

Типы Сервера телеметрии.

В данном документе рассматриваются наиболее употребляемые типы, их структура и назначение.

1. Типы, используемые для подключения приборов.

Для организации связи между параметром сервера телеметрии и измеренным значением на приборе используется связка типов: <Port>-<Device>-<Obj>.

В описании используется термины:

- УСД - устройства сбора данных. Это один или группа контроллеров (приборов), подключенные, как правило, к технологической сети через конвертор Ethernet – RS485(232) (например ADAM – 4579);
- драйвер – программный модуль, основной задачей которого является приведение специфического интерфейса УСВ к унифицированному интерфейсу опроса приборов (см. соответствующую документацию) сервера телеметрии;
- модуль опроса – логическая часть сервера телеметрии, основной задачей которого является сбор информации с контролируемых УСД (подробнее в соответствующей документации).

Тип «**Port**» - в этом типе описывается, сколько и какие драйвера должны быть запущены в процессе работы сервера телеметрии. Каждый объект типа описывает один экземпляр драйвера. Информация является входной для модуля опроса.

Тип «**Device**» - в этом типе описывается набор устройств (приборов), с которых необходимо запрашивать данные. Каждый объект типа описывает одно устройство. Эта информация так же является входной для модуля опроса.

Тип <Port>.

Глубина хранения - 1. Объекты типа **Port** должны содержать следующие поля (параметры), описывающие режимы взаимодействия модуля опроса с драйверами:

- **PortData** – здесь указывается как именно драйвер должен соединяться с контролируемым УСД и параметры этого соединения.

В случае, когда драйвер подключается к УСД через Com-порт, поле PortData должно содержать параметры этого соединения (а именно: имя Com-порта; скорость обмена по Com-порту в бодах; количество бит в байте; четность; количество стоповых бит), перечисленных через запятую. Например: /dev/this0,1200,8,n,1.

В случае, когда драйвер подключается к УСД посредством TCP/IP сокета, указываются IP-адрес и IP-порт этого сокетного соединения через двоеточие. Например: 10.0.9.34:5201.

- **tOutRequest** – определяет таймаут ожидания модулем опроса ответа от драйверного модуля на каждый отправляемый запрос.

Величина таймаута должна устанавливаться в зависимости от конкретного УСД и условий его функционирования. В качестве рекомендации можно указать следующее: таймаут должен быть как минимум в полтора раза больше, чем максимально возможное время ответа УСД на запрос драйвера.

Так же на значение поля налагается ограничение модулем опроса: $tOutRequest * 2.5 \leq tOutTest$. Такое ограничение достаточно условно и введено для того, чтобы тестовые пакеты (пакеты, проверяющие работоспособность соединения сервера телеметрии с драйвером) «ходили» не чаще, чем полезные запросы к драйверу.

В процессе обмена информацией между модулем обмена и драйвером, последний играет ведомую роль, и данный таймаут обязывает драйвер ответить до его (таймаута) истечения, даже в случае незавершенного опроса УСД. Если драйвер не успевает ответить (ответа нет более tOutRequest мск), модуль опроса рвет соединение с драйверным модулем. Для драйверного модуля разрыв соединения с сервером телеметрии является безусловной причиной остановки.

- **Parallel** – признак параллельности опроса драйвером контролируемых УСД.

Если драйвер работает с одним УСД, значение флага не влияет на режим опроса УСД. В этом случае рекомендуется выставить значение 1.

Если драйвер работает с несколькими УСД, то:

- Значение флага должно быть равно 1, если параллельный опрос УСД возможен (напр. приборы включены в общую шину RS-485);

- Значение флага должно быть равно 0, если параллельный опрос УСД не возможен (напр. каждый прибор соединен с драйвером через медленный модем, и опрос одного УСД невозможен, пока не завершен опрос другого).

- **tOutTest** - определяет таймаут тестовых запросов.

