

## Описание примера EventTable

Для отображения аварийных и информационных сообщений обычно используется элемент **Историческая таблица**. Но в некоторых случаях его функционала оказывается недостаточно – например, если необходимо:

- отображать метку времени в формате dd.MM.yyyy HH:mm:ss (в исторической таблице используется формат yy.MM.dd HH:mm);
- в строке события отображать значения некоторых регистров (например, значение температуры при аварии);
- организовать архивацию аварий и событий на USB-накопитель.

В данном примере демонстрируется, как реализовать собственную таблицу событий на базе элемента **Таблица отображения данных**.

В рамках примера рассматривается система дозации некоего вещества. В системе необходимо фиксировать два события (начало и конец дозации), с которыми связаны параметры Режим работы (автоматический и полуавтоматический режим) и Объем вещества.

Табл.1. Список используемых в примере переменных

Название	Тип данных	Бит/Регистр элемента
Начало дозации	BOOL	PSB340
Конец дозации	BOOL	PSB341
Автоматический режим	BOOL	PSB350
Полуавтоматический режим	BOOL	PSB351
Объем вещества	WORD	PSW266

Таким образом, таблица отображения данных содержит 4 столбца:

- время (STRING, 10 регистров = 20 символов);
- событие (STRING, 10 регистров = 20 символов);
- режим (STRING, 9 регистров = 18 символов);
- время (WORD, 1 регистр).

Соответственно, одна строка таблицы занимает 30 регистров (10+10+9+1).

Данные таблицы размещены в области PFW (энергонезависимая память) – так что история событий будет сохранена даже при выключении панели. В рамках примера таблица состоит из 100 строк (параметр **Всего записей** в настройках таблицы), на экране отображаются 10 из них (параметр **Строк на странице**). Прокрутка строк осуществляется кнопками **Вверх/Вниз**. Кроме того, таблица имеет дополнительную строку, которая не

видна пользователю (см. ниже). В памяти панели таблица занимает  $30 \times 100 + 30 = 3030$  регистров (PFW300...3329).

Для очистки памяти таблицы при загрузке нового проекта использован механизм рецептов (в программе **Конфигуратор СП300** в меню **Файл** выбрать **Создание рецептов**).

Новая запись добавляется в верхнюю строку таблицы, остальные строки при этом смещаются вниз.

Для обработки событий используется две функциональных области (по области на каждое событие), которые отличаются только привязанными битами (**PSB340** и **PSB341**). При вызове функциональной области происходит следующее:

- Смещение строк таблицы на одну вниз за счет копирования регистров **PFW300...3299** в **PFW330...3329** (для этого и нужна дополнительная 101-я строка);
- Вызов макроса **EventTable** для формирования в регистрах панели **PSW300...329** новой строки таблицы. Текстовые сообщения на русском языке в рамках макроса записываются в виде набора ASCII-кодов;
- Копирование регистров **PSW300...329** в **PFW300...329** (запись нового события в первую строку таблицы).

Еще 5 функциональных областей используются для управления режимами (для исключения ситуаций, когда одновременно активны оба режима работы или неактивен ни один из них).

Для записи событий на USB используются два элемента **Архивирование на USB** (по элементу на каждое событие), которые отличаются только привязанными битами (**PSB340** и **PSB341**). В качестве регистров хранилища элементы используют регистры **PSW400...463** (см. формулу расчета в РП).

Функциональные области и элементы **Архивирование на USB** рекомендуется делать **глобальными**, чтобы они работали независимо от текущего открытого экрана.

### Вопросы и ответы

**Вопрос:** в примере используются биты и регистры самой панели. В моем случае панель используется в режиме **master**, и биты событий и режима, а также значение объема должны считываться из slave-устройства. Как адаптировать пример к этой ситуации?

**Ответ:** считывайте нужные данные с помощью функциональной области в регистры панели (действия **Копировать бит**, **Копировать регистр**, **Копировать группу регистров**).

**Вопрос:** как изменить число строк в таблице?

**Ответ:** см. параметр **Всего строк** в настройках таблицы. Также следует поменять диапазоны регистров в функциональных областях при смещении строк таблицы и в меню **Создание рецептов**.

**Вопрос:** как редактировать и добавлять тексты событий? Как изменить их размер?

**Ответ:** см. макрос **EventTable** – в частности, define'ы – и описание функции **snprintf** в Интернете. Следует учесть, что размера массива символов больше фактического размера его строкового представления на 2 байта.

**Вопрос:** если событий будет много, то надо для каждого из них создавать отдельную функциональную область и элемент **Архивирование на USB**?

**Ответ:** нет, не обязательно. В рамках примера события всего два – и, действительно, было проще обойтись копированием элементов. При значительном числе событий более разумно добавить одну функциональную область, которая **циклически** вызывает дополнительный макрос, в котором генерируется «резльтирующий» бит по всем событиям. Этот результирующий бит привязан к **одной** функциональной области и **одному** элементу **Архивирование на USB** (таким образом, на всю таблицу с любым количеством событий достаточно двух функциональных областей и одного элемента архивации).

**Вопрос:** в примере в архиве используются англоязычные заголовки столбцов. Можно ли использовать русскоязычные?

**Ответ:** да, см. пояснения в [этом видео](#).

**Вопрос:** в столбце **Объем** при отсутствии события отображается значение 0. Можно ли сделать, чтобы отображалась пустая строка?

**Ответ:** да, для этого потребуется привести это значение к строковому виду (в таблице использовать тип STRING, в макросе произвести конвертацию числа в строку с помощью функции **snprintf** – по аналогии с остальными строками). Очевидно, что это увеличит размер строки таблицы.