

Мы стремимся к тому, чтобы каждый сервис был полностью отделен от других.

VCS/CI/CD на сервис: один сервис – один репозиторий, один сервис – один CI-конвейер, один сервис – один CD-конвейер.

Это значит, что у одного сервиса свой репозиторий имеет собственный конвейер сборки и собственный конвейер развертывания, собственная БД.

Для взаимодействия сервисов есть два подхода: синхронный RPC (HTTP REST); асинхронный Events (FMQP, MSMQ и т.д.).

Недостатками первого, в нашем случае, являются: отложенная согласованность; широковещательные сообщения; отложенные запросы.

Поэтому выбираем Events, асинхронные взаимодействия на шинах данных. Будем использовать независимый от платформы протокол AMQP и брокер сообщений RabbitMQ. Это позволит нам использовать в нужных местах, нужные языки и тем самым не привязываться к стеку, а взаимодействия всех сервисов будет происходить через events, то что нам и нужно.

На стороне клиентов есть какой-то frontend и они скорее всего захотят общаться с backend`ом через Rest API, для этого используем ApiGateway. ApiGateway – сервис, который позволяет создать тонкий фасад на сетевом уровне, к которому мы подключаем все наши микросервисы, тем самым делая единую точку входа для front`а.

Так как у нас микросервисная архитектура, нам надо решить проблему с логированием. Для этого все запросы связываются единым ID, который формируется на уровне ApiGateway и потом добавляется ко всем запросам, events которые идут во внутреннюю систему, таким образом можно использовать ActivityID.

Использование данного подхода положено в основу проекта электронного документооборота ВГТУ.

Список использованных источников

1. Amazon API Gateway [Электронный ресурс]. – Режим доступа: <https://aws.amazon.com/ru/api-gateway/getting-started/>. – Дата доступа: 15.05.2020.
2. RannitMQ [Электронный ресурс]. – Режим доступа: <https://www.rabbitmq.com/#getstarted>. – Дата доступа: 15.05.2020.

УДК 004.4

SCADA-СИСТЕМА КОНТРОЛЯ ТЕМПЕРАТУРЫ

**Самусев А.М. маг., Кузнецов А.А., проф., Ринейский К.Н. ст. преп.,
Чернов Е.А., ст. преп.**

*Витебский государственный технологический университет,
г. Витебск, Республика Беларусь*

Реферат. Статья посвящена разработке программы дистанционного мониторинга температурных параметров промышленного объекта.

Ключевые слова: SCADA, SCADA-система, MasterSCADA, контроль температуры.

SCADA (supervisory control and data acquisition, диспетчерское управление и сбор данных) – программный пакет, предназначенный для разработки или обеспечения работы в реальном времени систем сбора, обработки, отображения и архивирования информации об объекте мониторинга или управления. SCADA может являться частью АСУ ТП, АСКУЭ, системы экологического мониторинга, научного эксперимента, автоматизации здания и т. д.

Разработанный проект на основе SCADA (MasterSCADA) контроля температуры направлен на сбор и архивирование данных, непрерывный мониторинг температурных показателей распределённой системы и уведомление оператора о нештатных ситуациях. Структура проекта имеет вид (рис. 1):