Таймаут используется в двух случаях:

- Для задания времени между тестовыми запросами, задача которых – проверка работоспособности TCP/IP соединения с драйвером. Необходимость тестовых запросов обусловлена теоретической возможностью очень низкого трафика между драйвером и модулем опроса. Падение трафика может быть

обусловлено как функционированием УСД, так и аварийными (технологическими) ситуациями. Например: УСД может вести только архивы; в результате аварии произошел сбой на линии и опрос УСД временно невозможен, но это не является причиной остановки драйвера, более того, он должен работать, чтобы сервер телеметрии вовремя получил информацию о восстановлении связи с УСД; и пр.

- Задание таймаута между событиями: останов драйвера (точнее закрытие модулем опроса сокетного соединения с драйвером) – попытка модуля опроса вновь запустить драйвер.

- **Driver** – здесь задается та информация, которая должна быть передана драйверу при запуске в командной строке и которая неинтересна модулю опроса. Зависит от конкретного драйвера, требования к формату и содержанию строки должны быть описаны в документации на драйверный модуль.

Например:

```
/stm/setport LOG=set130Ps25P1.log DEBUG=18
```

- **Version** – определяет специфику опроса текущих и архивных данных для каждого конкретного драйвера.

На текущий момент существует 4 разновидности базового протокола обмена информацией между модулем опроса и драйверами, и именно здесь задается, какая именно будет использоваться в каждом конкретном случае. Модуль опроса поддерживает все 4 разновидности, и, следовательно, содержимое поля определяется только возможностями драйверного модуля.

Внесем некоторые пояснения:

- Асинхронный режим – это режим, в котором драйвер по собственной инициативе может передать информацию модулю опроса об изменении значения некоторого измеряемого УСД параметра. Более того, при получении запроса, в котором признак асинхронности установлен, драйвер обязан самостоятельно начать периодический опрос прибора. Этот режим касается ТОЛЬКО текущих параметров. Если режим включен, имеет смысл несколько увеличить значение поля «Period» типа «Devices» для всех УСД, которые опрашиваются данным драйвером.

- Интервальный опрос архивов – предусматривает возможность получения драйвером запроса от сервера телеметрии на выдачу нескольких идущих подряд архивных точек.

В таблице 1 приведены характеристики различных версий протокола.

Остальные характеристики полей представлены в таблице 2.

Таблица 1. Версии протокола.

Значение поля Version	Асинхронный режим для текущих.	Интервальный опрос архивов
0	Нет	нет
1	Нет	разрешен
2	Включен для всех текущих	разрешен
3	Включен, есть возможность выбора: какие текущие будут опрашиваться в асинхронном режиме, а какие нет	разрешен

Таблица 2. Характеристики полей типа Port.

Имя поля	Комментарий	Тип поля	Специфичность
Driver	Командная строка запуска драйверного модуля	Строка	Object_Spec
Parallel	Признак параллельности	Целое без знака 8-бит	Object_Spec
PortData	Характеристики порта	Строка	Object_Spec
tOutRequest	Таймаут отправляемый в запросе, мсек	Целое со знаком 32-бит	Object_Spec
tOutTest	Таймаут тестовых запросов, мсек	Целое со знаком 32-бит	Object_Spec
Version	Режим опроса.	Целое без знака 8-бит	Object_Spec

Таблица 3. Для примера приводим значение полей типа Port, для контроллера СЭТ-4М, как наиболее часто используемого

Имя поля	Значение поля
Driver	/stm/setport LOG= set130Ps25P1.log DEBUG=18
Parallel	1
PortData	10.0.9.34:5201
tOutRequest	25000
tOutTest	100000
Version	2

Тип <Device>

Глубина хранения - 1. Объекты типа Device описывают устройства сбора данных и содержат следующие поля:

- **Port** – здесь указывается имя объекта из типа «Port» (т.е. имя драйвера), который будет опрашивать УСД.
- **Address** – определяет адрес УСД или номер телефона, если УСД подключено через медленный модем. Например, адресом устройства может быть адрес прибора на общей шине (интерфейс 485). Если УСД не имеет адреса или он не необходим драйверу для успешного опроса УСД, рекомендуем ставить в этом поле 0.
- **DeviceTimeDelt** – определяет разницу между временем устройства и временем сервера (в сек). Это поле заполняется «модулем опроса», используется для контроля за временем прибора. Время на сервере считается эталонным. Расхождение времени между УСД и сервером более трети минимального отчетного периода (обычно минимальным являются 3-х минутные показания) нежелательно.

