

Сергей Солдатов, Кирилл Наранов

Реализация SMS-информирования персонала средствами ScriptWorX64

Компания «Инсайт-НГ» – молодая на рынке услуг автоматизации, тем не менее, задачи, которые её команда успешно решает, зачастую весьма нетривиальные. Одной из них была организация SMS-информирования оперативного персонала об аварийных событиях в SCADA-системе ICONICS GENESIS64.

Задача возникла на этапе внедрения на одном из объектов компании, но условия её решения были осложнены, поскольку пакет программных средств по согласованию с заказчиком был уже приобретён. Закупка каких-то новых специализированных программных средств была затруднительна и нежела-

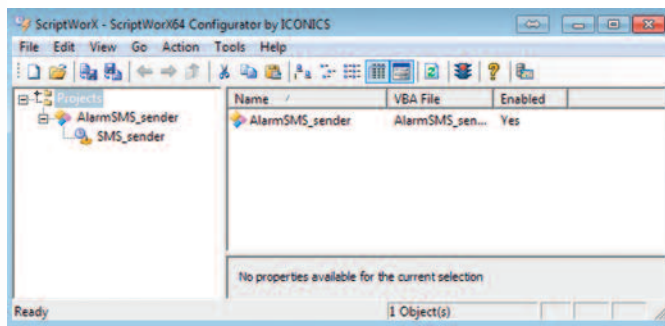


Рис. 1. Интерфейс ScriptWorX64 Configurator

