WEBSOCKET SCADA SERVER

(LOGGER)

руководство по эксплуатации

2019 - 2023

(2.2.x, 2.3.x)

Оглавление

1. ОБЩИЕ СВЕДЕНИЯ

Реализация:

a) среда разработки
   - Qt5

b) язык программирования
   - C++11

Исполнение:

a) консольное, клиент-серверное, многопоточное
b) служба (сервис) ОС

Поддержка:

a) ОС
   - Windows (x86, x64)
   - Linux (x86, x64, armv6, armv7)

b) протоколы физического уровня передачи данных
   - RS-485
   - Ethernet

c) протоколы прикладного уровня передачи данных
   - WebSocket TCP
   - ModBus RTU / TCP
   - DCON

d) СУБД
   - MySQL 5, MariaDB 5

e) структуры данных
   - JSON

Связь с клиентами:

a) Ethernet / WebSocket
b) структура передаваемых данных - JSON
c) клиент подключается и переходит в режим получения данных
d) данные клиенту передаются по готовности (после опроса устройств)
e) клиент может отправлять серверу задание на получение конкретных данных (не всех)
f) клиент может отправлять серверу данные для целевых устройств
g) клиентом может быть «пассивным» (Регистратор сам установит с ним связь / наладит общение, такие клиенты указываются отдельно — в Config["WsCli"]

Связь с устройствами:

a) устройства группируются по сетям
b) сети группируются по протоколам
c) каждое устройство содержит набор регистров
d) чтение данных периодическое
e) запись в устройство при наличии данных от клиента

Архивирование данных:

a) в файл или базу данных
b) два алгоритма (*отдельный поток на каждый алгоритм*):
   - периодический
   - по событию

Настройки:

a) файлы формата JSON

2. ЗАПУСК / ОСТАНОВ

1) Работает в качестве службы ОС
2) Управление (запуск/останов)
   - ОС Linux: systemd
   - ОС Windows: диспетчер задач
3) Аргументы исполняемого файла

```
--config PathToConfigFile [--sname ServiceName --sdesc ServiceDescription]
```

где,
PathToConfigFile - путь к файлу Основных настроек
ServiceName - имя службы (необязательное)
ServiceDescription - описание службы (необязательное)

Пример:

```
--config E:\wsscada.conf.d\config.json --sname wsSCADA --sdesc wsSCADA
```

3. НАСТРОЙКИ

3.1 Основные настройки (Config)

Файл описывает основные настройки Регистратора:

- один ассоциативный массив

Пример файла настроек (config.json):

```
{ "ID": "wsSCADA",
  "Port": 8100,
  "ConnMax": 30,
  "ConnPerCli": 3,
  "ConnLifeTime": 28800,
  "SurveyDelay": 1000,
  "FirstSurveyNow": 1,
  "Random": 0,
  "UseLog": 1,
  "UseWs": 1,
  "UseWsCli": 1,
  "UseWsBlack": 1,
  "UseArh": 1,
  "UseEvent": 1,
  "WsBlack": "E:\\wsscada.conf.d\\wsblack.json",
  "WsCli": "E:\\wsscada.conf.d\\wscli.json",
  "Roles": "E:\\wsscada.conf.d\\roles.json",
  "Arh": "E:\\wsscada.conf.d\\arh.json",
  "Networks": "E:\\wsscada.conf.d\\networks.json",
  "Log": "E:\\wsscada.conf.d\\wsscada.log"
}
```

Пути к файлам для ОС Linux:

```
  "WsBlack": "/usr/local/etc/wsscada.conf.d/wsblack.json",
  "WsCli": "/usr/local/etc/wsscada.conf.d/wscli.json",
  "Roles": "/usr/local/etc/wsscada.conf.d/roles.json",
  "Arh": "/usr/local/etc/wsscada.conf.d/arh.json",
  "Networks": "/usr/local/etc/wsscada.conf.d/networks.json",
  "Log": "/usr/local/etc/wsscada.conf.d/wsscada.log"
```

Аргументы ассоциативного массива следующие:

Аргумент	Тип данных	Значение	Описание
"ID"	строка	любая строка	ID сервера = «» (по-умолчанию)
"Port"	число	0 … 65535	Номер сетевого порта WebSocket-сервера = 8100 (по-умолчанию) * для подключения клиентов * клиент только подключается и ожидает данных от сервера (запрос не отправляет) или отправляет данные серверу для их записи в устройство * сервер, по готовности, отправляет данные подключенным клиентам