- **ArchDepth** - определяет глубину анализа целостности базы данных по старту сервера телеметрии. Заполняется поле по следующему правилу: имя_типа число_точек, типы разделяются пробелами. Например:

PowerTIT_Min3 180 PowerTIT_Min30 100 PowerTIT_Hour 96

При таком значении строки, модуль опроса проверяет, все ли 3-х минутки на глубину $180 \cdot 3 = 540$ мин (9 часов) присутствуют в БД и достоверны ли они. 30-ти минутки проверяются на глубину – $100 \cdot 30 = 3000$ мин (50 часов ~ 2 дня), часовые архивы - на глубину 96 часов (4 дня). В случае отсутствия некоторых архивных значений или при их низкой достоверности, они (значения) будут вновь запрошены с УСД.

При заполнении этого поля надо учитывать глубину архива данных УСД и/или драйвера (в случае если драйвер ведет собственные промежуточные БД). Не имеет смысла устанавливать глубину анализа на глубину, большую, чем архив УСД и/или драйверного модуля, т.к. в случае отсутствия данных в БД сервера телеметрии будут сформированы и посланы запросы к драйверу, а ответов на эти запросы не будет.

Также необходимо понимать, что задание большой глубины анализа может привести к существенному замедлению работы по выкачке данных при старте сервер телеметрии.

- **Activate** – определяет состояние прибора. 1- прибор в работе, 0 - прибор выведен из работы. Это поле может быть изменено оператором. В этом случае модуль опроса соответствующие действия по запуску (остановке) драйвера, т.е. у персонала есть возможность управлять опросом УСД.
- **TimeActivate** – определяет время вывода/ввода УСД из работы/в работу. Устанавливается оператором с экрана монитора до снятия УСД в ремонт или после установки

устройства в работу. Одновременно изменяется оператором и значение поля Activate. Поле предназначено для фиксации времени выключения (включения) УСД.

- **Period** – указывает на период опроса текущих. Период опроса определяется в миллисекундах. Рекомендуем устанавливать значение поля в соответствии с формулой: $tOutRequest * 2.5 * N \sim Period$ (где N – количество текущих для данного УСД). Если значение поля равно 0, то опрос текущих значений не производится.
- **EndPeriod** - определяет по началу или по концу периода производить опрос архивных точек у драйверного модуля. 1 – опрос по концу периода, 0 – опрос по началу периода. Например, период 30-минуток с 14:00 до 14:30. Если опрос по концу периода, то в 14:31 модуль опроса выдаст заказ на получение 30-минутки за время 14:31. А если опрос по началу периода, то в 14:31 модуль опроса выдаст заказ на получение 30-минутки за время 14:00.
- **EndPeriodDB** – определяет по началу или по концу периода производить запись в базу данных. 1 – запись по концу периода, 0 – запись по началу периода. Например, период 30-минуток с 14:00 до 14:30. Рассмотрим все 4 варианта:
 - § опрос по концу периода и запись в базу данных по концу периода, то в 14:31 «модуль опроса» выдаст заказ на получение 30-минутки за время 14:30. В базу данных запишется значение на время 14:30;
 - § опрос по началу периода и запись в базу данных по началу периода, то в 14:31 «модуль опроса» выдаст заказ на получение 30-минутки за время 14:00. В базу данных запишется значение на время 14:00;
 - § опрос по концу периода и запись в базу данных по началу периода, то в 14:31 «модуль опроса» выдаст заказ на получение 30-минутки за время 14:30. В базу данных запишется значение на время 14:00;
 - § опрос по началу периода и запись в базу данных по концу периода, то в 14:31 «модуль опроса» выдаст заказ на получение 30-минутки за время 14:00. В базу данных запишется значение на время 14:30;
- **Pereopros** – формируется оператором с экрана монитора и определяет дату, начиная с которой будет произведен переопрос всех архивных точек всех типов, относящихся к данному УСД. Если указана дата раньше, чем глубина в базе данных, то дата корректируется модулем опроса.

