

# Топикатор

Целью модуля является возможность реакции на сообщения от MQTT брокера.

## Общие положения

На основании БД mosquitto реализуется паттерн «доска объявлений».

Модуль устанавливается инсталлятором на целевой контроллер и запускается в виде сервиса.

Настройки подключения хранятся в файле настроек yaml формата.

Правила преобразования хранятся в файле настроек. Доступ управляется правами пользователя. Все значения (системных и из пользовательских файлов) объединяются.

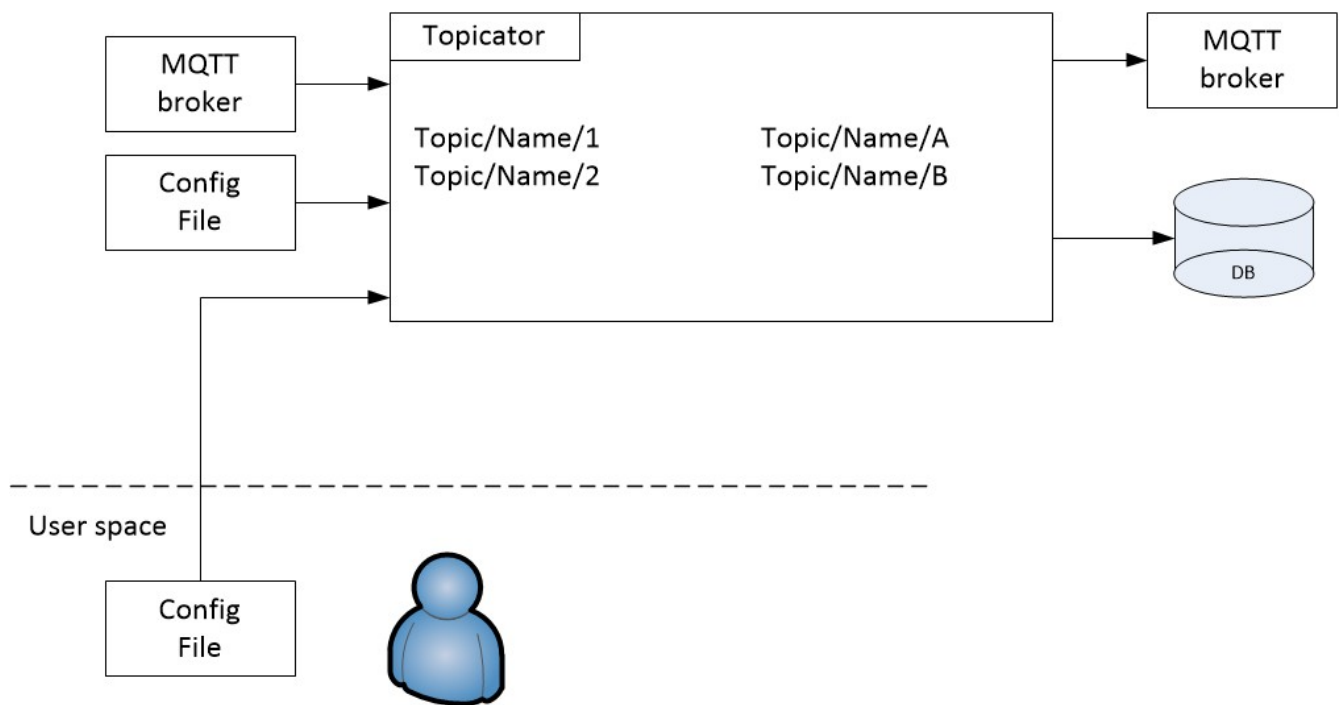
## Этапы реализации

### 1. Переименование топиков

Имеется проблема: имена топиков (из сторонних источников) не соответствуют принятому формату или неудобны в использовании.

Модуль берет список исходных и целевых имен топиков из файла — подписывается на сообщения брокера — и сохраняет пришедший контент с новым именем поэлементно.

Для удобства управления функционалом со стороны пользователя — все входные файлы конфигурации объединяются. В результате управлять логикой становится возможным простым копированием готовых текстовых файлов.