Аргумент	Тип данных	Значение	Описание
"ConMax"	число	0 … 250	Максимальное количество подключений (клиентов) (с учетом значения "ConnPerCli") = 0 (по-умолчанию)
"ConnPerCli"	число	0 … 250	Количество подключений на одного клиента = 0 (по-умолчанию)
"ConnLifeTime"	число	-1 … 65000	Длительность клиентского подключения, сек = -1 - без ограничений (по-умолчанию)
"SurveyDelay"	число	300 … 65000	Пауза между сеансами опроса, мсек 300 (по-умолчанию)
"FirstSurveyNow"	число	0 1	Первый сеанс чтения данных без задержки - выключено (по-умолчанию) - включено
"Random"	число	0 1	Алгоритм случайных значений - выключено (по-умолчанию) - включено
"UseLog"	число	0 1	Использовать запись журнала работы процесса сбора данных - выключено (по-умолчанию) - включено
"UseWs"	число	0 1	Использовать WebSocket-сервер - выключено (по-умолчанию) - включено
"UseWsCli"	число	0 1	Использовать связь с WebSocket-клиентами - выключено (по-умолчанию) - включено
"UseWsBlack"	число	0 1	Использовать «черный-список» - выключено (по-умолчанию) - включено
"UseArh"	число	0 1	Использовать процесс периодического архивирования в базу данных - выключено (по-умолчанию) - включено
"UseEvent"	число	0 1	Использовать событийное архивирование в базу данных - выключено (по-умолчанию) - включено
"WsBlack"	строка	путь к файлу формата JSON или пусто	Файл «черный-список» = «» (по-умолчанию)
"WsCli"	строка	путь к файлу формата JSON или пусто	Файл со списком WebSocket-клиентов = «» (по-умолчанию)

Аргумент	Тип данных	Значение	Описание
"Roles"	строка	путь к файлу формата JSON или пусто	Файл со списком ролей пользователей = «» (по-умолчанию)
"Arh"	строка	путь к файлу формата JSON или пусто	Файл настроек архивирования = «» (по-умолчанию)
"Networks"	строка	путь к файлу формата JSON или пусто	Файл со списком настроек сетей = «» (по-умолчанию)
"Log"	строка	путь к текстовому файлу	Файл журнала работы процесса сбора данных = «» (по-умолчанию)

Архивирование данных

- файл или база данных
- *задается индивидуально для каждого устройства*
  файл описания устройств - Dev["ArhFile"] или Dev["ArhTable"]
- выполняется периодически и/или по событию

Настройки архивирования:

- подключение к СУБД
- период архивирования
  задаются в отдельном файле (путь к файлу настроек задается через аргумент "Arh")

Архивирования выполняется в отдельном потоке:

- периодическое архивирование отдельно
- архивирование событийное отдельно

Настройки событий задаются для каждого регистра устройства отдельно
в файле описания регистров — Reg["Events"]

3.2 «Черный-список» клиентов (WsBlack)

Файл содержит список IP-адресов клиентов, которым запрещен доступ к Регистратору.
Список адресов представлен одним массивом (списком) строк.

Формат одной записи (строки):
`"IP[:Port]"`

Пример файла (wsblack.json):

```
[ "192.168.1.77:0",
  "127.0.0.1:1234"
]
```

Здесь:
> клиент с IP 192.168.1.77 блокируется по всем портам
> клиент с IP 127.0.0.1 блокируется только по порту 1234

Запись поддерживает следующие аргументы:

Аргумент	Тип данных	Значение	Описание
IP	строка	строка IP-адреса	IP-адрес блокируемого клиента (подключения)
Port	число	0 … 65535	Номер сетевого порта клиента (подключения) = 0 — любой порт (по-умолчанию)

3.3 WebSocket-клиенты (WsCli)

Файл содержит список WebSocket-клиентов, которым Регистратор должен самостоятельно установить связь и передавать данные.

Список клиентов представлен в виде одного массива строк.

Формат одной записи (строка):
```
"ws://IP-or-HostName:Port"
```

Это необходимо тогда, когда, например:

- Регистратор находится в «серой зоне» (имеет свободный доступ в сеть Интернет, но из сети к нему доступа нет по причине отсутствия «белого IP-адреса»)
- WebSocket-клиент находится в «белой зоне» (имеет свободный досутп в сеть Интернет и к нему есть доступ из сети в виду наличия «белого IP-адреса»)

Пример файла (wscli.json):

```
["ws://at09.ddns.net:8100",
 "ws://wsrepeater.ddns.net:18000"
]
```

Запись поддерживает следующие аргументы:

Аргумент	Тип данных	Значение	Описание
ws	строка	протокол передачи данных	Код протокола передачи данных = ws (константа)
IP-or-HostName	строка	строка IP-адреса	IP-адрес или хост-имя WebSocket-клиента
Port	число	1 … 65535	Номер сетевого порта WebSocket-клиента

3.4 Роли пользователей (Roles)

Файл содержит список ролей клиентов. Роль ряд ограничений/разрешений для клиента в части общения с Регистратором. Если клиента нет в списке, то ему назначается роль «по-умолчанию» (viewer). Функционал ролей клиентов на данный момент не поддерживается Регистратором.