Остальные характеристики полей представлены в таблице 4.

Таблица 4. Характеристики полей типа Device.

Имя поля	Комментарий	Тип поля	Специфичность
Port	Имя объекта типа Port, которому соответствует прибор	Ссылка на объект типа Port	Object_Spec
Address	Адрес устройства	Целое со знаком 16-бит	Object_Spec
DeviceTimeDelt	Разница между временем устройства и временем сервера (в секундах)	Целое со знаком 32-бит	Object_Spec
ArchDepth	Глубина анализа архивов (Пары Имя_типа число_точек)	Строка	Object_Spec
Activate	1 - прибор в работе, 0 - прибор выведен из работы	Целое без знака 8-бит	Object_Spec
TimeActivate	время вывода прибора из работы	Дата, время	Object_Spec
Period	Период опроса текущих значений в мс	Целое без знака 32-бит	Object_Spec
EndPeriod	1-запрос по концу периода, 0-по началу	Целое без знака 8-бит	Object_Spec
EndPeriodDB	1-запись в базу по концу периода, 0-по началу	Целое без знака 8-бит	Object_Spec
Pereopros	переопрос устройства со времени: дата, время	Дата, время	Object_Spec
DevDayMode	Смещение суточных на начало периода, если EndPeriodDB установлен в 1. *	Целое без знака 8-бит	Object_Spec

*Использовать только вместе с версиями, 3-4 модуля, 2.0 и старше.

Таблица 5. Для примера приводим значение полей типа Device, для контроллера СЭТ-4М, как наиболее часто используемого

Имя поля	Значение поля
Port	set130Ps25P1
Address	9
DeviceTimeDelt	0
ArchDepth	PowerTIT_Min 480 PowerTIT_Min30 200 PowerTIT_Day 10
Activate	1
TimeActivate	0
Period	15000
EndPeriod	1
EndPeriodDB	0
Pereopros	0

Имя объекта: 130Ps25_N1 – гор. -130, п/с 25, насос 1. Объект содержит ссылку на объект типа Port: set130Ps25P1. Шинный адрес контроллера – 9.

2. Типы, используемые для накопления информации, получаемой с приборов.

Объекты STM могут содержать произвольный набор полей. Но типы, объекты которых должны заполняться модулем опроса данными, полученными с УСД, обязательно (кроме IsTrac) должны содержать в себе следующие поля:

- **_avalue** – Значение телеизмерения от оборудования. В это поле записывается значение, полученное по запросу от драйвера (УСД).
- **_mvalue** – Значение телеизмерения от ручного ввода. В это поле записывается значение, полученное от клиента (введенное оператором).
- **Man** – Признак ручного ввода (0/1). Задается с клиента. Определяет порядок заполнения поля **Value**. Если **Man=0**, то в поле **Value** копируется значение поля **_avalue**. Если **Man=1**, то в поле **Value** копируется значение поля **_mvalue**.
- **Arc** – Номер параметра в приборе. Для некоторых контроллеров. Например, для УСДС-16М указывает номер контакта и коэффициент пересчета. Смотри описание драйверов.
- **Device** - Имя объекта в типе **Device**, к которому привязан ТПТ. Обязательно для заполнения. Указывает на УСД, с которого читается параметр.
- **IsTrac** – флаг, позволяющий указывать необходимость асинхронного режима опроса данного параметра. Нет необходимости добавлять это поле, если нет драйверов, поддерживающих 3-ю версию протокола обмена. Так же, не имеет смысла добавлять это поле в архивные типы.

Если драйвер работает с 3-й версией протокола, то поле обязательно. Возможные значение поля: 1 – данный параметр будет опрашиваться в асинхронном режиме; 0 – данный параметр будет опрашиваться в нормальном режиме.

Таблица 6. Характеристики объектов типа PowerТПТ, TechТПТ и ...

