

Входные порты (рис.2):

1. ADC Channels – 16 аналоговых входов. Диапазон измерения 0-3.3Вольт. Предназначен для подключения внешнего модуля адаптации сигналов (0-5В, 0-10В, 4-20мА), фоторезисторы, дальномеры, датчики холла, группа резистивных переключателей, потенциометры, простые переключатели/кнопки.

2. Input opto ports – 4 дискретных входа 5/12/24V с оптической развязкой для подключения внешних переключателей управления режимами ПЛК.

3. RS-485 – последовательный порт опроса внешних устройств по протоколу Modbus RTU.

4. Ethernet – RJ45 порт для подключения контроллера к локальной сети посредством восьмижильной витой пары. Возможно подключение двух контроллеров между собой напрямую без коммутатора. С помощью локальной сети по протоколу

HTTP можно производить настройки контроллера через WEB интерфейс, встроенный в контроллер. Также с помощью локальной сети контроллеры взаимодействуют по протоколу HTTP REST API друг с другом или с другим устройством.

Выходные порты (рис.2):

1. Out ports 5V TTL – 8 каналов по 8мА каждый, но не более 30мА на все каналы. Для управления внешними устройствами (твердотельные реле, переключатели, индикация).

2. Out ports 12-24DC – три транзисторных выхода, объединенные с напряжением питания контроллера и коммутирующие землю внешнего устройства. Нагрузочная способность каждого канала до 2А. Подключаемое оборудование: электромоторы постоянного тока, электромагнитные замки, электромагнитные реле, твердотельные реле, электромагнитные контакторы, прожектора, светильники, электромеханические замки и т. д.

При подключении индуктивной нагрузки необходимо подключать параллельно контактам нагрузки выпрямительный диод обратной полярностью для предотвращения выхода из строя транзистора разрядом катушки индуктивности. При подключении к транзисторным портам нагрузки необходимо увеличить мощность блока питания до двукратной мощности подключаемой нагрузки плюс 2А.

Индикация работы контроллера (рис.2):

1. Run – индикация режима работы ПЛК.

2. Func – индикация переключения функций.

3. Srv – индикатор работы сетевого интерфейса Ethernet.

Кнопки на плате контроллера (рис.2):

1. REBOOT – кнопка аппаратной перезагрузки контроллера. Аналогично выключению и включению питания. Кратковременное нажатие приводит к перезагрузке микроконтроллера.

2. RESET – сброс пользовательских настроек и полная очистка памяти (возврат к заводскому состоянию). При нажатии и удержании в нажатом состоянии ~5 секунд индикатор Srv начинает моргать. Это означает, что, отпустив кнопку в данный момент, но не позднее чем индикатор начнет гореть непрерывно (<20 секунд), произойдет сброс IP адреса, Логина и Пароля пользователя на начальные (IP addr: 192.168.88.55; Login: admin; Pass: admin). При удержании в нажатом состоянии кнопки RESET более ~20 секунд, индикатор Srv начинает гореть непрерывно. Это означает, что при отпускании кнопки в данный момент произойдет очищение информации Flash памяти включая все текущие настройки, медиа файлы и веб интерфейс.

Примеры реализации подключения входов и выходов

См. Рис.3, Рис. 4

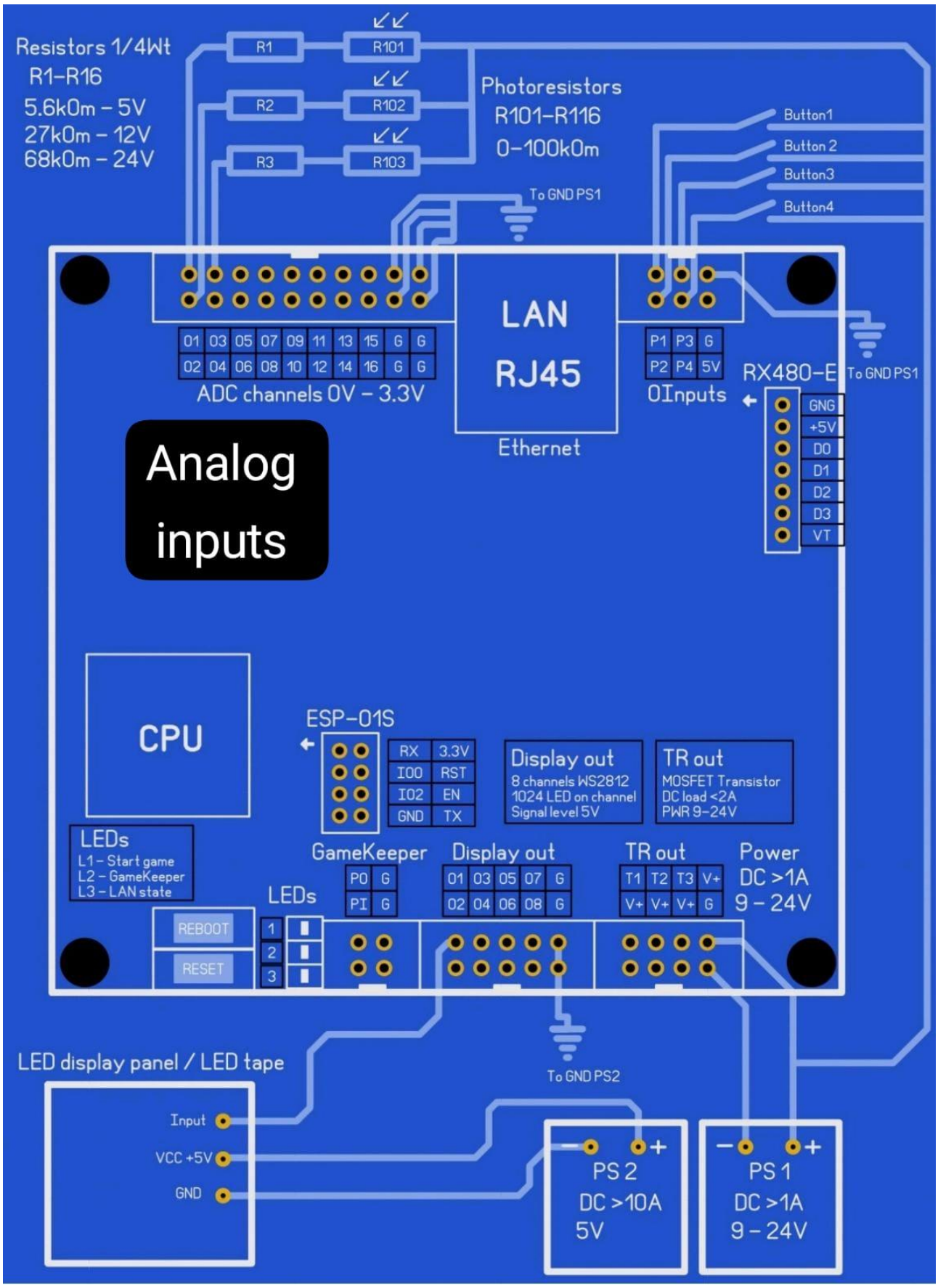

Resistors 1/4Wt
R1–R16
5.6kOm – 5V
27kOm – 12V
68kOm – 24V
R1
R2
R3
R101
R102
R103
Photoresistors
R101–R116
0–100kOm
To GND PS1
Button1
Button 2
Button3
Button4
LAN
RJ45
Ethernet
01 03 05 07 09 11 13 15 G G
02 04 06 08 10 12 14 16 G G
ADC channels 0V – 3.3V
P1 P3 G
P2 P4 5V
OInputs
RX480-E
GNG
+5V
D0
D1
D2
D3
VT
To GND PS1
Analog inputs
CPU
ESP-01S
RX 3.3V
IO0 RST
IO2 EN
GND TX
Display out
8 channels WS2812
1024 LED on channel
Signal level 5V
TR out
MOSFET Transistor
DC load <2A
PWR 9–24V
LEDs
L1 – Start game
L2 – GameKeeper
L3 – LAN state
GameKeeper
PO G
PI G
Display out
01 03 05 07 G
02 04 06 08 G
TR out
T1 T2 T3 V+
V+ V+ V+ G
Power
DC >1A
9 – 24V
LEDs
1
2
3
REBOOT
RESET
LED display panel / LED tape
Input
VCC +5V
GND
To GND PS2
PS 2
DC >10A
5V
PS 1
DC >1A
9 – 24V