Роли пользователей представлены в виде одного ассоциативного массива.

Формат одной записи (ассоциативный массив):
```
{ "IP":"IP-address", "Role":"user-role" }
```

Пример файла (roles.json):

```
[{"IP":"192.168.1.220", "Role":"viewer"},
 {"IP":"192.168.1.77", "Role":"manager"},
 {"IP":"127.0.0.1", "Role":"manager"}
]
```

Ассоциативный массив поддерживает следующие аргументы:

Аргумент	Тип данных	Значение	Описание
"IP"	строка	строка IP-адреса	IP-адрес клиента-пользователя
"Role"	строка	"viewer" "manager"	Роль клиента-пользователя - только получение данных - получение данных и отправка запросов (управление)

3.5 Архивирование (Arh)

Файл содержит настройки архивирования: тип архива, подключение к базе данных, профиль архивирования и т. п.

Настройки архивирования представлены в виде одного ассоциативного массива.

Пример файла (arh.json):

```
{"Host": "192.168.1.220",
 "Port": "3306",
 "User": "login",
 "Passwd": "password",
 "Db": "arh1",
 "Profile": "5min",
 "UseLog": 1,
 "UseLogEvent": 1,
 "Log": "E:\\wsscada.conf.d\\arh.log",
 "LogEvent": "E:\\wsscada.conf.d\\arh-ev.log"
}
```

Пути к файлам для ОС Linux:

```
  "Log": "/var/log/wsscada/arh.log",
  "LogEvent": "/var/log/wsscada/arh-ev.log"
```

Ассоциативный массив поддерживает следующие аргументы:

Аргумент	Тип данных	Значение	Описание
"Host"	строка	строка IP-адреса или Хост-имени	IP-адрес или Хост-имя СУБД = «localhost» (по-умолчанию)
"Port"	число	1 … 65535	Номер сетевого порта СУБД = 3306 (по-умолчанию)
"User"	строка	любая строка	Имя пользователя СУБД = «» (по-умолчанию)
"Passwd"	строка	любая строка	Пароль пользователя СУБД = «» (по-умолчанию)

Аргумент	Тип данных	Значение	Описание
"Db"	строка	любая строка	Имя базы данных = «» (по-умолчанию)
"Profile"	строка	«1min» «5min» «15min» «30min» «hour» «event»	Профиль периодичности архивирования = «» (по-умолчанию) - 1 минута - «5min» - 5 минут - «15min» - 15 минута - «30min» - 30 минут - «hour» - 1 час - «event» - по событию
"UseLog"	число	0 1	Использовать запись журнала работы процесса периодического архивирования - выключено (по-умолчанию) - включено
"UseLogEvent"	число	0 1	Использовать запись журнала работы процесса архивирования по событию - выключено (по-умолчанию) - включено
"Log"	строка	путь к текстовому файлу	Файл журнала работы процесса периодического архивирования = «» (по-умолчанию)
"LogEvent"	строка	путь к текстовому файлу	Файл журнала работы процесса архивирования по событию = «» (по-умолчанию)

3.6 Сети (Net)

Файл содержит список (один массив) настроек сетей связи. Каждая отдельная сеть представлена ассоциативным массивом. За каждой сетью закреплено целевое оборудование, с которым Регистратор будет взаимодействовать.

Пример файла (networks.json):

```
[ {"ID": 1,
   "ProtoComm": "eth",
   "ProtoData": "modbus_tcp",
   "Allow": 1,
   "Devices": "E:\\wsscada.conf.d\\devices.eth.json"
   },
  {"ID": 2,
   "ProtoComm": "serial",
   "ProtoData": "modbus_rtu",
   "Port": 9,
   "Spd": 9600,
   "Prty": "None",
   "DataBits": 8,
   "StopBits": 1,
   "Mode": "RS485",
   "Allow": 1,
   "Devices": "E:\\wsscada.conf.d\\devices.rs485.json"
   },
   {"ID": 3,
```

```
    "ProtoComm": "dummy",
    "Allow": 1,
    "Devices": "E:\\wsscada.conf.d\\devices.dummy.json"
    }
]
```

Пути к файлам для ОС Linux:

```
  "Devices": "/usr/local/etc/wsscada.conf.d/devices.eth.json"
  "Devices": "/usr/local/etc/wsscada.conf.d/devices.rs485.json"
  "Devices": "/usr/local/etc/wsscada.conf.d/devices.dummy.json"
```

Ассоциативный массив с настройками одной сети поддерживает следующие аргументы:

Аргумент	**Тип данных**	**Значение**	**Описание**
"ID"	число	0 … 65535	ID сети = 0 (по-умолчанию)
"ProtoComm"	строка	"serial" "eth" "dummy"	Протокол физического уровня передачи данных (коммуникационный интерфейс) - последовательный интерфейс (COM, UART, COM через USB) - ethernet (TCP/IP) - пустышка"
"ProtoData"	строка	"modbus_rtu" "modbus_tcp" "dcon"	Протокол прикладного уровня передачи данных: - для Net["ProtoComm"]: "serial" - для Net["ProtoComm"]: "eth" - для Net["ProtoComm"]: "dcon"
"Port"	число	0 … 250	Номер последовательного порта для Net["ProtoComm"]: "serial" = 0 (по-умолчанию)
"PortPref"	строка	"COM" "/dev/ttyUSB" "/dev/ttyS"	Префикс имени последовательного порта Net["ProtoComm"]: "serial" - для ОС Windows - для ОС Linux - для ОС Linux
"PortDev"	строка	"COMn" если n > 9, то "\\\\.\\COMn" "/dev/ttyUSBn" "/dev/ttySn" где, n от 0 до 250	Полное имя последовательного порта Net["ProtoComm"]: "serial" - для ОС Windows - для ОС Linux - для ОС Linux
"Spd"	число	2400 4800 9600 14400	Скорость последовательного порта Net["ProtoComm"]: "serial" - (по-умолчанию)

Аргумент	Тип данных	Значение	Описание
		19200 28800 38800 57600 115200	
"Prty"	строка	"None" "Even" "Odd"	Четность последовательного порта Net["ProtoComm"]: "serial" - (по-умолчанию)
"DataBits"	число	8 7	Количество бит данных последовательного порта Net["ProtoComm"]: "serial" - (по-умолчанию)
"StopBits"	число	1 2	Количество стоп-бит последовательного порта Net["ProtoComm"]: "serial" - (по-умолчанию)
"Mode"	строка	"RS232" "RS485"	Режим работы последовательного порта Net["ProtoComm"]: "serial" - (по-умолчанию)
"Allow"	число	0 1	Разрешение на работу с сетью - выключено (по-умолчанию) - включено
"Devices"	строка	путь к файлу формата JSON или пусто	Файл со списком настроек устройств одной сети = «» (по-умолчанию)

ProtoComm == "dummy" используется, если:

- значения регистров:
  - случайные (Dev.Reg["Random"])
  - предустановленный константы (Dev.Reg["Value"])
- рекомендуется все виртуальные регистры (со случайными или предустановленными значениями) привязывать к отдельной сети суказанием протокола физического уровня данного типа, класс устройства (Dev"Class"]) при этом не учитывается (можно указывать как "dummy")

Если определен аргумент PortPref, то имя последовательного порта в операционной системе формируется следующим образом: `"PortPref" + "Port"`

Если определен аргумент PortDev, то его значение будет использоваться в качестве имени последовательного порта в операционной системе (значения аргументов Port и PortPref будут проигнорированы).

3.7 Устройства (Dev)

Файл содержит список устройств, запрепленных за определенной сетью. Для каждой сети свой файл со списком устройств (см. выше описание Net). Для каждого устройства закреплен свой список регистров.

Пример файла devices.eth.json:

```
[ {"ID": 1,
   "Class": "plc",
   "BaseAddr": 1,
   "IP": "192.168.1.197",
   "Port": 502,
   "Allow": 1,
   "Registers": "E:\\wsscada.conf.d\\reg.plc1.l2n.json"
   "ArhTable": "arh_plc1_l2n"
   }
]
```

Пути к файлам для ОС Linux:

```
  "Registers": "/usr/local/etc/wsscada.conf.d/reg.plc1.l2n.json"
```

Пример файла devices.rs485.json:

```
[ {"ID": 1,
   "Class": "plc",
   "BaseAddr": 1,
   "Allow": 1,
   "Registers": "E:\\wsscada.conf.d\\reg.plc1.water.json",
   "ArhFile": "E:\\wsscada.conf.d\\arh_plc1_water.txt"
   }
]
```

Пути к файлам для ОС Linux:

```
  "Registers": "/usr/local/etc/wsscada.conf.d/reg.plc1.water.json"
    "ArhFile": "/usr/local/etc/wsscada.conf.d/arh_plc1_water.txt"
```

Пример файла devices.dummy.json:

```
[ {"ID": 1,
   "Class": "dummy",
   "Allow": 1,
   "Registers": "E:\\wsscada.conf.d\\reg.dummy1.json"
   },
  {"ID": 2,
   "Class": "dummy",
   "Allow": 1,
   "Registers": "E:\\wsscada.conf.d\\reg.dummy2.json"
   }
]
```

Ассоциативный массив с настройками одного устройства поддерживает следующие аргументы:

Аргумент	Тип данных	Значение	Описание
"ID"	число	0 … 65535	ID устройства = 0 (по-умолчанию)