Имя поля	Комментарий	Тип поля	Специфичность
_avalue	Значение телеизмерения от оборудования	С плавающей точкой 64-бит	Time_Spec
_mvalue	Значение	С плавающей точкой	Time_Spec

	телеизмерения от ручного ввода	64-бит	
Arc	Номер параметра в приборе	Строка	Object_Spec
Device	Имя объекта в типе Device, к которому привязан ТИТ. Обязателен для заполнения.	Строка	Object_Spec
Value	Значение телеизмерения	С плавающей точкой 64-бит	Time_Spec
Man	Признак ручного ввода (0/1)	Целое без знака 8-бит	Time_Spec
IsTrac	Включение/выключение асинхронного режима опроса текущих	Целое без знака 8-бит	Object_Spec

* Более полную информацию о типовом наборе полей смотри в Приложении 1.

На один контроллер (объект в типе **Device**) ссылается как правило несколько параметров. Например, контроллер СЭТ-4М от 6 до 37 параметров (объектов типа PowerTIT...). Перечень опрашиваемых параметров задан по умолчанию либо описан в специальном файле драйверного модуля. См. описание драйверов. Для инициации опроса параметра достаточно правильно описать его в конфигурации драйверного мастера.

Например:

Объект **130Ps25_N1-P** описанный в типе **PowerTIT** – текущее значение активной мощности.

Объект **130Ps25_N1-Wa** описанный в типе **PowerTIT_Min30** – архивное значение активной энергии за 30 минутный интервал. Оба параметра должны ссылаться на один объект типа **Device**. Порядок опроса различных типов определяется модулем опроса (модуль 3-4). См. описание модуля.

Для текущих значений в конфигураторе необходимо задавать глубину ретроспективы =1 и период опроса. Для архивных значений в конфигураторе необходимо задавать требуемую глубину ретроспективы и периодичность записей. См. описание конфигуратора.

3. Специальные типы.

В специальные типы можно выделить:

3.1 Тип List.

Тип **List** предназначен для создания журналов событий. Используется совместно со специализированным скриптом “Модуль журналов”. Необходимо задавать глубину ретроспективы.

Тип List содержит следующие поля.

Таблица 6. Поля типа List.

Имя поля	Комментарий	Тип поля	Специфичность
Description	Описание события	Строка	Time_Spec
Event	Код события(0 – не определен, 1 - квитирование)	Целое со знаком 8- бит	Time_Spec
EventTime	Время события	Дата,время	Time_Spec
ObjectDescr	Объект, с которым связано событие	Строка	Time_Spec
ScrName	Источник информации о событии	Строка	Time_Spec

Все поля кроме **Description** заполняются скриптом.

Тип **List**, как правило используется на серверах-концентраторах и серверах обслуживающих отдельные небольшие п/станции, участки и тд.

3.2 Тип Ping

Тип предназначен для хранения информации о текущем состоянии связи с приборами, имеющими IP адреса. Диагностика. Глубина ретроспективы = 1.

Таблица 7. Характеристики объектов типа Ping.

Имя поля	Комментарий	Тип поля	Специфичность
_avalue	Значение телеизмерения от оборудования	Целое без знака 8-бит	Time_Spec
_mvalue	Значение телеизмерения от ручного ввода	Целое без знака 8-бит	Time_Spec
Arc	IP – адрес устройства	Строка	Object_Spec
Comment	Имя контролируемого устройства	Строка	Object_Spec
Value	Значение телеизмерения	Целое без знака 8-бит	Time_Spec
Man	Признак ручного ввода (0/1)	Целое без знака 8-бит	Time_Spec

Поля **_avalue**, **_mvalue** и **Man** не обязательны, но могут использоваться для индикации вывода прибора в ремонт, временного отключения и тд.

3.3 Тип PowerMax_Day

Специальный тип для расчета максимальной мощности в часы утренних и вечерних максимумов. Используется совместно со

специализированным скриптом “Подсчет утренних и вечерних максимумов”. См. описание модуля: “Мдуль_MAX.doc”. Работает на базе 30 минутных значений мощности (PowerTIT_Min30 или Gr_Min30).