Рис.3

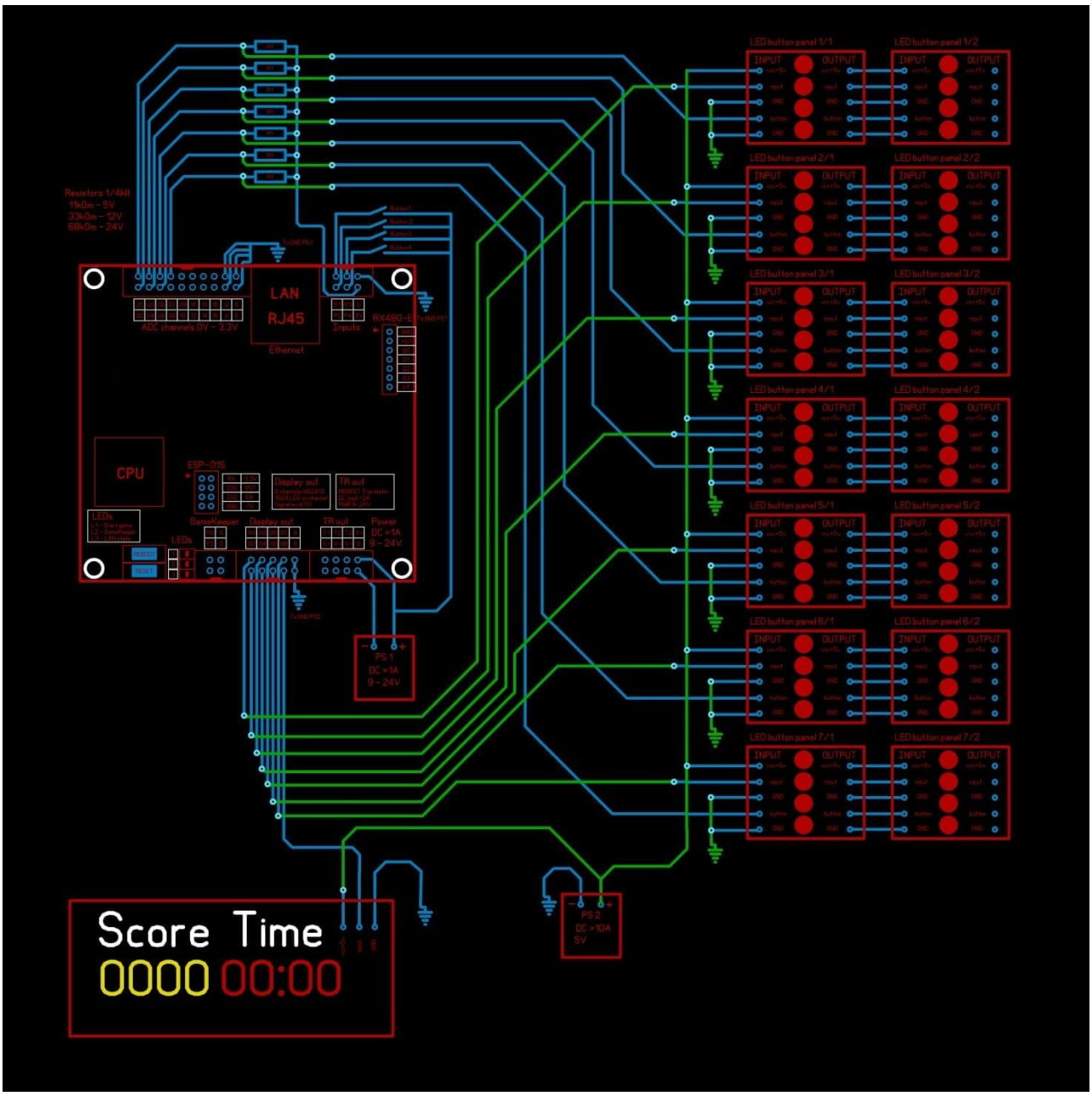

LED button panel 1/1
LED button panel 1/2
LED button panel 2/1
LED button panel 2/2
LED button panel 3/1
LED button panel 3/2
LED button panel 4/1
LED button panel 4/2
LED button panel 5/1
LED button panel 5/2
LED button panel 6/1
LED button panel 6/2
LED button panel 7/1
LED button panel 7/2
INPUT
OUTPUT
Resistors 1/4W
1k0m - 5V
33k0m - 12V
68k0m - 24V
LAN
RJ45
Ethernet
ADC channels 0V - 3.3V
Inputs
CPU
ESP-01S
Display out
TR out
Power
DC >1A
9 - 24V
PS 1
DC >1A
9 - 24V
PS 2
DC >10A
5V
Score Time
0000 00:00

Рис.4

Общие указания по монтажу

1. Не устанавливайте ПЛК в местах с наличием большого количества токопроводящей пыли, в химически-агрессивных средах, в местах с влажностью более 50% или возможностью попадания влаги, вблизи источников тепла, и в местах с вибрацией более 55Гц с амплитудой не более 0,15 мм 2. Рекомендуется устанавливать ПЛК в шкафы или щиты управления с необходимой степенью защиты и охлаждением.

3. Для обеспечения нормального теплового режима ПЛК его необходимо устанавливать в вертикальном положении, обеспечив свободную конвекцию воздуха в воздушном коридоре: с боков – не менее 50 мм, сверху и снизу – не менее 50 мм, как показано на (рис.5).

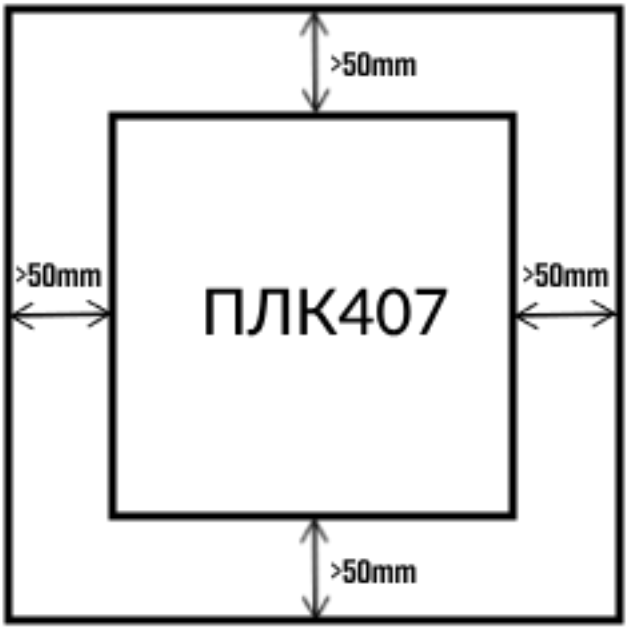


Рис.5

4. Все ПЛК, поставляемые в корпусе, имеют встроенный крепеж для установки на DIN-рейку и так же могут быть установлены на плоскую поверхность с помощью винтов М4.

5. Монтаж должен проводиться квалифицированным персоналом с соблюдением требований настоящего Руководства, а также ПУЭ-98 и СНиП - 4.6. – 82.

6. Не допускайте попадания металлических предметов (обрывки проводников, стружка и т.д.) внутрь при подключении.

7. Исключите возможность попадания на входы и выходы повышенного напряжения или напряжения, несоответствующего спецификации контроллера.

8. Не подсоединяйте и не отсоединяйте провода при поданном напряжении питающей сети.