Аргумент	Тип данных	Значение	Описание
"Class"	строка	любая строка "i7041"	Класс устройства: - «» (по-умолчанию) - для Net["ProtoData"]: "dcon"
"BaseAddr"	число	0 … 250	Базовый адрес устройства в сети: = 0 (по-умолчанию)
"IP"	строка	строка IP-адреса	IP-адрес устройства для Net["ProtoComm"]: "eth" = «» (по-умолчанию)
"Port"	число	0 … 65535	Номер сетевого порта устройства для Net["ProtoComm"]: "eth" = 0 (по-умолчанию)
"Allow"	число	0 1	Разрешение на работу с устройством - выключено (по-умолчанию) - включено
"Registers"	строка	путь к файлу формата JSON или пусто	Файл со списком настроек регистров устройства = «» (по-умолчанию)
"Checksum"	строка	0 1	Использовать вычисление контрольной суммы для Net["ProtoData"]: "dcon" - выключено (по-умолчанию) - включено
"WaitRead"	число	0 … 65535	Время ожидания чтения из последовательного порта (мсек) для Net["ProtoData"]: "dcon" или "modbus_rtu" = 1000 (по-умолчанию)
"Reconnect"	число	0 1	Открывать и закрывать связь с последовательным портом устройства после чтения/записи одной группы регистров для Net["ProtoData"]: "modbus_rtu" - выключено (по-умолчанию) - включено
"ArhTable"	строка	значение в формате SQL	Имя таблицы БД для архивирования показателей устройства = «» (по-умолчанию)
"ArhFile"	строка	путь к текстовому файлу	Файл для хранения показателей устройства = пусто (по-умолчанию)

Если определен аргумент «ArhTable» и заданы настройки подключения к СУБД, то показатели устройства будут архивироваться в таблицу БД.

Если определен аргумент «ArhFile», то показатели устройства будут архивироваться в файл.

Если определен аргумент «ArhTable» и «ArhFile», то показатели устройства будут архивироваться в файл.

Если аргументы «ArhTable» и/или «ArhFile» не определены, то архивирование выполняться не будет.

Если Config["UseArh"] == 0, то архивирование отключено (глобально) для всех устройств.

Пример SQL-запроса на создание таблицы БД:

```
DROP TABLE IF EXISTS `arh_plc1_l2n`;
CREATE TABLE `arh_plc1_l2n` (
  `time_stamp` timestamp NOT NULL DEFAULT CURRENT_TIMESTAMP ON UPDATE
CURRENT_TIMESTAMP, -- Автоматическая временная метка
  `stamp` datetime DEFAULT NULL, -- Дата и время записи
  `profile` ENUM( -- Профиль
                '1min',  -- 1 минута
                '5min',  -- 5 минут
                '15min', -- 15 минут
                '30min', -- 30 минут
                'hour',  -- час
                'event'  -- событие
                ),
  `device_id` bigint(20) unsigned DEFAULT NULL, -- ID устройства
  `register_id` bigint(20) unsigned DEFAULT NULL, -- ID регистра
  `value` float(10,2) NULL DEFAULT NULL, -- значение регистра
  `ex` int(11) NULL DEFAULT NULL, -- код исключения (0 — нет)
  `err` int(11) NULL DEFAULT NULL, -- код ошибки (0 — нет)
  `sign` int(11) NULL DEFAULT NULL, -- код признака (0 — нет)
  KEY `time_stamp_k` (`time_stamp`),
  KEY `stamp_k` (`stamp`),
  KEY `profile_k` (`profile`),
  KEY `ex_k` (`ex`),
  KEY `err_k` (`err`),
  KEY `sign_k` (`sign`),
  KEY `device_id_k` (`device_id`),
  KEY `register_id_k` (`register_id`)
) ENGINE=InnoDB DEFAULT CHARSET=utf8;
```

Пример SQL-запроса на вставку данных (генерируется Регистратором):

```
INSERT INTO `proc_l2n`
 (`stamp`, `profile`, `device_id`, `register_id`, `value`, `ex`, `err`, `sign`)
VALUES
 ('2019-03-21 11:13:00', '15min', 1, 26, 3, NULL, NULL, NULL),
 ('2019-03-21 11:13:00', '15min', 1, 27, 1, NULL, NULL, NULL),
 ('2019-03-21 11:13:00', '15min', 1, 32, 2, NULL, NULL, NULL);
```

Значение поля `value` отформатировано по:

- Dev.Reg["Type"]
- Dev.Reg["Offset"]
- Dev.Reg["Round"]

3.8 Регистры данных (Reg)

Файл содержит один массив со списком настроек регистров для одного конкретного устройства. Один регистр описывается ассоциативным массивом.