Таблица 8. Характеристики объектов типа PowerMax_Day

Имя поля	Комментарий	Тип поля	Специфичность
Value	Максимальное значение	С плавающей точкой 64-бит	Time_Spec
MaxType	Тип в котором расположены объекты для расчета максимума	Строка	Object_Spec
MaxTime	Время максимума	Дата, время	Time_Spec
Comment	Комментарий. Не обязательно	Строка	Object_Spec

Для успешной работы необходимо задать в типе **PowerMax_Day** по 3 объекта на каждый анализируемый объект (мощность).

Например: для анализа параметра мощности описываемого объектом **PowerTIT_Min30. 130Ps25_N1-P** необходимо задать:

PowerMax_Day. 130Ps25_N1-P-Eve - вечерний максимум
PowerMax_Day. 130Ps25_N1-P-Morn - утренний максимум
PowerMax_Day. 130Ps25_N1-P - общий максимум

Также в типе **Staff** (см. п/п 3.2) необходимо задать 8 объектов задающих начало и конец периодов максимума.

Staff.Time_UMAX_Start-Hour Утренний макс., начало, часы
Staff.Time_UMAX_Start-Min Утренний макс., начало, мин.
Staff.Time_UMAX_End-Hour Утренний макс., конец, часы
Staff.Time_UMAX_End-Min Утренний макс., конец, мин.

Staff.Time_WMAX_Start-Hour Вечерний макс., начало, часы
Staff.Time_WMAX_Start-Min Вечерний макс., начало, мин.
Staff.Time_WMAX_End-Hour Вечерний макс., конец, часы
Staff.Time_WMAX_End-Min Вечерний макс., конец, мин.

Расчет в локальном времени и производится по окончанию суток. Значение этих объектов можно, при необходимости, изменять с клиента.

3.4 Тип ExelServer

Тип предназначен для организации связи между Master17 и клиентом (Программы ExelContrl, ContrSmen. Отчетные формы,

прогноз...)). Используется совместно со специализированным скриптом “Модуль STM_EXEL”. См. описание модуля: ”Модуль_STM-EXEL.doc”. Содержит один объект и одно поле.

ExelServer.Server.port

Поле содержит номер порта зарезервированного сервером для создания соединения. Например: 7820. Если на одной платформе, работает несколько серверов, номера портов должны быть разными.

4. Служебные типы.

4.1 Тип XXX.

Тип используемый для формирования журналов. Содержит один объект: **xxx** и два поля. **key** – ключ пользователя и **uname** – имя пользователя. Если на сервере ведутся журналы тип.объект **XXX.xxx** обязательны. Используются совместно с типом **List**, как правило, на серверах-концентраторах.

4.2 Тип Staff

Тип используется для ввода редко изменяемых параметров используемых в расчетах. Это могут быть производительности агрегатов (компрессоры, вентиляторы и тд), коэффициенты пересчета параметров получаемых с приборов, временные точки и тд. Тип содержит три поля: **Value** – значение параметра, **stime** – время изменения параметра, **Comment** – комментарий. **Value** и **stime** – имеют специфичность **Time**. **Comment** имеет специфичность **Object**. Если нет необходимости отслеживать динамику изменения параметров, то полям **Value** и **stime** можно поставить специфичность **Object**. Вводя тс яс клиента.

4.3 Тип Flags

Тип используется для управления работой скриптов **master17**, использующихся эпизодически. Например, для генерации отчетов за нестандартные периоды (декада, квартал и тд). Структура типа произвольна.

5. Расчетные объекты (типы). Поле Calc

Как правило, одних параметров получаемых с приборов для полноценной работы системы недостаточно. Необходим пересчет параметров, например, для получения нужной размерности, суммирования параметров, расчет удельных характеристик и другие расчеты. Признаком расчетного объекта является **не пустое** поле

Calc. Если в конфигурации есть расчетные объекты (типы), то необходимо подключение модуля 5 (Калькулятор). Смотри описание модуля : “Модуль_5.doc”.