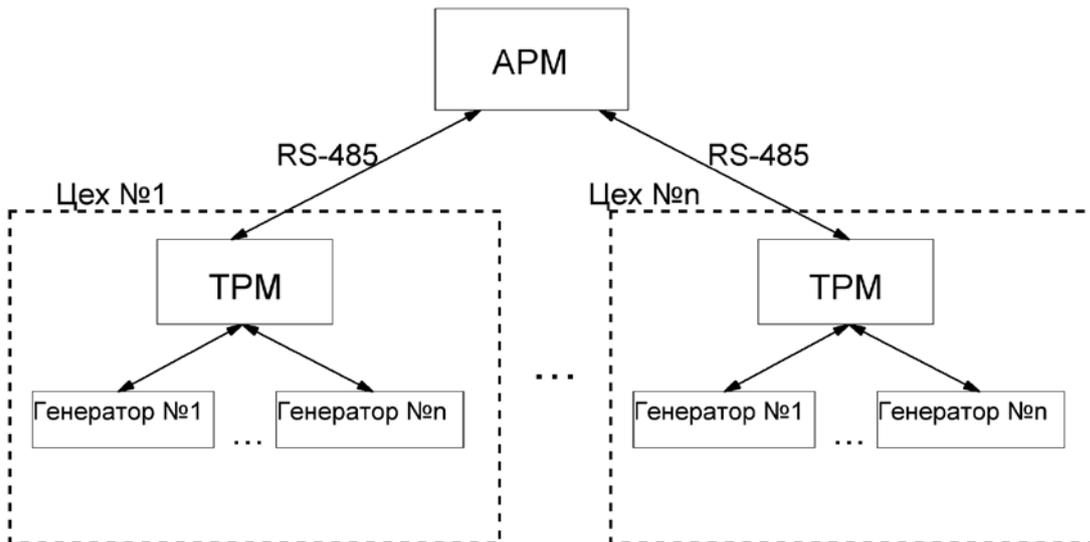


Рисунок 1 – Структура проекта

При открытии программы пользователь попадает на начальную страницу, где появляется возможность выбора объекта контроля (формовочный цех, полигоны) и настройки автоматических суточных отчётов (рис. 2). При выборе объекта пользователь может выбрать доступные устройства (генераторы) или вернуться на начальную страницу. При выборе суточного отчёта пользователь переходит во вкладку настройки параметров и выбора режима (автоматический или ручной). В автоматическом – программа сама формирует отчёт в течении 24 часов с шагом в 1 час. В ручном – можно выбрать диапазон отчётного периода.

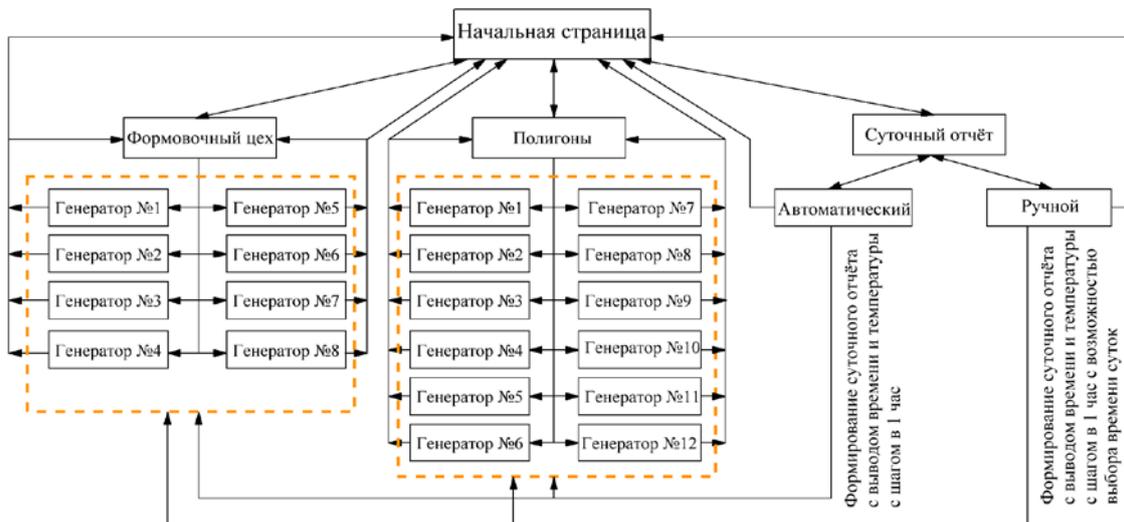


Рисунок 2 – Структура окна визуализации программы

При выборе устройств отображается суточный тренд, диапазон отчёта и шаг. Дополнительной функцией является возможность формирования отчёта по выбранному объекту с пользовательскими настройками (рис. 3).



Рисунок 3 – Суточный тренд по каналу «Формовочный цех. Генератор № 1»

Для реализации выборки режима формирования суточного отчёта написан скрипт на языке ST. Он предназначен для создания отчёта: автоматического или ручного. Скрипт программы представлен ниже:

```

PROGRAM MAIN_PROGRAM

VAR_INPUT
    AutoHand: uint;          (*выборка режима 0 - автоматический, 1 -
ручной*)

    AutoBegin: dt;          (*начало для автоматического режима*)
    AutoEnd: dt;           (*окончание для автоматического*)
    AutoStep: byte;        (*шаг автоматического*)
    AutoTrek: string;      (*путь сохранения автоматического*)
    AutoStart: bool;       (*признак начала автоматического*)

    HandBegin: dt;         (*начало для ручного режима*)
    HandEnd: dt;           (*окончание для ручного*)
    HandStep: byte;        (*шаг ручного*)
    HandTrek: string;      (*путь сохранения ручного*)
    HandStart: bool;       (*признак начала ручного*)
END_VAR

VAR_OUTPUT
    ReportBegin: dt;       (*начало для итогового режима*)
    ReportEnd: dt;         (*окончание для итогового*)
    ReportStep: byte;      (*шаг итогового*)
    ReportTrek: string;    (*путь сохранения итогового*)
    ReportStart: bool;     (*признак начала итогового*)
    ReportText: string;    (*текст в отчете*)
END_VAR

(*выборка итоговых значений*)
case AutoHand of
    0:
ReportBegin:=AutoBegin;ReportEnd:=AutoEnd;ReportStep:=AutoStep;ReportTrek:=Auto