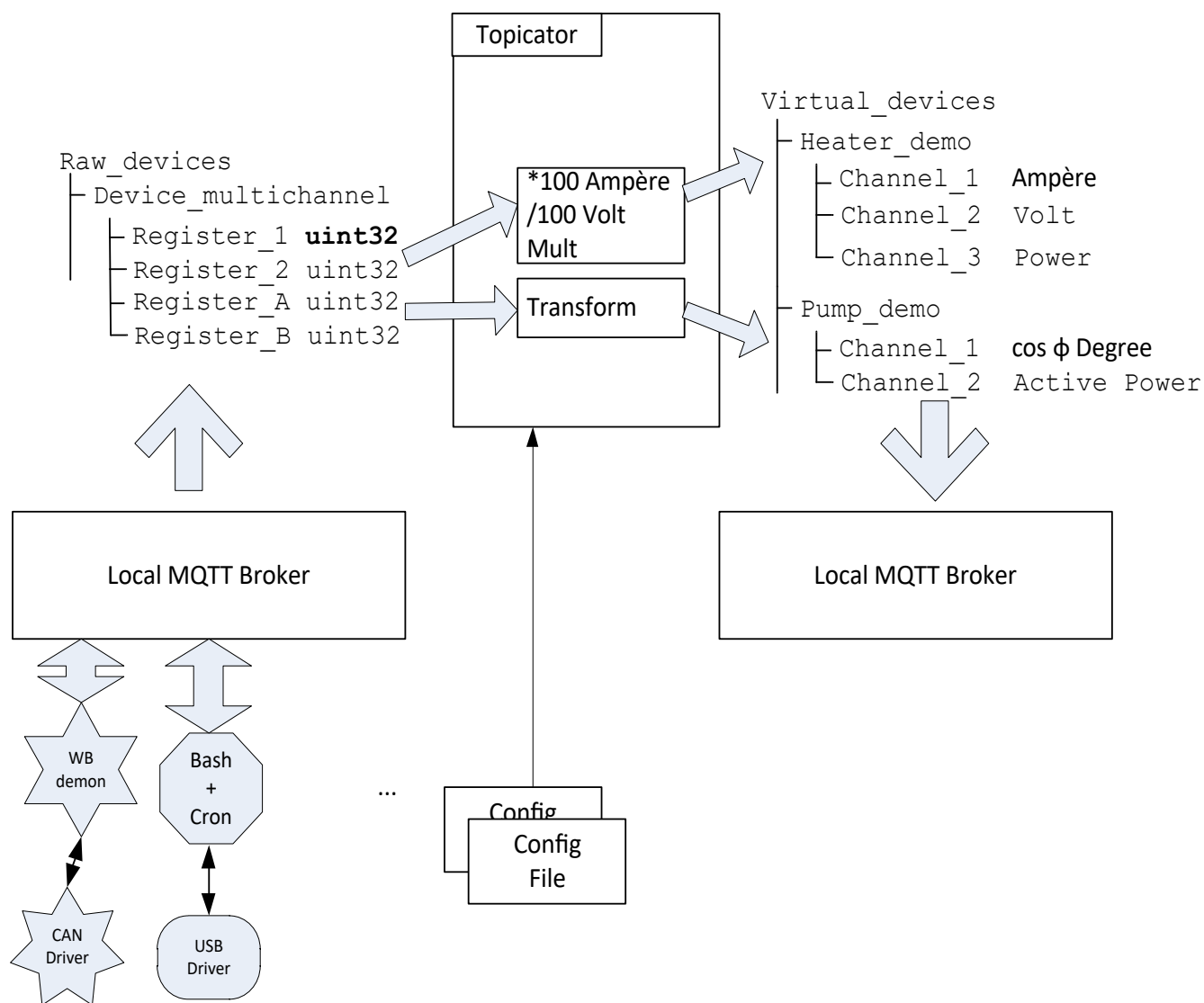


## 2. Операции над контентом

Имеется проблема: «сырые» данные, считанные из регистров периферии, могут быть некорректны/непрезентативны без дополнительной обработки (к примеру калибровки или преобразования единиц измерения). Сейчас данная функциональность размазана между драйверами устройств и визуализацией результатов измерений.