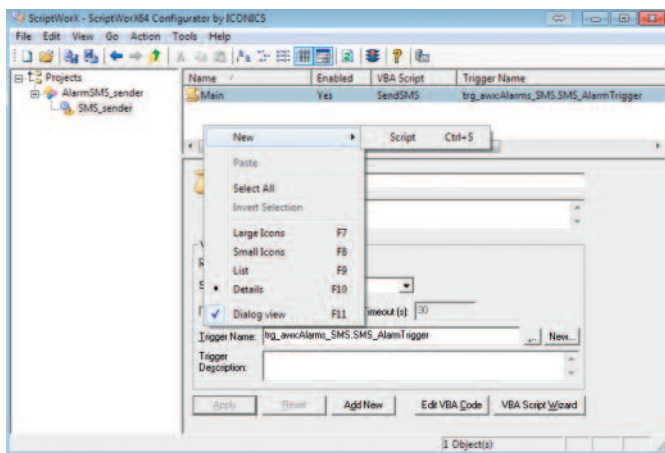


Рис. 2. Создание и настройка скрипта в ScriptWorX64 Configurator

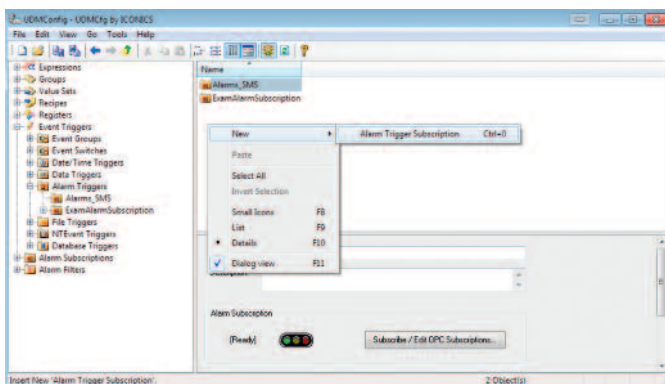


Рис. 3. Создание подписки на аварийные события

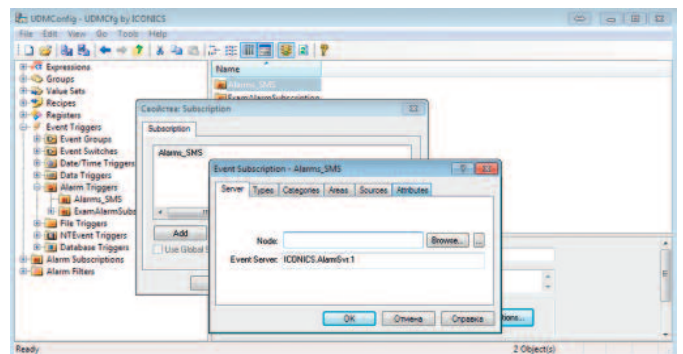


Рис. 4. Настройка подписки на аварийные события в UDMConfig

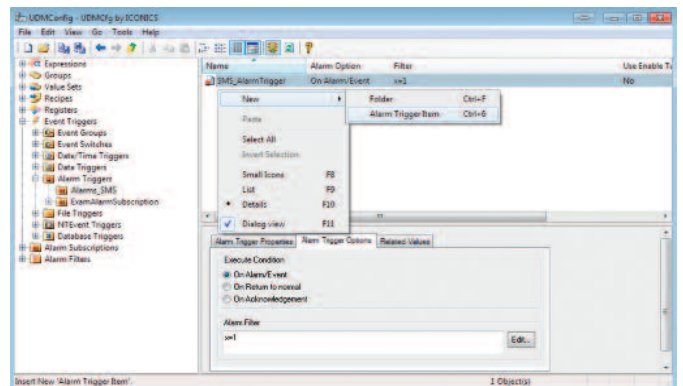


Рис. 5. Создание и настройка триггера в UDMConfig

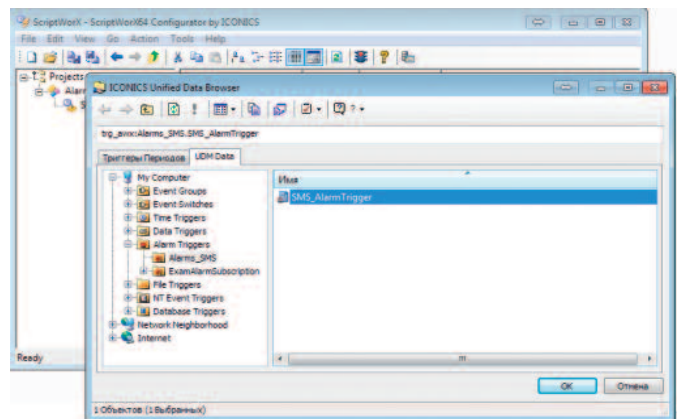


Рис. 6. Выбор созданного триггера для скрипта

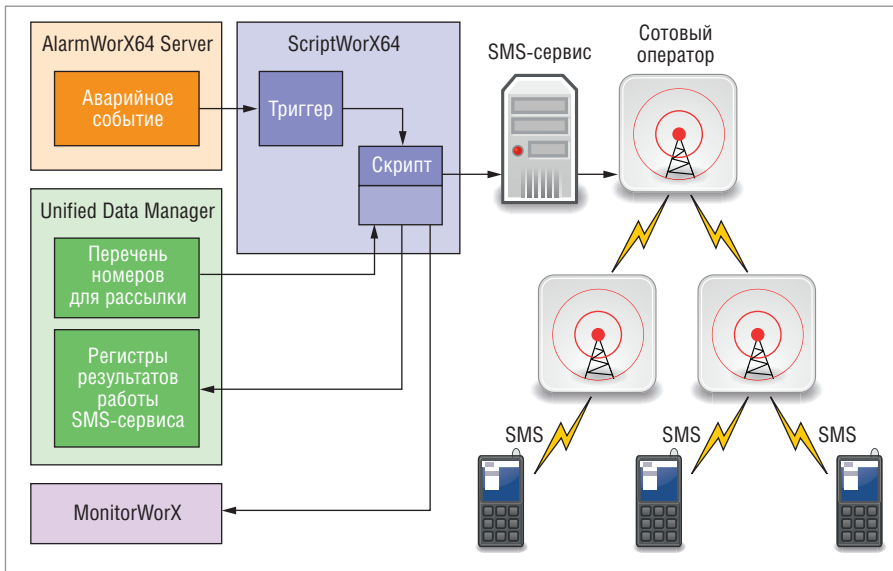


Рис. 7. Структурная схема разрабатываемой системы SMS-информирования

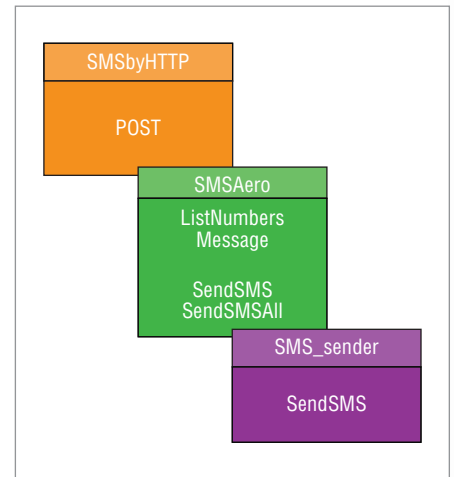


Рис. 8. Структура разработанных классов системы SMS-информирования: курсивом отмечены методы, прямым шрифтом – поля и свойства классов