```

```
Trek;ReportStart:=AutoStart;ReportText:= 'Суточный_автоматический_отчет';
1:
ReportBegin:=HandBegin;ReportEnd:=HandEnd;ReportStep:=HandStep;ReportTrek:=Hand
Trek;ReportStart:=HandStart;ReportText:= 'Суточный_ручной_отчет';
end_case;
END_PROGRAM
```

Результаты работы: разработанная программа позволяет дистанционно отслеживать параметры объекта по каналам температуры, создавать несколько видов отчётов, архивировать данные с глубиной архива в 45 дней. Она предназначена для использования в технологическом процессе производства железобетонных конструкций.

Список использованных источников

1. SCADA назначение систем, [Электронный ресурс]. – Режим доступа: http://www.tadviser.ru/index.php/%D0%A1%D1%82%D0%B0%D1%82%D1%8C%D1%8F:SCADA_%D0%BD%D0%B0%D0%B7%D0%BD%D0%B0%D1%87%D0%B5%D0%BD%D0%B8%D0%B5_%D1%81%D0%B8%D1%81%D1%82%D0%B5%D0%BC. – Дата доступа: 21.04.2020.
2. Система автоматического контроля и сбора информации (SCADA) [Электронный ресурс]. – Режим доступа: <http://bourabai.kz/dbt/scada.htm>. – Дата доступа: 21.04.2020.

УДК 004.057.4

ИССЛЕДОВАНИЕ ПРОТОКОЛА HTTP

Черненко Д.В., ст. преп., Гниденко А.К., ст. преп., Панкевич Д.С., студ.

*Витебский государственный технологический университет,
г. Витебск, Республика Беларусь*

Реферат. В статье рассмотрен протокол HTTP, его исследование на примере сайтов *vstu.by* и *sdo.vstu.by*, а также четкая инструкция, по которой любой пользователь, повторив эти действия, сможет узнать, как отправляют и получают запросы/ответы любые сайты.

Ключевые слова: сетевой протокол, HTTP, HTTPS, клиент-сервер, сеть.

HTTP – широко распространённый протокол передачи данных, изначально предназначенный для передачи гипертекстовых документов, то есть документов, которые могут содержать ссылки, позволяющие организовать переход к другим документам. Клиенты и серверы взаимодействуют, обмениваясь одиночными сообщениями, а не потоком данных. Сообщения, отправленные клиентом, обычно веб-браузером, называются запросами, а сообщения, отправленные сервером, называются ответами. HTTP-сообщения – это обмен данными между сервером и клиентом. Есть два типа сообщений: запросы, отправляемые клиентом, чтобы инициировать реакцию со стороны сервера, и ответы от сервера.

Сообщения HTTP состоят из текстовой информации в кодировке ASCII, записанной в несколько строк. В HTTP/1.1 и более ранних версиях они пересылались в качестве обычного текста. В HTTP/2 текстовое сообщение разделяется на фреймы, что позволяет выполнить оптимизацию и повысить производительность.

Веб-разработчики не создают текстовые сообщения HTTP самостоятельно – это делает программа, браузер, прокси или веб-сервер. Они обеспечивают создание HTTP-сообщений через конфигурационные файлы (для прокси и серверов), APIs (для браузеров) или другие интерфейсы.

Так как HTTP это клиент-серверный протокол, соединение всегда устанавливается клиентом. Открыть соединение в HTTP – значит установить соединение через соответствующий транспорт, обычно TCP.

В случае с TCP, в качестве порта HTTP-сервера по умолчанию на компьютере используется порт 80, хотя другие также часто используются, например 8000 или 8080. URL загружаемой страницы содержит доменное имя и порт, который можно и не указывать, если он соответствует порту по умолчанию.