Пример файла reg.plc1.l2n.json:

```
[ {"ID": 1,
   "Addr": 29,
```

```
    "Class": "in",
    "Type": "int",
    "Var": "Mo",
    "Allow": "r|w|h|a"
    },
   {"ID": 2,
    "Addr": 62,
    "Class": "in",
    "Type": "int",
    "Var": "Lv",
    "Allow": "r|h|a",
    "Events": "E:\\wsscada.conf.d\\ev.plc1.l2n.lv.json"
    },
   {"ID": 3,
    "Addr": 200,
    "Class": "in",
    "Type": "word",
    "Var": "Plc",
    "Allow": "r|h|d",
    "RandMin": 0,
    "RandMax": 1000
    },
   {"ID": 4,
    "Addr": 210,
    "Class": "in",
    "Type": "word",
    "Var": "PsPls",
    "Allow": "r|h|a",
    "Events": "E:\\wsscada.conf.d\\ev.plc1.l2n.pspls.json"
    },
   {"ID": 5,
    "Addr": 211,
    "Class": "in",
    "Type": "word",
    "Var": "Wo1",
    "Allow": "r"
    },
   {"ID": 6,
    "Addr": 212,
    "Class": "in",
    "Type": "word",
    "Var": "Wo2",
    "Allow": "r"
    },
   {"ID": 7,
    "Class": "calc",
    "Type": "int",
    "Var": "Wo",
    "Allow": "r|h",
    "Targets": [5, 6],
    "Alg": "Merge32"
    }
]
```

Ассоциативный массив с настройками одного регистра поддерживает следующие аргументы:

Аргумент	**Тип данных**	**Значение**	**Описание**
"ID"	число	0 … 65535	ID регистра = 0 (по-умолчанию)
"Addr"	число	0 … 65535	Адрес сетевого регистра

Аргумент	Тип данных	Значение	Описание
			= 0 (по-умолчанию) *для: Net["ProtoData"]: "modbus_rtu", "modbus_tcp"* *не для: Reg["Class"]: "const", "random", "calc"*
"Class"	строка	"const"	Класс регистра: - константа *для: Net["ProtoData"]: "dummy"* *требуется: Reg["Value"]*
		"random"	- случайное значение *для: Net["ProtoData"]: "dummy"* *требуется: Reg["RandMin"], Reg["RandMax"]*
		"calc"	- виртуальный (вычисленный) регистр *требуется: Reg["Targets"]*
		"coil"	- coil registers *для: Net["ProtoData"]: "modbus_rtu", "modbus_tcp"*
		"disc"	- dicrete input registers *для: Net["ProtoData"]: "modbus_rtu", "modbus_tcp"*
		"in", "inpt"	- input registers *для: Net["ProtoData"]: "modbus_rtu", "modbus_tcp"*
		"hold"	- holding registers *для: Net["ProtoData"]: "modbus_rtu", "modbus_tcp"*
		"6"	- слово-состояний дискретных входов/выходов модулей i7000 *для: Net["ProtoData"]: "dcon"*
"Type"	строка	"bool" "word" "int" "float" "raw"	Тип выходного значения - boolean - unsigned int (16 bit) - signed int (32 bit) - float - исходное значение (по-умолчанию)
"Offset"	число	-1, 0 > 0	Выполнить кодирование/декодирование смещения дробной точки вправо: - нет смещения (по-умолчанию) - величина смещения *для: Reg["Type"]: "float"* *или* *Reg["Class"]: not "calc"* *декодирование = (прочитано / Offset)*

Аргумент	Тип данных	Значение	Описание
			для: Reg["Class"]: "calc" *Reg["Alg"]: "float"* *Reg["Type"]: "float"* *кодирование = (прочитано * Offset)*
"Round"	число	-1 0 >0	Выполнить округление - нет округления (по-умолчанию) - округление до целого - округление до заданного количества знаков после дробной точки *для Reg["Type"]: "float"* *результат = round(прочитано, Round)*
"Var"	строка	любая строка	Имя переменной (как будет передаваться клиенту) = "" (по-умолчанию)
"Allow"	строка	"r" "w" "h" "a" "d"	Маска разрешений - чтение значения с устройства, вычисления - запись значения в устройство - отправка значения клиенту - архивирование - генерирование случайного значения * несколько значений разделяются символом |
"RandMin"	число	0 … 65535	Минимальное значение генератора случайных чисел = 0 (по-умолчанию) *для: Reg["Class"]: "random"*
"RandMax"	число	0 … 65535	Максимальное значение генератора случайных чисел = 0 (по-умолчанию) *для: Reg["Class"]: "random"*
"Value"	число	любое число	Предустановленное число (константа) *для: Reg["Class"]: "const"*
"Events"	строка	путь к файлу формата JSON или пусто	Путь к файлу со списком настроек событий для архивирования = «» (по-умолчанию) *для: Config["UseArh"]: 1*
"Targets"	массив	[ID, ...]	Список идентификаторов регистров для вычислений (преобразований) *для: Reg["Class"]: "calc"* требуется: *Reg["ID"], Reg["Alg"]* = [] (пусто, по-умолчанию)
"Alg"	строка	"merge32" "dword" "float" "bitstodword"	Алгоритм вычисления значения виртуального регистра = «» (по-умолчанию) - объединение в DWORD - объединение в FLOAT - объединение бит в DWORD *для: Reg["Class"]: "calc", Reg["Targets"]*