9. Избегайте образования острых изгибов проводников.

10. Всегда используйте как можно более короткий провод (максимум 50 м – экранированный, 30 м – неэкранированный). Провода должны прокладываться попарно с использованием нейтрального, или общего провода в паре с проводом под напряжением, или проводом, несущим сигнал.

11. При длинных кабелях сечение их должно выбираться с учетом возможного падения напряжения.

12. Прокладывайте сигнальные кабели отдельно от силовых кабелей или под углом примерно 90° к силовым проводам. Так же старайтесь прокладывать отдельно входные и выходные кабели.

13. Рекомендуется использовать медный многожильный кабель с кабельными наконечниками.

14. Снабдите индуктивные нагрузки гасящими цепочками, которые ограничивают скачки напряжения при выключении питания.

15. Устанавливайте надлежащие устройства ограничения перенапряжений для всех проводов, подверженных ударам молнии.

Предупреждение!

Устройства управления могут выходить из строя в опасном состоянии, приводя к непредвиденному действию управляемого оборудования. Такие непредвиденные действия могут привести к смерти или серьезной травме персонала и/или повреждению оборудования. Обдумайте вопрос об использовании функции аварийного останова, электромеханических блокировок или других резервных предохранительных устройств, не зависимых от ПЛК.

Монтажные размеры платы

Плата контроллера имеет четыре отверстия под винты М3, расположенные симметрично относительно центров сторон платы (рис.6)

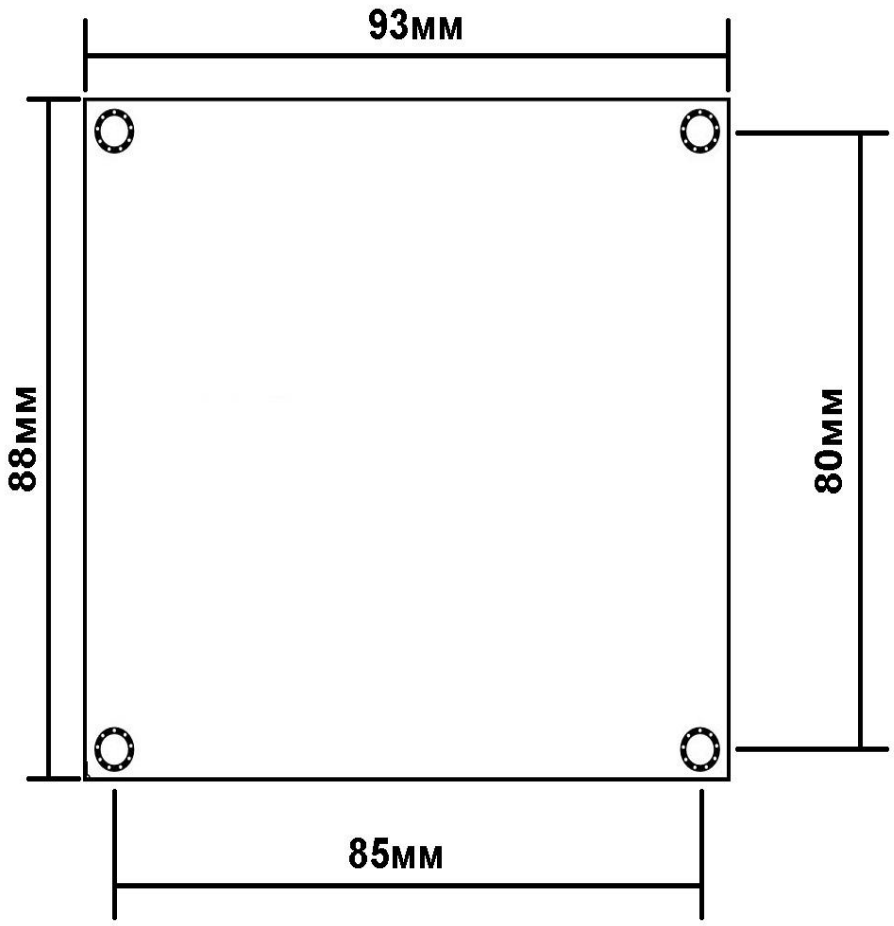


Рис.6

Корпус контроллера

Корпус 90x95x40мм (рис.7, рис.8) контроллера может устанавливаться как на DIN рейку, так и непосредственно на монтажную поверхность двумя винтами M4 или саморезами 3мм длиной не менее 10мм. Данный корпус снабжен прозрачной крышкой.

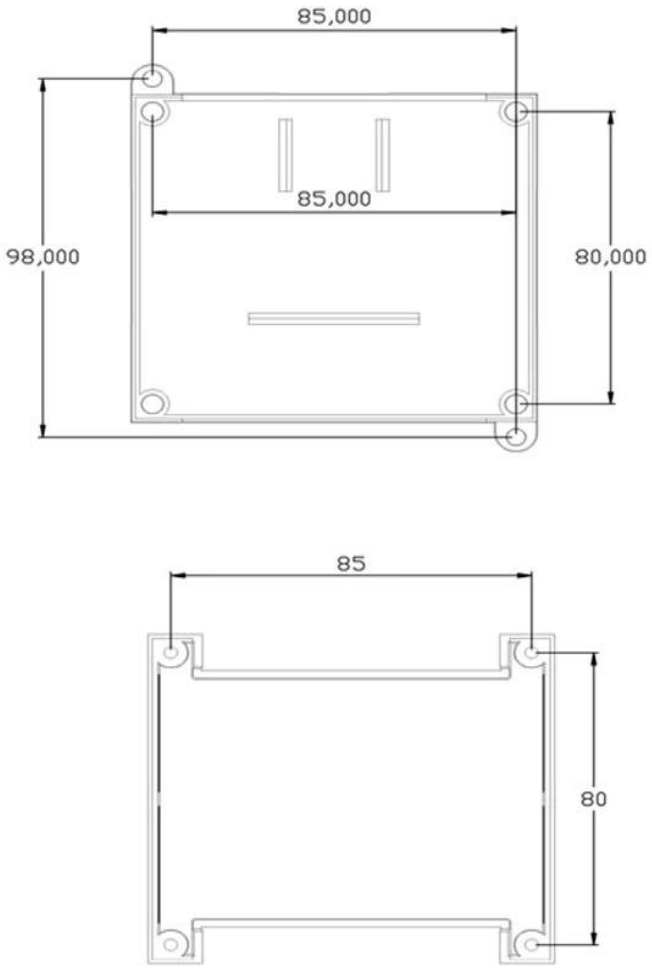


Рис.7

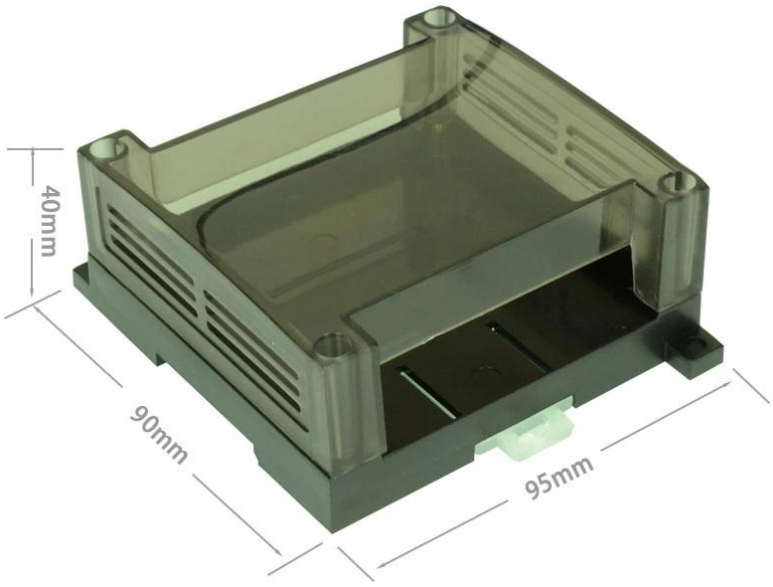


Рис.8

Подключение контроллера

Подключение к датчикам и кнопкам осуществляется многожильным медным проводом сечением не менее 0.75мм^2 и длиной не более 10м.