Часть расчетных объектов удобно сгруппировать в отдельных типах. Например, суммарные электрические параметры в некоторых конфигурациях STM объединены в типах: **Gr_Day**, **Gr_Min30**, **Gr_Min3**.

Пример. П/станция №1 имеет два ввода и три потребителя. Суммируем активную энергию по вводам.

Поле **Calc** суммарного объекта **Gr_Day.Ps1_SUM_Vv-Wa** содержит строку:

$$\{\text{PowerTIT_Day.Ps1_Vv1-Wa,0,100}\}+$$

$$\{\text{PowerTIT_Day.Ps1_Vv2-Wa,0,100}\}$$

Суммируем активную энергию по потребителям.

Поле **Calc** суммарного объекта **Gr_Day.Ps1_SUM_Pp-Wa** содержит строку:

$$\{\text{PowerTIT_Day.Ps1_P1-Wa,0,100}\}+$$

$$\{\text{PowerTIT_Day.Ps1_P2-Wa,0,100}\}+$$

$$\{\text{PowerTIT_Day.Ps1_P3-Wa,0,100}\}$$

Получив энергию по вводам и потребителям можно рассчитать расход энергии на собственные нужды.

Поле **Calc** объекта собственных нужд **Gr_Day.Ps1_SV-Wa** содержит строку:

$$\{\text{Gr_Day.Ps1_SUM_Vv-Wa,0,100}\}-$$

$$\{\text{Gr_Day.Ps1_SUM_Pp-Wa,0,100}\}$$

Также, можно рассчитать среднюю мощность за сутки по вводам, потребителям, собственным нуждам.

Строки в поле **Calc**:

$$\text{Gr_Day.Ps1_SUM_Vv-P} - \{\text{Gr_Day.Ps1_SUM_Vv-Wa,0,100}\}/24$$

$$\text{Gr_Day.Ps1_SUM_Pp-P} - \{\text{Gr_Day.Ps1_SUM_Pp-Wa,0,100}\}/24$$

$$\text{Gr_Day.Ps1_SUM_SV-P} - \{\text{Gr_Day.Ps1_SUM_SV-Wa,0,100}\}/24$$

Подробнее, о формате поля **Calc**, смотрите в описании расчетного модуля.

6. Типы получаемые с других серверов. Источники данных.

Для создания иерархических и прочих структур в STM реализована система обмена данными. Смотри описание “Менеджера потоков”. Передавать можно произвольные поля любых объектов и типов. Для организации передачи данных необходимо:

- В конфигурации сервера-приемника создать источник данных. Указать имя конфигурации сервера-источника, его IP адрес и порт. Задать “Имя пользователя” и пароль. Пользователь и пароль должны быть заведены в конфигурации сервера-источника.
- В описании типа создать правила копирования. Смотри рисунок. Копируются поля из типа Work_Min3 (Источник данных – конфигурация icp_sh14.stm) в тип Agregat_Min3 текущей конфигурации. Имена поля типа приемника и типа источника могут совпадать. T_Str – Value. Главное требование: у копируемых полей должны совпадать типы. Например: ”Строка” - ”Строка”. Для удобства заполнения рекомендуется, чтобы имена объекта-источника и объекта-приемника совпадали.

Правило копирования

Название правила :

Комментарий :

Источник данных :

Тип данных :

Поле	Источник	Инициализация
\$Time	\$Time	
\$Valid	\$Valid	
Sign	Sign	
Ton	Ton	
Toff	Toff	
Non	Non	
Noff	Noff	
Wa		
Gud		
VS		
Nob		
Comment		
T_Str	Value	
C_Gud		

- Создаем связи между объектами-источниками и объектами-приемниками. Выбираем объект. Ставим галочку в боксе “Правила копирование”. Выбираем нужное правило. Если имена объекта-источника и объекта-приемника совпадают и правило выбрано правильно, то в боксе “Объекта источник:” появится имя объекта из конфигурации-источника .

Правило копирования :

Agregat

Объект источник :

140Ps_AU1-W