тельна, поэтому провели анализ документации на приобретённый комплект ICONICS GENESIS64, по итогам которого приняли решение воспользоваться для организации SMS-информирования компонентом ScriptWorX64.

ScriptWorX64 – это сервер приложения на основе скриптов. Он используется для запуска VBA-скриптов по расписанию или как реакции на определённые события (триггеры) и представляет собой многопоточный сервис, предназначенный для работы 24 часа в сутки, готовый выполнить созданные разработчиком сценарии в любой момент.

Далее поэтапно описан процесс конфигурирования ScriptWorX64 и разработки необходимых скриптов для организации SMS-информирования. Данная статья не ставит своей целью подменить руководство ICONICS, поэтому ряд моментов был сознательно опущен, например, описание стандартных процедур в коде VBA-скриптов.

Предварительные настройки

Разработка велась в системе следующей конфигурации:

- Windows 7 Professional x64;
- ICONICS GENESIS64 v.10.71, HotFixPack 1, ServicePack 1, ServicePack 2.

В Unified Data Manager (UDM) были предварительно созданы регистры, а в AlarmWorX64 Server – аварийные сигналы, связанные с регистрами UDM, и Area – объекты для группировки аварийных сигналов, например, для разных объектов.

Поскольку ScriptWorX64 является опциональным компонентом GENESIS64, необходимо выполнить его установку. После установки, к сожалению, выяснилось, что не хватает одной библиотеки – AlarmOle.dll. Выразалось это в невозможности настроить в ScriptWorX64 Configurator подписку на аварийные события. Проблема была успешно решена совместно с сотрудниками компании ПРОСОФТ, они выслали недостающую библиотеку и описали процесс её установки.

На этом этап предварительной настройки завершён, и можно перейти к настройке триггеров ScriptWorX64 на аварийные события.

Настройка триггеров аварийных событий в ScriptWorX64

После установки ScriptWorX64 необходимо запустить ScriptWorX64 Configurator и создать новый проект (рис. 1).

Внутри проекта создаётся объект Thread, в котором создаётся объект Script (рис. 2). Теперь можно перейти непосредственно к созданию триггера аварийных событий. На экранной форме под списком скриптов надо нажать кнопку New напротив Trigger Name, откроется приложение UDMConfig (рис. 3). В нём необходимо выбрать раздел Event Triggers->->Alarm Triggers. Триггеры аварийных событий основаны на отслеживании аварийных событий от заданного сервера (сервер аварий). Связь с сервером аварий называется подпиской. Для создания новой подписки необходимо выбрать в контекстном меню пункт New->Alarm Trigger Subscription (рис. 3). Необходимо будет задать имя подписки, а затем выполнить связку с сервером аварий, для этого нажать кнопку Subscribe/Edit OPC Subscriptions. В открывшемся диалоговом окне (рис. 4) на вкладке Server надо нажать кнопку Browse и выбрать в открывшемся окне сервер аварий. Далее на вкладке Types (рис. 4) был выбран диапазон Severity – уровень аварийных событий, для которых должен срабатывать триггер.

На этом можно было бы и завершить создание подписки триггера, теперь через него будут проходить все аварии с сервера. Но в случае сложного технического объекта, скорее всего, аварии далеко не всех объектов требуют SMS-информирования. На вкладке Areas необходимо выбрать интересующие нас Area на сервере аварий, в которых заранее сгруппированы нужные аварийные сигналы. При необходимости можно задать и остальные настройки.