Подключение к транзисторным портам осуществляется многожильным медным проводом сечением не менее 1.0мм^2 и длиной не более 10м.

Интерфейс контроллера

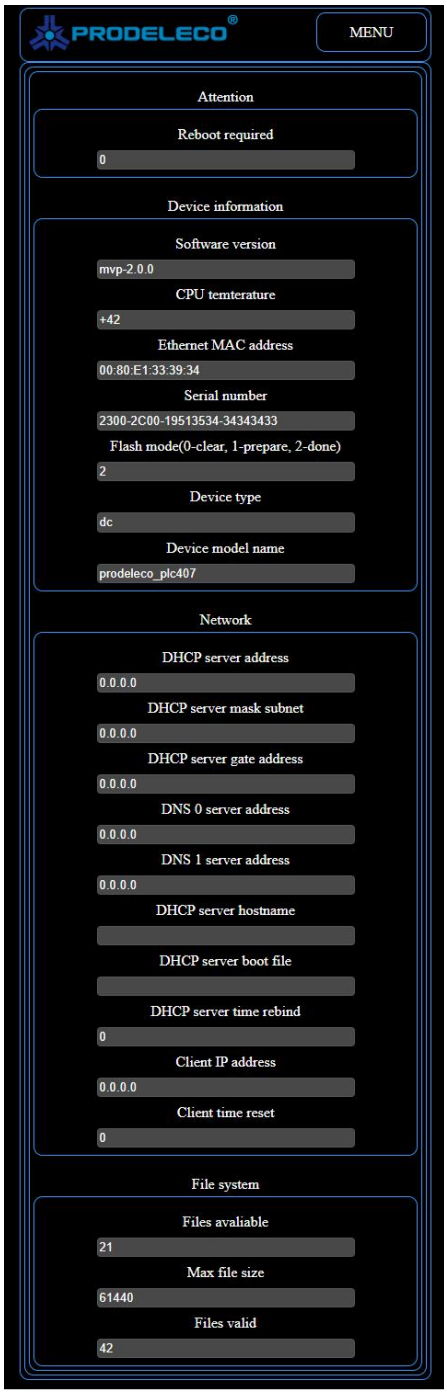
Зайдите браузере устройства по указанному на контроллере IP адресу.

Укажите указанный на корпусе контроллера логин и пароль в первом и втором поле соответственно.

Откроется главная страница интерфейса (Main). На странице отображаются основные параметры состояния контроллера.

В меню располагаются разделы, описанные в п.1-10.

1. Main

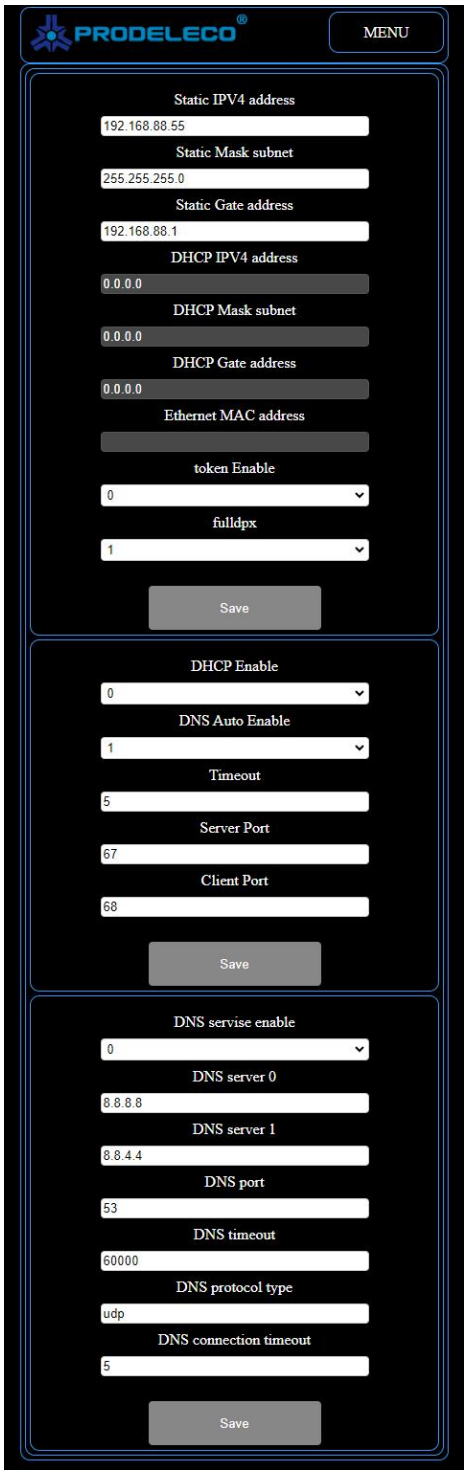


На странице отображается состояние контроллера. Настройка осуществляется в соответствующем разделе меню.

- **Attention**
  - **Reboot requred**.
    0 - не требуется перезагрузка контроллера.
    1 - требуется перезагрузка контроллера.
- **Device information**. Информация о контроллере
  - **Software version**. Версия программы контроллера
  - **CPU temterature**. Текущая температура процессора
  - **Ethernet MAC address**. MAC адрес
  - **Serial number**. Серийный номер
  - **Flash mode (0-clear, 1-prepare, 2-done)**. Режим работы flash памяти: 0 очищена, 1 подготовка, 2 готова.
  - **Device type**. Тип питания
  - **Device model name**. Модель контроллера
- **Network**. Сетевые настройки
  - **DHCP server address**. Адрес DHCP сервера
  - **DHCP server mask subnet**. Маска подсети DHCP сервера
  - **DHCP server gate address**. Адрес шлюза DHCP сервера
  - **DNS 0 server address**. Предпочтительный DNS адрес
  - **DNS 1 server address**. Альтернативный DNS адрес
  - **DHCP server hostname**. Имя DHCP сервера
  - **DHCP server boot file**. Загрузочный файл DHCP сервера
  - **DHCP server time rebind**. Повторная привязка времени DHCP сервера
  - **Client IP address**. IP адрес клиента
  - **Client time reset**. Сброс времени клиента.
- **File system**. Сетевые настройки
  - Files available. Доступное количество файлов к загрузке
  - **Max file size**. Максимальный размер одного загружаемого файла
  - **Files valid**. Количество ликвидных файлов

При изменении настроек необходимо сохранить изменения, нажав на кнопку Save соответствующего блока.

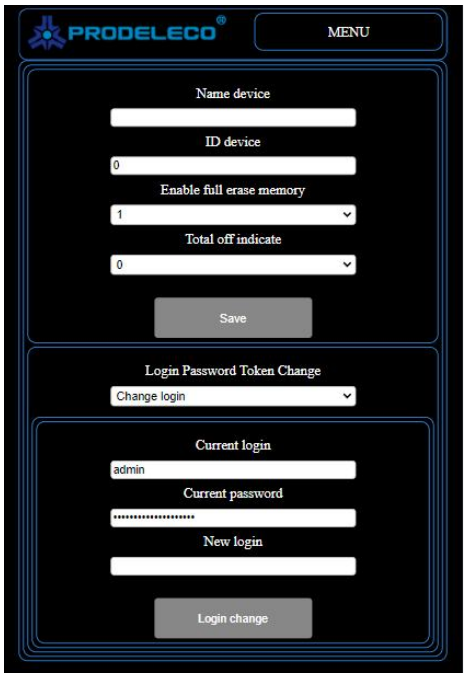
2. Network



- Static IPV4 address. Статический IP адрес контроллера. Поле редактируемо, если используется статический IP адрес.
- Static Mask subnet. Маска подсети. Поле редактируемо, если используется статический IP адрес.
- Static Gate address. Адрес шлюза. Поле редактируемо, если используется статический IP адрес.
- DHCP IPV4 address. Динамический IP адрес контроллера.
- DHCP Mask subnet. Маска подсети.
- DHCP Gate address. Адрес шлюза.
- Ethernet MAC address. MAC адрес
- token Enable. Режим использования токена. 0 - не используется, 1 - используется.
- Fulldpx. Режим Full duplex (одновременный прием и передача данных). 0 - не используется, 1 - используется.
- Настройки DHCP
- Настройки DNS

При изменении настроек необходимо сохранить изменения, нажав на кнопку Save соответствующего блока.