Модуль берет список исходных и целевых имен топиков из файла, а так же математическую операцию, которую требуется выполнить над данными — подписывается на сообщения брокера — и сохраняет пришедший контент с новым именем и значением поэлементно. Проверяется корректность входных данных.

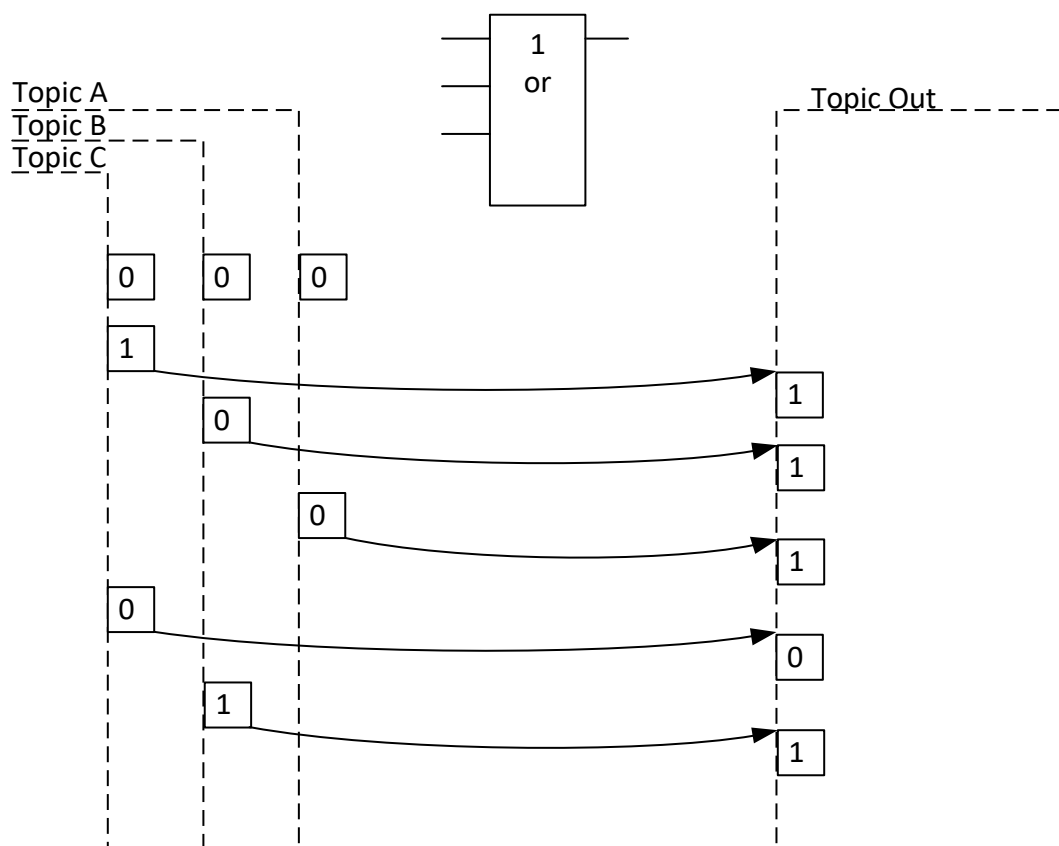
Дополнительным эффектом от подобного подхода является возможность формирования «виртуальных» устройств — на основе результатов обезличенных многоканальных измерителей. Пользователь формирует отдельный файл с описанием целевого устройства (модели), где описывает имена выходных топиков в общей ветке, и входных топиков типового измерителя с операцией преобразования. Файл может копироваться между устройствами. Соответственно при необходимости настройки такого «виртуального» устройства — правки нужно вносить в единое место — что удобно и не дает затронуть иную функциональность.



### 3. Групповые операции

Проблема: события реального мира могут быть логически связанными — к примеру: включение даже одной лампочки в комнате означает наличие света в ней. Требуется возможность генерации событий по нескольким взаимосвязанным условиям.