Теперь надо открыть существующую подписку на аварийные события и создать объект Alarm Trigger Item (рис. 5): задать имя триггера, а на вкладке Alarm Trigger Options в поле Alarm Filter прописать x=1. Это означает, что будут обрабатываться все аварийные события, на которые выполнена подписка. При необходимости можно прописывать расширенные условия, по которым срабатывают триггеры в поле Alarm Filter, их примеры есть в документации на UDMConfig. Альтернативой фильтру будет задание части условий при настройке подписки. Теперь можно всё сохранить и закрыть UDMConfig. Остаётся только связать VBA-скрипт и триггер, для этого на экранной форме, показанной на рис. 2, под списком скриптов нужно нажать кнопку «...» напротив Trigger Name и выбрать созданный триггер алармов (рис. 6).

Теперь можно приступить к созданию VBA-скриптов.

Создание VBA-скриптов

SMS-сообщения возможно отправлять с компьютера различными средствами:

- с использованием GSM-модема;
- с использованием SMS-сервисов в сети Интернет.

В первом случае требуется приобрести модем, установить внешнюю антенну (внутренней антенны при использовании модема внутри помещения может быть недостаточно), приобрести SIM-карту выбранного оператора сотовой связи и разработать скрипты для взаимодействия с модемом.

Второй вариант сводит процесс к двум шагам: выбрать сервис для отправки SMS-сообщений и приобрести пакет SMS-сообщений в Интернете; написать скрипт взаимодействия с сервисом. Важную роль в выборе средства отправки SMS-сообщений сыграло расположение объекта управления на берегу моря, потому что погодные условия могли периодически негативно сказываться на качестве GSM-связи. В то же время для связи с сетью Интернет был проложен в земле оптоволоконный кабель, что гарантировало практически круглосуточную доступность канала связи.

Итак, был выбран вариант с использованием SMS-сервисов в сети Интернет. В русскоязычном сегменте можно найти множество подобных сервисов. Не вдаваясь в детали их различий (в основном отличия в ценовой политике), можно смело сказать, что описанный далее скрипт применим к большинству из них. В данном проекте выбран сервис SMS Aero.

На рис. 7 приведена структурная схема разработанного средства отправки SMS-сообщений. Как видно из неё, при формировании аварийного сообщения срабатывает триггер и запускается скрипт, выполняющий отправку запроса на SMS-

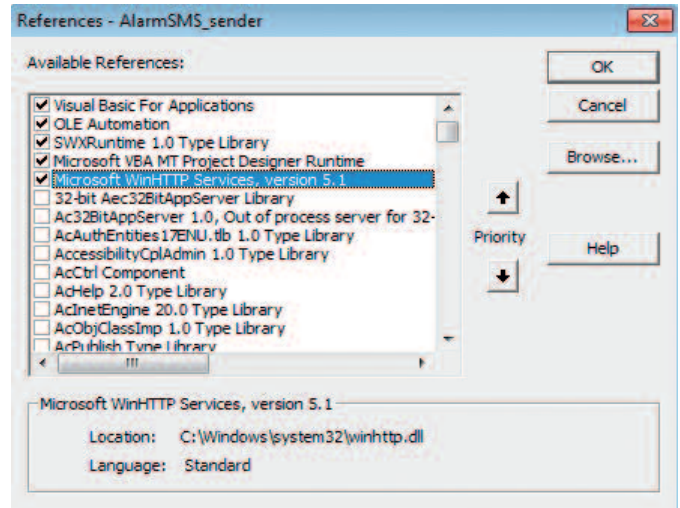


Рис. 9. Подключение компонента Microsoft WinHTTP Service в VBA-проект скрипта

сервис. Далее SMS-сервис по своим каналам связи отправляет SMS-сообщение оператору сотовой связи, который рассылает сообщение по указанным номерам (берутся из переменной, созданной в UDM). При возникновении ошибки в ходе работы скрипта выставляется бит, который заранее создаётся в UDM, а в текстовую переменную, также предварительно созданную, пишется последнее аварийное сообщение, помимо этого, сообщение об ошибке в скрипте пишется в консоль MonitorWorX.

Интерфейс API (Application Programming Interface) SMS-сервиса в общем случае основан на отправке запросов GET



MobileHMI

Мобильная SCADA-система



- Полноценный клиент SCADA-системы на мобильном устройстве
- Легкая навигация с поддержкой технологии multitouch
- Поддержка операционных систем Android, iOS, Windows Phone
- Большое количество используемых интерфейсов: OPC, OPC UA, .NET, SNMP, BACnet, SQL, Oracle
- Наглядные графические инструменты для анализа собираемых данных: графики, диаграммы, pivot-таблицы
- Работа с картографическими сервисами



Управление, визуализация и анализ данных предприятия в Вашем кармане с ICONICS MobileHMI!