3. Config



— Name device. Наименование устройства.

— ID device. Идентификатор устройства

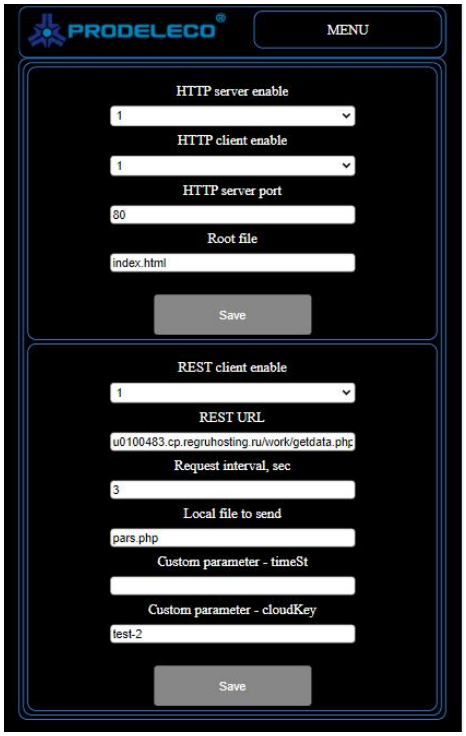
— Enable full erase memory. Режим полной очистки памяти. 0 - не задействован, 1 - задействован.

— Total off indicate. 0 - выключено, 1 - включено.

— Login/Password/Token Change. Смена логина/пароля. При выборе соответствующего пункта, ниже открываются поля для смены Логина, пароля или токена.

При изменении настроек необходимо сохранить изменения, нажав на кнопку Save соответствующего блока.

4. Services



— HTTP server enable. Включение HTTP сервера: 0 - выключен, 1 - включен.

— HTTP client enable. Включение HTTP клиента: 0 - выключен, 1 - включен.

— HTTP server port. Порт HTTP сервера.

— Root file. Наименование корневого файла

— REST client enable. Включение REST клиента. 0 - выключен, 1 - включен.

— REST URL. URL адрес сервиса.

— Request interval, sec. Интервал запроса в секундах

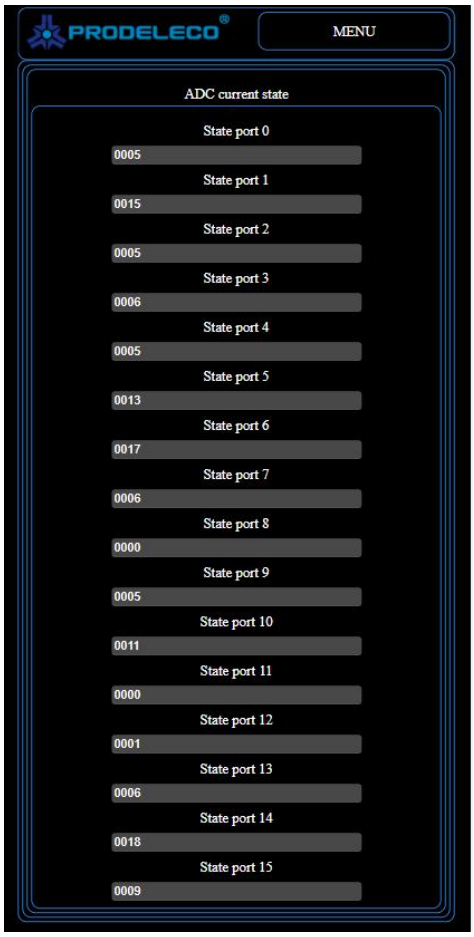
— Local file send. Локальный отправляемый файл.

— Custom parameter-time St

— Custom parameter-cloud key.

При изменении настроек необходимо сохранить изменения, нажав на кнопку Save соответствующего блока.

5. ADC state

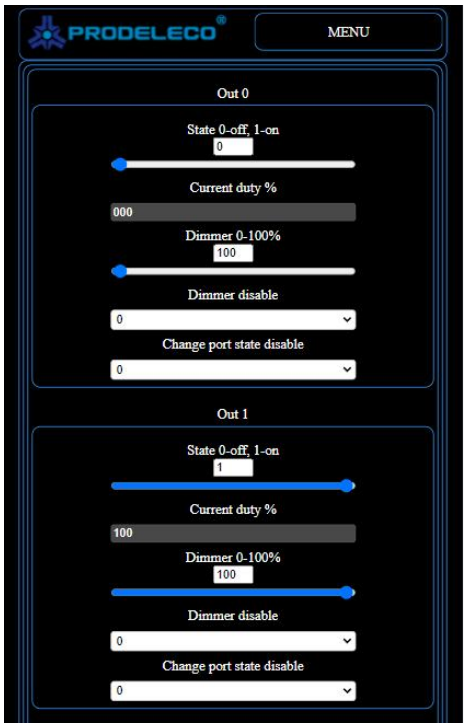


Указываются состояния портов аналогово-цифровых преобразователей

6. In state

Указывается состояние входных дискретных портов (0 или 1)

7. Outputs

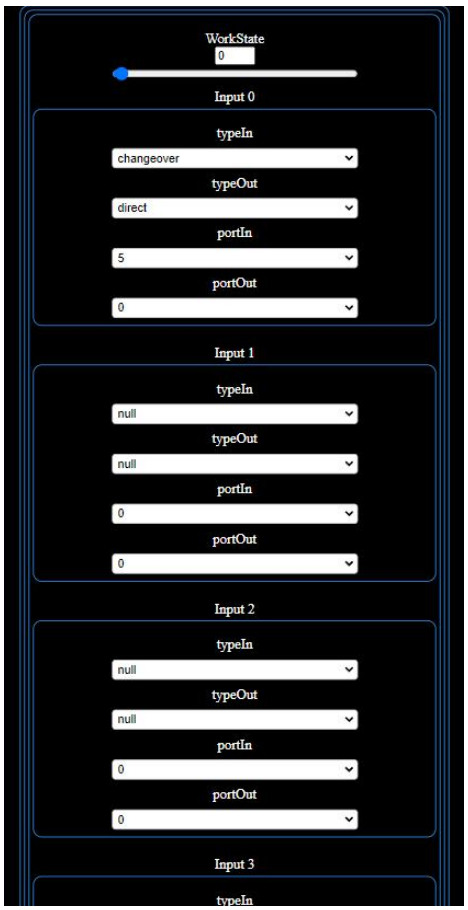


Настройка параметров выходных портов

Каждый из 12 портов имеет настройки:

- State. Текущее состояние
- Current duty %. Отображение текущего процента диммирования
- Dimmer. Установка уровня диммирования
- Dimmer disable. Отключение режима диммирования
- Change port state disable. Отключение режима смены состояния порта

8. Rules

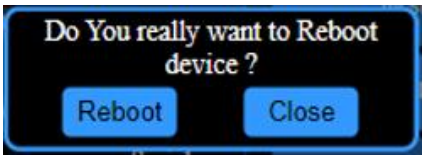


Настройка правил работы контроллера

Workstate - включение настроенных алгоритмов

- typeIn тип входящего сигнала имеет значения:
  - null - порт входа не используется
  - change over - смена состояния
  - stable - повторение входного сигнала
- typeOut тип входящего сигнала имеет значения:
  - null - порт выхода не используется
  - direct - передача состояния без изменения
  - inverse - передача обратного состояния
- portIn - назначеие входного порта
- portOut - назначение выходного порта.

