

# РУБИКОН-СКРИПТ СРЕДСТВО ДЛЯ НЕСТАНДАРТНОЙ ОБРАБОТКИ СОБЫТИЙ

У Вас бывало, что система спроектирована, собрана и настроена, а клиент говорит, что хотел бы, чтобы вот тут и тут она работала несколько иначе? Например, в системе пожаротушения, как известно, при открывании двери в помещении должен останавливаться процесс пуска тушения. А после закрывания двери он должен возобновляться? Нормативные документы нам ничего не говорят, видимо, полагая, что раз кто-то там есть, кто открывает и закрывает двери, то, значит, на его совести решить, что делать дальше. Я легко могу привести аргументы за и против любого решения. Честно говоря, я даже не помню, что в таких случаях делает Рубикон. Однако у конкретных ответственных за конкретные объекты могут быть свои убеждения и предпочтения, и нередко проще уступить клиенту, нежели объяснить, что он не вполне прав. Что же делать? Или, например, клиент хочет, чтобы после постановки на охрану гаража последовательно загоралось несколько фонарей на подъездной дороге. Ему приятно осознавать, какая могучая электроника заботится о его собственности. Как же это реализовать?

Разработчики Сигма-ИС давно пришли к выводу, что на всех клиентов не угодишь, и решили встраивать в изделия средства для более или менее свободного программирования их на объекте силами пусконаладчиков. И в системе «Рубикон» также имеется средство под названием «Рубикон-скрипт». Должен сразу предупредить, что этой возможности нет в младшем исполнении ППК, но она есть в ППК-М, ППК-Е и ППК-Т. Что же нам дает скрипт?

Скрипт – это, по сути, небольшая программа для контроллера. Последовательность инструкций, которые ППК должен выполнить по заданному условию. Условия могут быть разные. Основные варианты – изменение состояния области или событие в области (например, проход разрешен).

```
if this != пожар 1 & есть неи
Эта область (this)
не равно (!=)
пожар 1 & есть неисправности
```

```
номер переменной
событие: доступ разрешен
ручной старт
перезапуск до окончания
```

Разумеется, инструкции – это не язык Си-шарп и даже не Бейсик. Последовательность инструкций в скрипте можно задать даже с пульта ППК выбором в меню на его компактном экране.

```
Добав. скрипт 3
Встав. настройки старта
Удал. if this == норма
реле1/ППК включить
```



Основные инструкции – это действия. Можно включать/выключать реле, ставить/снимать с охраны области и выполнять другие действия, в том числе арифметические операции с переменными. Одна из особенностей нашего скриптового языка – возможность повторно использовать один и тот же скрипт. Например, если в нескольких областях надо настроить одинаковые скрипты, достаточно создать один скрипт и затем включить его в несколько областей. Этот скрипт будет выполняться по событиям в разных областях и будет управлять, соответственно, разными областями и техсредствами в них. При задании действий скрипта Вы можете явно указать имя (номер) области или технического средства – и всегда будет выполняться команда именно на нем. Или же Вы можете указать «this» – тогда команда будет действовать на область, вызвавшую исполнение скрипта.

```
Выберите объект
1. Эта область (this)
2. Область
3. Элемент оборудования
```

Аналогично переменные, которые используются в скрипте, могут быть глобальные (одни и те же для всех скриптов), переменные области (могут использоваться всеми скриптами, но у каждой области свои) и, наконец, локальные, свои у каждого экземпляра скрипта. Это позволяет независимо исполнять один и тот же скрипт в нескольких областях и тем не менее, если надо, обеспечивать обмен данными между разными скриптами или одним и тем же скриптом, но в контексте разных областей. Перечитал и сам ужаснулся заумным формулировкам. На самом деле все не так сложно. В большинстве случаев ни о чем задумываться не надо, простые скрипты будут работать без лишнего знания.

Приведу пример простого скрипта:  
Условие запуска – по снятии с охраны области.  
Задержка 2 минуты.

*Поставить область на охрану.*

*Если область не на охране, включить тревогу.*

Что делает этот скрипт? Проверяет, что область снимают с охраны не более, чем на 2 минуты. Если по истечении заданного интервала область не поставили обратно на охрану, скрипт сам ее ставит на охрану, а если не получилось – будет тревога. Например, для сейфа с деньгами это уместно – нечего оставлять его не на охране. Если закрыли и забыли поставить на охрану – скрипт сделает это за Вас. Если же даже не закрыли (на охрану поставить не удалось) – будет тревога.

Или для описанного в начале статьи случая: по закрывании двери скрипт может проверить, что область в пожаре, и если автоматика включена, снова запустить последовательность тушения.

А вот для второго случая из начала статьи:

*При условии «постановка на охрану».*

*Включить лампу 1.*

*Задержка 10 секунд.*

*Включить лампу 2.*

*Задержка 10 секунд.*

*Включить лампу 1.*

*Включить лампу 3.*

*Задержка 10 секунд.*

*Включить лампу 2.*

*Задержка 10 секунд.*

*Включить лампу 3.*

Скрипт примитивный, но какой эффект! Бегущая световая волна!

Что еще хочется отметить. Редактор скриптов в пульте ППК простой, но весьма эффективный. Кто сталкивался с возможностями настройки условных команд в других системах, оценит такие возможности, как «скопировать скрипт» (так легко создавать незначительно отличающиеся варианты, не перебивая заново основную последовательность команд) «вставить/удалить команды в середине скрипта» (простые системы предлагают все удалить и начать с начала), и, конечно, возможность указывать объекты действий не цифрами, а выбором из иерархического списка областей и отдельных технических средств.

**СИГМА-ИС**

**ООО «СИГМА-ИНТЕГРИРОВАННЫЕ СИСТЕМЫ»**

105173, Москва, ул. 9-мая, д. 126

тел. (495) 542-4170,

факс (495) 542-4180

e-mail: info@sigma-is.ru

www.sigma-is.ru