ОФИЦИАЛЬНЫЙ ДИСТРИБЬЮТОР ПРОДУКЦИИ ICONICS



Тел.: (495) 234-0636 • Факс: (495) 234-0640 • info@prosoft.ru • www.prosoft.ru



Платформа EuropacPRO — евромеханика ВЫСОКОГО полёта



PROгрессивные блочные каркасы и приборные корпуса

- Безграничное разнообразие конфигураций из унифицированных компонентов
- Современный промышленный дизайн
- Высокая прочность и надёжность
- Доработка под индивидуальные требования

ОФИЦИАЛЬНЫЙ ПОСТАВЩИК ПРОДУКЦИИ SCHROFF



Листинг 1. Функция POST

```
Private objHTTP As WinHttpRequest
' Выполняет отправку запроса на сервер
' URL - запрос
Public Function POST _
(ByRef URL As String) As String
    On Error GoTo ErrHandler
    With objHTTP
        .Open "GET", URL
        .Send
        POST = .ResponseText
    End With
    Exit Function
ErrHandler:
    ErrHandlerSub "'SMSbyHTTP' POST failed"
End Function
```

Листинг 2. Процедура ErrHandlerSub

```
Public g As SWXRuntimeLib.Global
Const SMS_cfg = _
"ICONICS.FwxServerOPC.1\args64:SMS_config."
' Обработчик аварийных сообщений
' Пишет в консоль MonitorWorX,
' в переменную SMS_Alarmlog
' Выставляет бит аварии при отправке SMS
' ErrorMessage - аварийное сообщение
Private Sub ErrHandlerSub(ByRef ErrorMessage As String)
    g.ConsoleMsg MSG_SEVERE_ERROR, "Designer", _
        ErrorMessage
    g.OPC.Write SMS_cfg & "SMS_Alarmlog.Value", _
        ErrorMessage
    g.OPC.Write SMS_cfg & "SMS_Alarm.Value", 1
End Sub
```

Листинг 3. Процедура SendSMS класса SMSAero

```
Public Message As String
' адрес сервера SMS Aero
Const gate_url = "http://gate.smsaero.ru/"
' запрос на отправку
Const sendcmd = gate_url + "send/"
' запрос на получение статуса сообщения
Const status = gate_url + "status/"
' пароль зашифрованный MD5,
' требование сервиса SMS Aero
Const password = "XXXXXXXX"
' имя пользователя
Const user = "USER_NAME"
' подпись отправителя
Const sender = "SENDER"

' Процедура отправки SMS на заданный номер Number
Public Sub SendSMS(ByRef Number As String)
    Dim res As String
    On Error GoTo ErrHandler
```

```
    g.ConsoleMsg MSG_INFORMATION, "Designer", _
    "'SMSAero' Send message '" & Message & "' to '" & Number
    ' Формируем строку запроса к серверу и
    ' передаём объекту SMSSender
    res = SMSSender.POST(sendcmd & "?" & _
        "user=" & user & "&" & _
        "password=" & password & "&" & _
        "to=" & Number & "&" & _
        "text=" & Message & "&" & _
        "from=" & sender)
    ' Штатно должно вернуться два элемента,
    ' разделённых '='
    ' Если вернулся один, значит ошибка
    If InStr(1, res, "=") = 0 Then
        ErrHandlerSub "'SMSAero' SendSMS warning '" & res
    Else
        g.ConsoleMsg MSG_INFORMATION, "Designer", _
        "'SMSAero' Message to '" & Number & "' sent"
    End If
    Exit Sub
ErrHandler:
    ErrHandlerSub "'SMSAero' SendSMS failed"
End Sub
```

Листинг 4. Процедура SendSMS модуля SMS_sender

```
' Script SendSMS
Public Sub SendSMS(td As TriggerData)
    On Error GoTo ErrHandler
    SMSAero_obj.Message = Now & ";" & td.Attributes(3)
    SMSAero_obj.SendSMSAll
    Exit Sub
ErrHandler:
    ErrHandlerSub "'SMS_sender' SendSMS failed"
End Sub
```