9. Reboot



При выборе пункта меню перезагрузка откроется окно подтверждения перезагрузки устройства и применения параметров.

10. Logout

При выборе пункта меню Logout (Выйти) откроется окно подтверждения выхода из-под текущего пользователя.

REST API

REST API - набор инструментов для взаимодействия различных (разнородных) программ, работающих в формате запрос/ответ.

REST PLC поддерживает протоколы передачи данных: HTTP, HTTPS, FTP, NTP, DNS, DHCP, MODBUS. Это позволяет применять технологию на большинстве современных контроллеров и интегрировать их в уже имеющиеся автоматизированные системы управления.

ПЛК оснащен программой, обеспечивающей реализацию стека TCP/IP, UDP с прикладными протоколами HTTP, HTTPS, FTP, NTP, DNS, DHCP. Это позволяет реализовывать полноценный веб сервер, управляющий ресурсами микроконтроллера на 8ми битных и 32х битных микроконтроллерах.

- позволяет настроить локальную (автономную) логику работы контроллера на скриптовом языке с синтаксисом PHP и набором специальных функций управления выходами и анализа входных сигналов (чтение датчиков, регистров, портов микроконтроллера),

- имеет возможность формировать отчеты в текстовых форматах (yml, html, json, xml), загрузку и выгрузку файлов в память микроконтроллера, передавать данные в raw формате.

Примеры скриптов RestPLC:

1)управление выходным портом по изменению состояния входного (chageOver)

```
<php
//сохранение состояния входного порта в переменную $state
$state =pI[0].value;
//основной цикл контроллера
while(1)
{
//условие определяющее изменение состояния входного порта
if ($state != pI[0].value)
{
//сохранение нового состояния входного порта в переменную $state
$state = pI[0].value;
//условие определяющее состояние выходного порта
if (pO[0].value!=0)
{
//установка состояния выходного порта в 0
pO[0].value=0;
}
else
{
//установка состояния выходного порта в 1
pO[0].value=1;
}
}
}
?>
```

2)Входящие HTTP запросы

Платформа RestPLC позволяет обрабатывать входящие запросы по HTTP протоколу, выполнять роль мини HTTP сервера со встроенным препроцессором nanoPHP и возвращать данные в форматах HTML, JSON, XML, YAML. Мини HTTP сервер позволяет создавать непосредственно в памяти RestPLC Web интерфейс, управляющий функциями и настройками ПЛК. Так как RestPLC может хранить файлы любого формата, то в сочетании с JS, CSS и графическими файлами можно создавать удобный графический интерфейс с применением навыков только веб разработчика.

Страница с настройками IP адресов:

```
<?php
header('Content-Type: text/html');
echo '<HTML><BODY><table>\r\n';
echo '<tr><td>IP Addr</td><td><INPUT Type=Text Name="lan.ipV4A" Value="'. $lan[ipV4A].'"></td></tr>\r\n'.
'<tr><td>Mask subnet</td><td><INPUT Type=Text Name="lan.ipV4M" Value="'. $lan[ipV4M].'"></td></tr>\r\n'.
'<tr><td>Getway</td><td><INPUT Type=Text Name="lan.ipV4G" Value="'. $lan[ipV4G].'"></td></tr>\r\n'.
'<tr><td>MAC</td><td><INPUT Type=Text Name="lan.macAddr" Value="'. $lan[macAddr].'"></td></tr>\r\n';
echo '</table></BODY></HTML>';
?>
```

Запрос параметров контроллера встроенным скриптом pars.php в формате JSON.

4) Исходящие HTTP запросы.

Request.

В платформе RestPLC реализован HTTP клиент, с помощью которого можно обеспечить ситуативную (по расписанию или по событиям от датчиков) отправку HTTP запросов на внешние сервера с необходимым набором данных в форматах HTML, JSON, XML, YAML, обработать ответ содержащий данные управляющие настройками и функциями ПЛК.

Пример запроса на сервер в формате JSON, содержащий состояния выходных портов контроллера:

```
{"out":{
"0": {
"state":"1",
"currentDuty":"20",
"disableDuty":"0",
"disableChange":"0",
"pwmDuty":"20"
},
"1": {
"state":"0",
"currentDuty":"0",
"disableDuty":"0",
"disableChange":"0",
```

"pwmDuty":"35"
},
"2": {
"state":"1",
"currentDuty":"50",
"disableDuty":"0",
"disableChange":"0",
"pwmDuty":"50"
},
"3": {
"state":"1",
"currentDuty":"100",
"disableDuty":"1",
"disableChange":"0",
"pwmDuty":"15"
}
}
}

В соответствии с примером видно следующее состояние выходных портов контроллера:

0-й — включен в режиме PWM (ШИМ) со скважностью 20%,

1-й — выключен, режим PWM разрешен, настройка скважности 35%,

2-й — включен в режиме PWM со скважностью 50%,

3-й — включен с запрещенным режимом PWM.

Response.

Сервер может ответить на запрос данными в формате JSON для управления параметрами контроллера (вкл/выкл портов, режимы, настройки интерфейсов, внутренняя логика)

Пример ответа сервера в формате JSON, содержащий команды установки состояния выходных портов контроллера:

{"out":{
"0": {
"state":"0"
},
"1": {
"state":"1",
"disableDuty":"1",
"pwmDuty":"100"
},
"2": {
"state":"1",
"disableDuty":"0",
"pwmDuty":"20"
},

```
"3": {
"state":"0",
}
}
}
```

Ответ сервера обрабатывается в момент получения и сразу применяются параметры. В данном примере 0-й и 3-й порт выключаем, 1-й включается и запрещается PWM, 2-й включается в режиме PWM со скважностью 20%.

Ответ сервера может содержать произвольные файлы, как и скрипты для управления в автономном режиме.

Ключевой особенностью является удаленная замена алгоритма ПЛК с центрального сервера, что существенно снижает затраты на обслуживание систем автоматизации и повышает вариабельность системы.

Примеры скриптов RestPLC:

1) управление выходным портом по изменению состояния входного (chageOver)

```
<php
//сохранение состояния входного порта в переменную $state
$state =pI[0].value;
//основной цикл контроллера
while(1)
{
//условие определяющее изменение состояния входного порта
if ($state != pI[0].value)
{
//сохранение нового состояния входного порта в переменную $state
$state = pI[0].value;
//условие определяющее состояние выходного порта
if (pO[0].value!=0)
{
//установка состояния выходного порта в 0
pO[0].value=0;
}
else
{
//установка состояния выходного порта в 1
pO[0].value=1;
}
}
}
?>
```

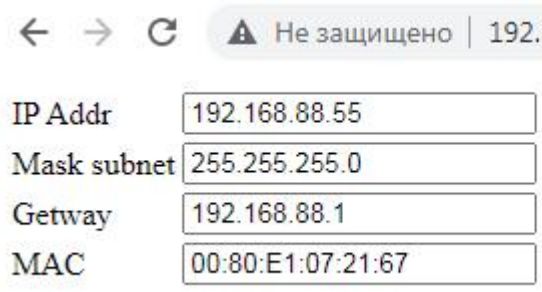
2) Входящие HTTP запросы

Платформа RestPLC позволяет обрабатывать входящие запросы по HTTP протоколу, выполнять роль мини HTTP сервера со встроенным препроцессором nanoPHP и возвращать данные в форматах HTML, JSON, XML, YAML. Мини HTTP сервер позволяет создавать непосредственно в памяти RestPLC Web интерфейс, управляющий функциями и настройками ПЛК. Так как RestPLC может хранить файлы любого формата, то в сочетании с JS, CSS и графическими файлами можно создавать удобный графический интерфейс с применением навыков только веб разработчика.