Исходное значение, передаваемое по протоколу ModBus, имеет тип uint16 (беззнаковое целое, 16 бит — слово). Целочисленные значения со знаком кодируются по алгоритму дополнительного кода. Числа с плавающей точкой кодируются алгоритмом IEEE754 или путем смещения точки вправо на значение «Offset», а декодируются путем деления исходного значения на значение смещения (при данном способе кодирования теряется точность). Для достижения точности, число с плавающей точкой хранится в таблицах в виде нескольких словных регистров (для 4-х байтных чисел — 2 слова), передаются в определенном порядке (например, младшее слово вперед, затем старшее) и «склеиваются» получателем в одно число типа «float» - это определяется аргументами «Class» (= «calc»), «Targets» (где указываются исходные регистры), «Alg» («Float»).

Маска разрешений задается одной строкой, для разделения значений используется символ «|» (ИЛИ), например: `"r|w|h|a|d"`

Алгоритмы объединения: "Merge32", "Dword" и "Float" - требуют определения массива исходных регистров в массиве «Targets» и используют его значения в следующем порядке:
[Targets][0] — младшее слово,
[Targets][1] — старшее слово.

Алгоритм объединения "BitsToDword" требует «Targets» и использует его значения в следующем порядке:
бит 0 = 1, если [Targets][0] > 0, иначе = 0,
бит 1 = 1, если [Targets][1] > 0, иначе = 0,
…
бит 31 = 1, если [Targets][31] > 0, иначе = 0.

Пример настроек регистров для получения числа типа Float по протоколу ModBus RTU (предположительно, устройство хранит число типа Float в виде двух слов — 4 байта в таблице Input Registers: младшее слово по адресу 0, старшее слово по адресу 1, передача идет младшим словом вперед):

```
[ {"ID": 1,
   "Addr": 0,
   "Class": "inpt",
   "Type": "word",
   "Var": "FloLo",
   "Allow": "r"
   },
  {"ID": 2,
   "Addr": 1,
   "Class": "inpt",
   "Type": "word",
   "Var": "FloHi",
   "Allow": "r"
   },
  {"ID": 3,
   "Class": "calc",
   "Type": "float",
   "Var": "Flo",
   "Allow": "r|h",
   "Targets": [0, 1],
   "Alg": "float"
   }
]
```

3.9 События для регистров данных (Reg.Ev)

Файл содержит один массив со списком настроек событий для одного регистра данных. Одно событие описывается ассоциативным массивом.

Пример файла (ev.plc1.l2n.lv.json):

```
[ {"Alg": "=Ref",
   "Ref": 0,
   "Sign": 1
   },
  {"Alg": ">Ref",
   "Ref": 40,
   "Err": 1
   }
]
```

Пример файла (ev.plc1.l2n.pspls.json):