Модуль берет список исходных и целевых имен топиков из файла, а так же логическую операцию, которую требуется выполнить над данными — подписывается на сообщения брокера по каждому исходному топик — по пришествии события применяет логическую операцию ко всем исходным данным — и сохраняет результат вычислений с выходным именем.



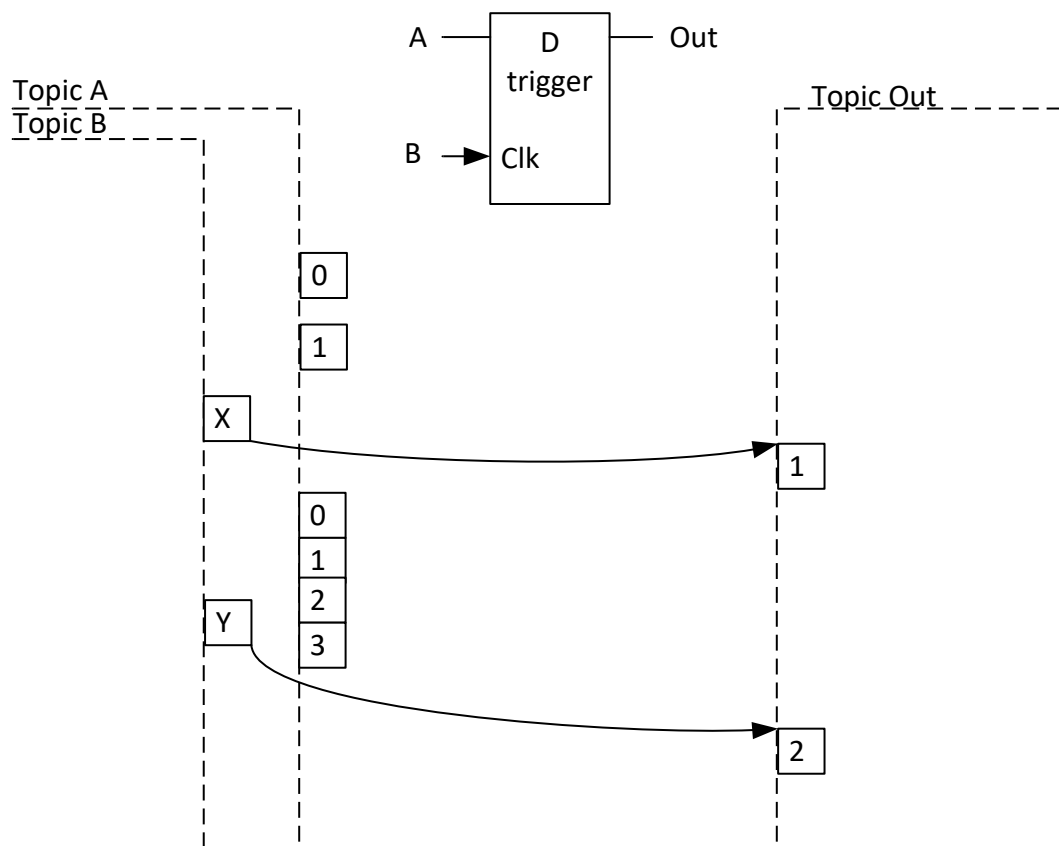
#### 4. Обработка по событию

Проблема: постоянное реагирование на изменения значений топиков может привести к лавинообразной обработке событий, а результат вычислений нигде использоваться не будет. В качестве примера — измерения датчика температуры нужны или периодически, или по запросу — нет смысла реагировать на шумы.

Модуль берет список исходных и целевых имен топиков из файла, а так же логическую операцию, которую требуется выполнить над данными. Один из входов помечается как «тактовый» - вычисления выполняются только по его изменению.

Обработка событий выполняется без памяти (QoS 0)

В логические операции добавляются генераторы событий — по таймеру, по дате, положению звезд на небе..



## Прочее

Визуализация.

Обработка Кешированных событий QoS 2

Пересылка сообщений в БД