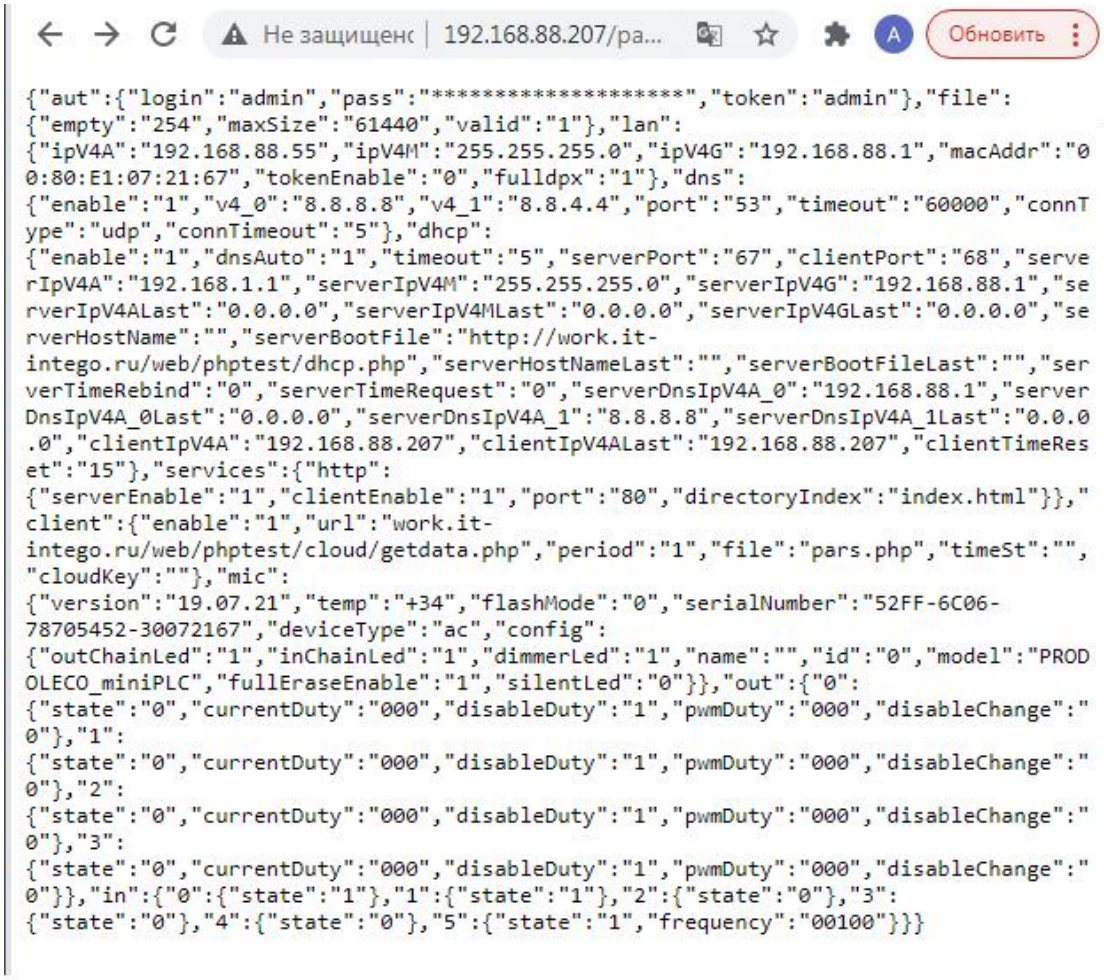
Страница с настройками IP адресов:

```
<?php
header('Content-Type: text/html');
echo '<HTML><BODY><table>\r\n';
echo '<tr><td>IP Addr</td><td><INPUT Type=Text Name="lan.ipV4A" Value="'. $lan[ipV4A].'"></td></tr>\r\n'.
'<tr><td>Mask subnet</td><td><INPUT Type=Text Name="lan.ipV4M" Value="'. $lan[ipV4M].'"></td></tr>\r\n'.
'<tr><td>Getway</td><td><INPUT Type=Text Name="lan.ipV4G" Value="'. $lan[ipV4G].'"></td></tr>\r\n'.
'<tr><td>MAC</td><td><INPUT Type=Text Name="lan.macAddr" Value="'. $lan[macAddr].'"></td></tr>\r\n';
echo '</table></BODY></HTML>';
?>
```

Внешний вид страницы, сгенерированной контроллером на основе PHP скрипта из примера выше.



Запрос параметров контроллера встроенным скриптом pars.php в формате JSON.



4) Исходящие HTTP запросы.

Request.

В платформе RestPLC реализован HTTP клиент, с помощью которого можно обеспечить ситуативную (по расписанию или по событиям от датчиков) отправку HTTP запросов на внешние сервера с необходимым набором данных в форматах HTML, JSON, XML, YAML, обработать ответ содержащий данные управляющие настройками и функциями ПЛК.

Пример запроса на сервер в формате JSON, содержащий состояния выходных портов контроллера:

{"out":{

"0": {

"state":"1",

"currentDuty":"20",

"disableDuty":"0",

"disableChange":"0",

"pwmDuty":"20"

},

"1": {

"state":"0",

"currentDuty":"0",

"disableDuty":"0",

"disableChange":"0",

"pwmDuty":"35"
},
"2": {
"state":"1",
"currentDuty":"50",
"disableDuty":"0",
"disableChange":"0",
"pwmDuty":"50"
},
"3": {
"state":"1",
"currentDuty":"100",
"disableDuty":"1",
"disableChange":"0",
"pwmDuty":"15"
}
}
}

В соответствии с примером видно следующее состояние выходных портов контроллера:

0-й — включен в режиме PWM со скважностью 20%,

1-й — выключен, режим PWM разрешен, настройка скважности 35%,

2-й — включен в режиме PWM со скважностью 50%,

3-й — включен с запрещенным режимом PWM.

Response.

Сервер может ответить на запрос данными в формате JSON для управления параметрами контроллера (вкл/выкл портов, режимы, настройки интерфейсов, внутренняя логика)

Пример ответа сервера в формате JSON, содержащий команды установки состояния выходных портов контроллера:

{"out":{
"0": {
"state":"0"
},
"1": {
"state":"1",
"disableDuty":"1",
"pwmDuty":"100"
},
"2": {
"state":"1",
"disableDuty":"0",
"pwmDuty":"20"
},

"3": {

"state":"0",

}

}

}

Ответ сервера обрабатывается в момент получения и сразу применяются параметры. В данном примере 0-й и 3-й порт выключаем, 1-й включаем и запрещаем PWM, 2-й включаем в режиме PWM со скважностью 20%.

Ответ сервера может содержать произвольные файлы, как и скрипты для управления в автономном режиме.

Ключевой особенностью является удаленная замена алгоритма ПЛК с центрального сервера, что существенно снижает затраты на обслуживание систем автоматизации и повышает вариабельность системы.

Запуск опроса устройств по протоколу Modbus

Чтобы запустить цикл опроса устройств по протоколу Modbus RTU, присоединённых к шине RS-485, необходимо:

в адресной строке браузера ввести IP-адрес ПЛК;

в открывшейся странице в первой строке ввести логин, во второй – пароль, нажать кнопку «Enter»;

в меню «MENU» выбрать пункт «Services»;

активировать клиентскую часть стека TCP/IP через путём выбора параметра «1» пункта «REST client enable»;

в пункте «REST URL» записать адрес удалённого сервера, куда следует отправлять результаты опроса устройства;

нажать кнопку «Save» для применения новых значений параметров;

в меню «MENU» выбрать пункт «Reboot»;

в появившемся контекстном меню выбрать «Reboot»;

дождаться перезапуска ПЛК (состояние работы контролируется по индикатору «Run», который плавно мигает);

в адресной строке браузера ввести ip-address/file_upload.html;

в открывшейся странице нажать кнопку «Browse…» и в проводнике выбрать файл modbus_settings.json;

нажать кнопку «Upload»

После выполненных действий сразу начинается проверка синтаксиса содержимого файла modbus_settings.json. Если результат не противоречит синтаксису JSON, то после проверки выполняется опрос MODBUS-устройств.