Листинг 5. Процедура GlobalInstance_OnConnection

```
' Called once when a new designer/thread is started
Private Sub GlobalInstance_OnConnection _
    (ByVal SwxGlobal As SWXRuntimeLib.Global, _
    ByVal MTD11Inst As Object, Custom() As Variant)

    On Error GoTo ErrHandler
    Set g = SwxGlobal
    Set SMSAero_obj = New SMSAero
    Set SMSAero_obj.g = g
    ' Read the numbers for SMS.
    ' Переменная SMS_cfg см. листинг 2.
    SMSAero_obj.ListNumbers = _
        g.OPC.Read(SMS_cfg & "SMS_Numbers.Value")
    Exit Sub
ErrHandler:
    ErrHandlerSub "'SMS_sender' OnConnection failed"
End Sub
```

или POST по HTTP-протоколу на сервер SMS-сервиса. В строке запроса указываются имя и пароль клиента SMS-сервиса, номер телефона, на который идёт отправка, текст сообщения и имя отправителя (задаётся в личном кабинете клиента).

Поскольку используемый в ScriptWorX64 язык VBA (Visual Basic for Applications) частично является объектно-ориентированным (полноценное наследование классов, к сожалению, не реализовано), то была предложена схема скрипта, представленная на рис. 8.

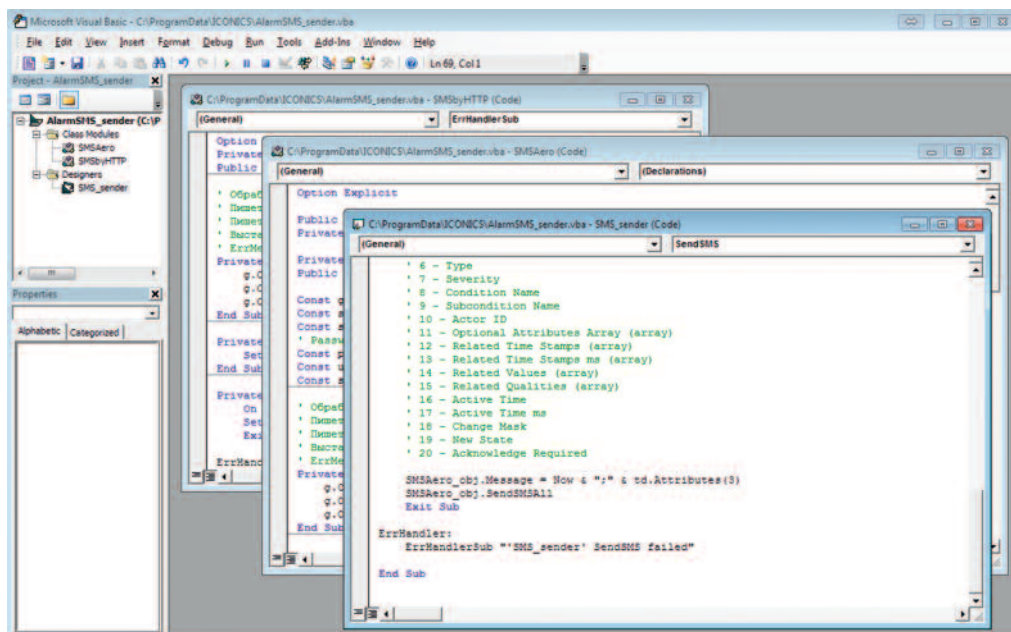


Рис. 10. VBA-проект разработанного скрипта SMS-информирования

Был создан базовый класс – SMSbyHTTP, который выполняет отправку запросов SMS-сервису по протоколу HTTP. Для работы класса необходимо добавить в VBA-проект ссылку на объект Microsoft WinHTTP Services, version 5.1 (рис. 9). Создание объекта Microsoft WinHTTP Services проводится при инициализации класса SMSbyHTTP. По сути, разработанный класс выполняет подготовку запроса по HTTP-протоколу и возвращает результат – ответ сервера SMS-сервиса. Он имеет только функцию POST, код которой приведён в листинге 1. Поскольку при отправке сообщений возможны нештатные ситуации, разработана процедура обработки аварийных сообщений ErrHandlerSub (листинг 2), аналогичная процедура используется и в других классах. ErrHandlerSub использует объект g типа SWXRuntimeLib.Global из основного кода скрипта.