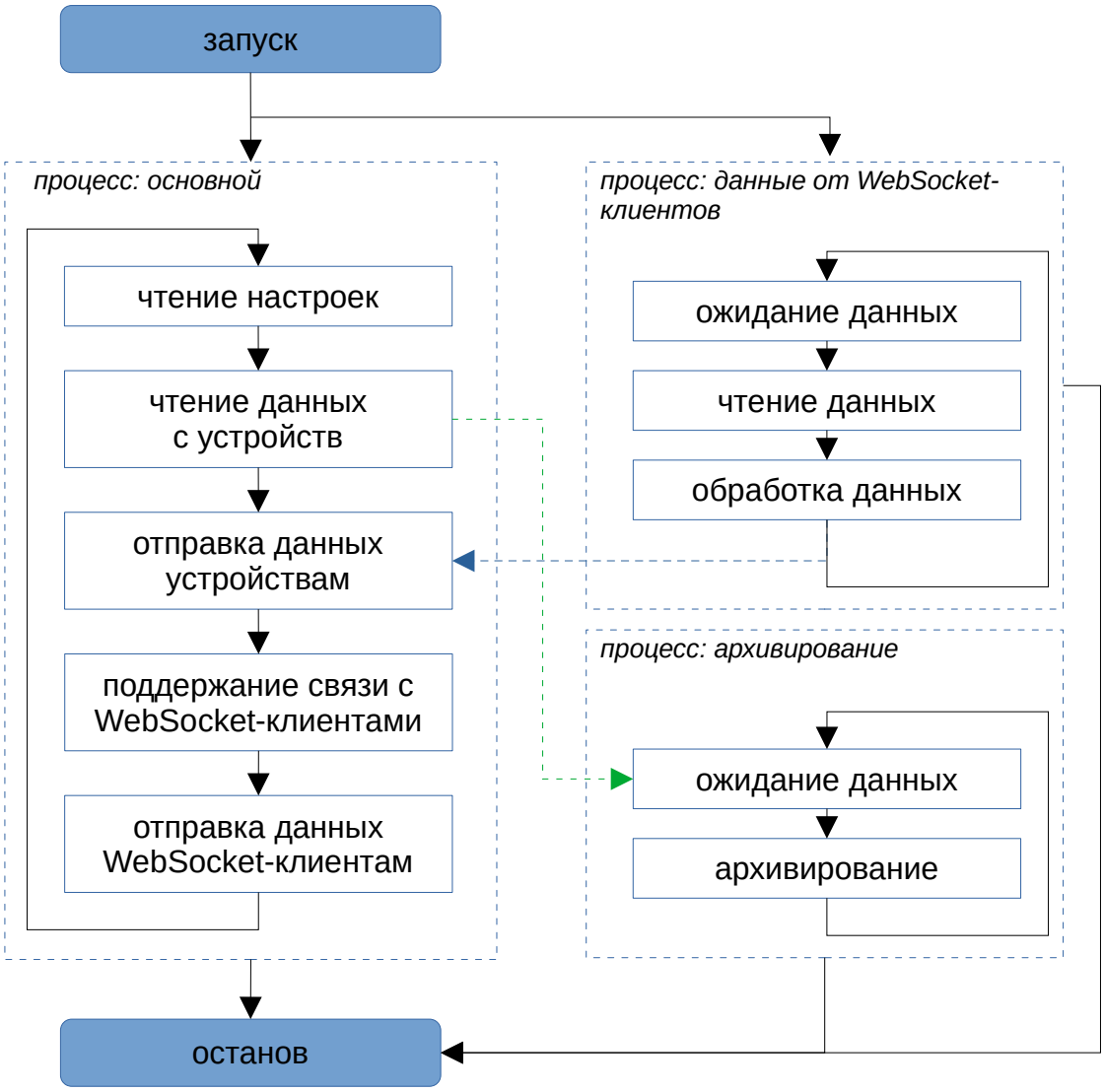
```
[ {"Alg": "OutLim",
   "Min": 0,
   "Max": 3000,
   "Err": 2
   },
  {"Alg": "=Ref|<Ref",
   "Ref": 500,
   "Sign": 2
   },
  {"Alg": "=Ref|>Ref",
   "Ref": 2500,
   "Sign": 3
   }
]
```

Ассоциативный массив с настройками одного события поддерживает следующие аргументы:

Аргумент	**Тип данных**	**Значение**	**Описание**
"Alg"	строка	"OutLim" "=Ref" ">Ref" "<Ref"	Алгоритм (или список алгоритмов) фиксации события. Список алгоритмов задается в одной строке, используя символ разделения `|` - выход за пределы допустимого значения Min и Max (требуются: "Min", "Max") - равно уставке Ref (требуется "Ref") - больше уставки Ref (требуется "Ref") - меньше уставки Ref (требуется "Ref")
"Min"	число	-32768 … 32767	Минимальное значение (для Reg.Ev["Alg"])
"Max"	число	-32768 … 32767	Максимальное значение (для Reg.Ev["Alg"])
"Ref"	число	-32768 … 32767	Уставка (для Reg.Ev["Alg"] = "=Ref|>Ref|<Ref")

Аргумент	Тип данных	Значение	Описание
"Sign"	число	-32768 … 32767	Числовой код признака, соответствующий возникшему событию (если требуется)
"Err"	число	-32768 … 32767	Числовой код ошибки, соответствующий возникшему событию (если требуется)
"Ex"	число	-32768 … 32767	Числовой код исключительной ситуации, соответствующий возникшему событию (если требуется)

4. ЦИКЛОГРАММА РАБОТЫ



5. СТРУКТУРА ПАКЕТА ДАННЫХ

5.1 Данные от сервера

Формат пакета: JSON
Пакет содержит данные, собранные сервером с устройств, определеных конфигурацией.

```
{       "ID":(Config["ID"]),
  "Networks":[ {
                       "ID":(Net["ID"]),
                "Devices":[ {
                                      "ID":(Dev["ID"]),
                                 "BaseAddr":(Dev["BaseAddr"]),
                                    "Class":(Dev["Class"]),
                          "(Dev.Reg["Var"])":(RegisterValue),
                                     ...
                                  },
                                  ...
                               ]
              },
              ...
            ],
     "Stamp":(UnixTimeStamp)
}
```

5.2 Данные от клиента

Формат пакета: JSON
Пакет содержит данные, которые сервер должен передать определенным устройствам.

```
{  "SrvID":(Config.ID),
   "NetID":(Network[n].ID),
   "DevID":(Network[n].Device[d].ID),
    "Data":{RegisterVar:RegisterValue, ...},
   "Stamp":(UnixTimeStamp)
}
```