Структура файла modbus_settings.json

Файл modbus_settings.json состоит из одного объекта первого уровня «rtu», характеризующий структуру пакета сообщения (ADU – application data unit). Второй уровень состоит из объекта «configuration» и массива объектов «devices». Объект «configuration» не является обязательным и может быть не указан, а массив объектов обязателен.

Объект «configuration»

Объект «configuration» содержит два ключа «periodic_sec» и «baudrate», принимающих целочисленные значения. Ключ «periodic_sec» отвечает за время в секундах ожидания после опроса всех MODBUS-устройств перед тем, как начать опрос сначала. Ключ «baudrate» задаёт скорость передачи данных через последовательный канал связи в бит/с. Если объект «configuration» не задан или не задан какой-либо из его ключей, то принимаются значения по умолчанию, указанные в Таблице 1.

Массив «devices»

Массив «devices» содержит объекты, описывающие передаваемый MODBUS пакет. Любой объект данного массива может содержать ключи «address», «code», «name», «min», «max», «multiplier», «unit», объект «data». Обязательные ключи зависят от кода функции. Структуры объектов для кодов 0x03, 0x06, 0x10 описаны в подразделах 2.3, 2.4 и 2.5.

Ключ «address» задаёт адрес MODBUS-устройства, ключ «code» – код функции, «name» – имя данного устройства, «min» – минимальное значение запрашиваемого значения регистра, «max» – максимальное значение запрашиваемого значения регистра, «multiplier» –коэффициент преобразования запрашиваемого значения регистра, «unit» – единица измерения запрашиваемого значения регистра.

Объект «data» может содержать ключи «starting_address», «registers», «quantity», «value», «address». Ключ «starting_address» задаёт начальный адрес регистра, «registers» – массив значений регистров, «quantity» – количество регистров, «value» – значение регистра, «address» – адрес регистра.

Таблица 1 – Список ключей

№	Ключ	Тип	Принимаемые значения	Значения по умолчанию
объект configuration				
1.	periodic_sec	число	1...4 294 967 295	3
2.	baudrate	число	2400, 4800, 9600, 14400, 19200, 28800, 38400, 56000, 57600, 115200, 128000, 256000	9600
объект data объекта массива devices				
1.	starting_address	число	0...65535	-
2.	registers	массив чисел	0...65535 (8 регистров)	-
3.	quantity	число	0...65535	-
4.	value	число	0...65535	-
5.	address	число	0...65535	-
объект массива devices				
1.	address	число	1...255	-
2.	code	строка	“0x03”, “0x06”, “0x10”	-
3.	name	строка	ASCII печатные символы (максимум 100 символов)	-
4.	min	число	0...65535	-

5.	max	число	0...65535	-
6.	multiplier	строка	ASCII печатные символы (максимум 100 символов)	-
7.	unit	строка	ASCII печатные символы (максимум 100 символов)	-

Пример команды чтения регистров хранения 0x03

```
{
    "rtu": {
        "configuration": {
            "periodic_sec": 3,
            "baudrate": 9600
        },
        "devices": [
            {
                "data": {
                    "starting_address": 64,
                    "quantity": 1
                },
                "address": 3,
                "code":"0x03",
                "name":"PTA8C04",
                "min":1,
                "max":1000,
                "multiplier":"0.1",
                "unit":"degree C"
            }
        ]
    }
}
```

Пример команды записи одного регистра 0x06

```
{
    "rtu": {
        "configuration": {
            "periodic_sec": 3,
            "baudrate": 9600
        },
        "devices": [
            {
                "name":"PTA8C04",
                "code": "0x06",
                "address": 3,
                "data": {
                    "value": 270,
                    "address": 64
                }
            }
        ]
```

```
    }
}
```

Пример команды записи нескольких регистров 0x10

```
{
    "rtu": {
        "configuration": {
            "periodic_sec": 3,
            "baudrate": 9600
        },
        "devices": [
            {
                "code": "0x10",
                "address": 3,
                "data": {
                    "starting_address": 64,
                    "registers":[270, 270, 270]
                },
                "name":"PTA8C04"
            }
        ]
    }
}
```

Формирование ответов

В течение разбора файла *modbus_settings.json* формируются запросы и опрашиваются MODBUS-устройства. Результаты опросов отправляются на сервер в виде JSON-ответов. Пример ответа при удачном опросе для команд чтения регистров хранения (0x03), записи одного регистра (0x06), записи множества регистров (0x10) представлены в подразделах 3.1, 3.2 и 3.3.

В случае возникновения синтаксической ошибки разбора файла *modbus_settings.json* выставляется флаг ошибки и дальнейший разбор файла прекращается. Флаг ошибки синтаксического разбора можно проконтролировать при запросе *pars.php*. При этом ответ на сервер не отправляется для данной ошибки.

Ошибки разбора объекта массива «devices» может происходить в следующих случаях:

1. неполное описание данного объекта, то есть указано недостаточное количество ключей для команды;

2. MODBUS-устройство вернуло код ошибки или вышло время ожидания ответа;

При этих ошибках на сервер отправляется JSON с указанием индекса (ключ «id») объекта в массиве «devices». Пример данного JSON представлен в подразделе 3.4.

Ответ для команды чтения регистров хранения 0x03


```
{
    "version": "1.1a",
    "type": "rtu",
    "app_version": "0.2.0",
    "id_transaction": 1984,
    "host": {
        "name": "atlant",
        "serial_number": "2800-2200-0E513630-34333934",
        "ip_address_v4": "192.168.88.88",
        "mac_address": "00:80:E1:33:39:34"
    },
    "device": {
        "id": 1,
        "name": "PTA8C04",
        "address": 3,
        "code": "0x03",
        "data": {
                "starting_address": 3,
                "quantity": 1
        }
    },
    "response": {
        "error": false,
        "data": {
                "min": 1,
                "max": 1000,
                "multiplier": "1.0",
                "unit": "degree F",
                "binary": {
                        "subtype": 42,
                        "bytes": [1, 195]
                }
        }
    }
}
```


Ответ для команды записи одного регистра 0x06


```
{
    "version": "1.1a",
    "type": "rtu",
    "app_version": "0.2.0",
    "id_transaction": 1984,
    "host": {
```

```
        "name": "atlant",
        "serial_number": "2800-2200-0E513630-34333934",
        "ip_address_v4": "192.168.88.88",
        "mac_address": "00:80:E1:33:39:34"
    },
    "device": {
        "id": 1,
        "name": "PTA8C04",
        "address": 3,
        "code": "0x06",
        "data": {
                "address": 3,
                "value": 1
        }
    },
    "response": {
        "error": false
    }
}
```


Ответ для команды записи нескольких регистров 0x10


```
{
    "version": "1.1a",
    "type": "rtu",
    "app_version": "0.2.0",
    "id_transaction": 1984,
    "host": {
        "name": "atlant",
        "serial_number": "2800-2200-0E513630-34333934",
        "ip_address_v4": "192.168.88.88",
        "mac_address": "00:80:E1:33:39:34"
    },
    "device": {
        "id": 1,
        "name": "PTA8C04",
        "address": 3,
        "code": "0x10",
        "data": {
                "starting_address": 3,
                "quantity": 1
        }
    },
    "response": {
        "error": false
    }
}
```

Ответ при возникновении ошибки


```
{
    "version": "1.1a",
    "type": "rtu",
    "app_version": "0.2.0",
    "id_transaction": 1984,
    "host": {
        "name": "",
        "serial_number": "2800-2200-0E513630-34333934",
        "ip_address_v4": "192.168.88.88",
        "mac_address": "00:80:E1:33:39:34"
    },
    "device": {
        "id": 1
    },
    "response": {
        "error": true
    }
}
```