Далее был создан зависимый от SMSbyHTTP класс – SMSAero, который определяет порядок взаимодействия уже с конкретным SMS-сервисом, в данном случае SMS Aero. У класса есть свойство ListNumbers, оно представляет собой массив номеров, по которым должна быть сделана рассылка. При присвоении свойству строки, содержащей мобильные номера, разделённые «;», происходит автоматический разбор строки и формирование массива номеров. Также в классе SMSAero есть поле Message – собственно отправляемое сообщение. В классе реализованы две процедуры: SendSMS отправляет сообщение Message на заданный номер, SendSMSAll рассылает сообщение Message по всем номерам, содержащимся в ListNumbers, для этого используется процедура SendSMS. Код процедуры SendSMS класса SMSAero приведён в листинге 3.

Основной код скрипта размещён в модуле SMS_sender, в нём создана процедура SendSMS. Именно её надо указать в поле Script на экранной форме под списком скриптов в ScriptWorX64 Configurator. Данная процедура получает в качестве параметра объект TriggerData, содержимое которого зависит от типа триггера. Для процедуры SendSMS требуется только атрибут № 3 (нумерация атрибутов идёт с 0), он содержит текстовое описание аварии, получаемое из AlarmWorX64 Server. Время аварии из TriggerData брать нежелательно, поскольку оно хранится без учёта часового

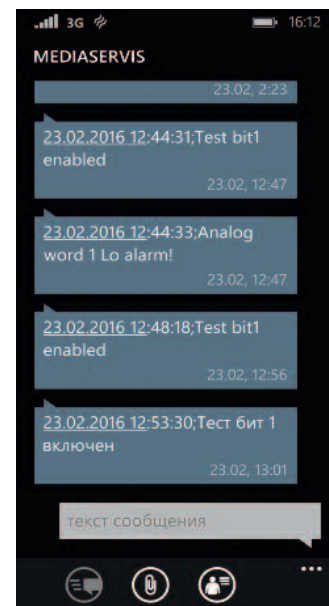


Рис. 11. Примеры сгенерированных SMS-сообщений на смартфоне

пояса. Код процедуры SendSMS из модуля SMS_sender приведён в листинге 4. Также в модуле SMS_sender при инициализации скрипта происходит получение списка номеров для рассылки (листинг 5).

После окончания разработки остаётся проверить состав VBA-проекта (рис. 10) и сохранить его.

Названия переменных в UDM, с которыми работает приведённый скрипт, следующие:

- SMS_config.SMS_Numbers – строка, содержащая номера, на которые делаем рассылку, они разделены точкой с запятой;
- SMS_config.SMS_Alarmlog – строка, содержащая последнее аварийное сообщение от скрипта;
- SMS_config.SMS_Alarm – бит, установленный в 1, если произошла авария при работе скрипта.

После сохранения результатов разработки необходимо выполнить запуск скрипта в ScriptWorX64. Для этого надо выбрать созданный скрипт в ScriptWorX64 Configurator и нажать кнопку в виде светофора на панели сверху (рис. 2), чтобы он стал зелёным, при этом запустится ScriptWorX Runtime.

Ход работы скрипта можно отслеживать в MonitorWorX, там будут следующие типы сообщений:

- 1) сообщения об ошибке в работе скрипта;
- 2) сообщение об отправке SMS-сообщения;
- 3) сообщение об успешной отправке SMS-сообщения.

Примеры полученных сообщений приведены на рис. 11, они содержат всю необходимую информацию: время и дату возникновения аварийного события и его описание.

ЗАКЛЮЧЕНИЕ

Результатом работы стало создание простого и прозрачного механизма SMS-информирования оперативного персонала. При этом были использованы только встроенные средства ICONICS GENESIS64, что лишний раз показывает широчайшие возможности данной SCADA-системы.

Разработанный код является во многом универсальным и может быть легко модернизирован для организации отправки SMS-сообщений через любой известный SMS-сервис. ●

E-mail: ssa-company@rambler.ru