NaftaVision®

НЕФТЕАВТОМАТИКА

Руководство по настройке интеграции

2025

Содержание

1. Условные обозначения и термины

1.1. Условные обозначения

Внимание:

Помечает информацию, с которой необходимо ознакомиться, чтобы учесть особенности работы какого-либо элемента программного обеспечения.

ОСТОРОЖНО:

Помечает информацию, с которой необходимо ознакомиться, чтобы предотвратить нарушения в работе программного обеспечения либо предотвратить потерю данных.

ОПАСНО:

Помечает информацию, с которой необходимо ознакомиться, чтобы избежать потери контроля над технологическим процессом.

1.2. Перечень терминов и сокращений

Проект

Набор данных, который представляет конфигурацию SCADA.

2. Введение

Документ "Руководство по настройке интеграции" (далее Руководство) относится к комплекту методической документации для использования программного обеспечения (ПО), и предназначена для настройки интеграции, то есть процесса организации и настройки взаимодействия ПО и сторонних информационных систем.

 **Внимание:** Справочная информация доступна:

- из главного меню командой **Помощь** > **Справка**;
- по клавише **“F1”**;
- выбором пункта **Справка** из контекстного меню дерева проекта.

3. Извлечение данных посредством ODBC

ODBC представляет собой программный интерфейс (API) доступа к источнику данных, который позволяет взаимодействовать и обмениваться с различными хранилищами данных, совместимых с ODBC.

Примечание: ODBC драйвер входит в комплект поставки системы.

3.1. Строка подключения

Строка подключения - это строка, которая содержит информацию, необходимую для подключения.

Для подключения ODBC драйвера к источнику данных необходимо настроить строку подключения:

```
Connection.Open "DSN=Jazz DSN;UID=Admin;PWD=abc123; Ip=127.0.0.1;Port=48012;"
```

где:

- DSN=Jazz DSN; – имя источника данных (DSN), используемого для описания соединения с источником данных (обязательный параметр);
- UID=Admin;PWD=abc123; – логин, пароль для доступа к серверу (обязательные параметры);
- Ip=127.0.0.1;Port=48012; – ip адрес и порт сервера. По умолчанию серверу присваиваются ip 127.0.0.1, порт 48012 (опциональные параметры).

3.2. Оператор запроса SELECT

SELECT - оператор запроса в языке SQL, возвращающий набор данных (выборку) из источника данных. Для формирования запроса ODBC драйвера к источнику данных необходимо создать SQL запрос:

```
strQuery = "SELECT * FROM History WHERE nodeId=
 'ns=2;s=AnalogOutputPoint_001.Output' AND sourceTimestamp > '2019-04-18
 17:16:40' AND sourceTimestamp < '2019-04-18 17:47:41'"
```

где:

- SELECT * FROM History – добавьте взамен символа «*» наименования столбцов таблицы через запятую из предложенных ниже:

- serverTimestamp – серверная временная метка;
- sourceTimestamp – временная метка источника;
- value – значение тега OPC UA (ноды);
- statusCode – статус-код.

В драйвере ODBC используется предопределенный формат запроса к историческим данным. Обязательные параметры:

- nodeId – имя тега в нотации OPC UA;
- sourceTimestamp – временная метка источника данных в формате 'YYYY-MM-DD HH:MM:SS'. Необходимо указывать обе временные границы среза исторических данных

3.3. Создание макроса в Excel

ODBC драйвер позволяет извлечь исторические данные с сервера ввода/вывода и сохранять их в электронные таблицы, например в MS Excel, посредством VBA-скриптов (макросов).

Внимание: ODBC драйвер может функционировать только на 64-разрядной версии MS Excel.

Для создания макроса в MS Excel выполните следующие действия:

1. В окне MS Excel нажмите кнопку **Макрос**.

2. Нажмите кнопку **Создать**.

3. В открывшемся диалоговом окне введите имя макроса.

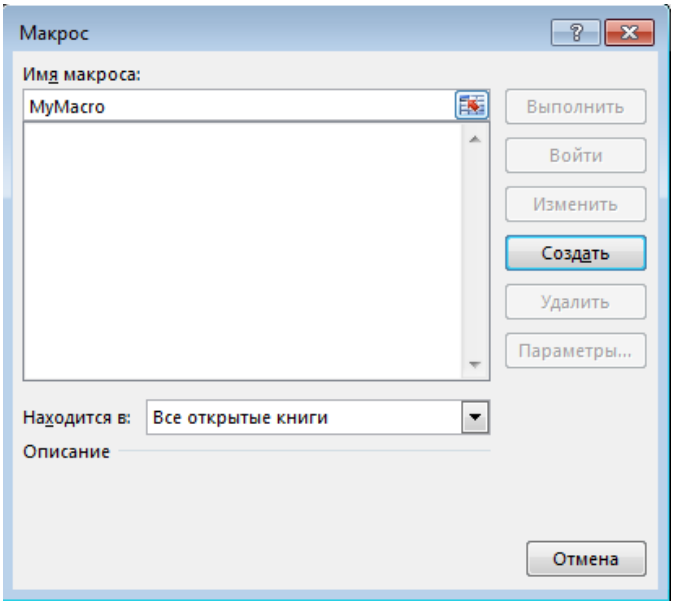
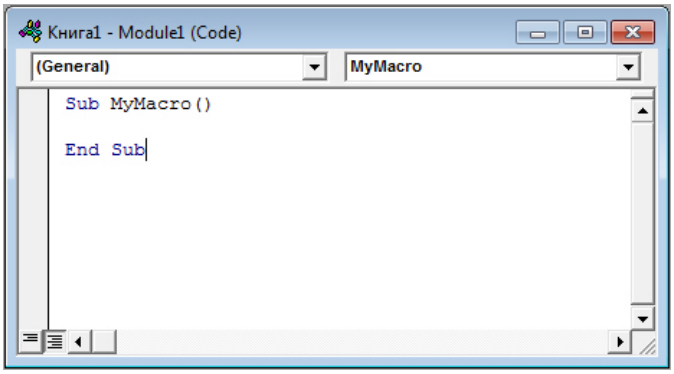


Рисунок 1. Создание макроса в Excel

4. Откроется редактор Visual Basic (далее VB) и окно программного кода с уже введенной структурой нового макроса.

Рисунок 2. Окно программного кода редактора Visual Basic



5. Введите между строчками **Sub <Имя макроса>** и **End Sub** VBA-скрипт для извлечения данных. Пример кода с пояснениями представлен ниже:

Таблица 1. Пример кода с пояснениями

Фрагмент кода	Описание
Set Connection = CreateObject("ADODB.Connection")	Создание объекта подключения
Connection.Open "DSN=Jazz DSN;UID=Admin;PWD=abc123; Ip=127.0.0.1;Port=48012;"	Формирование строки подключения (см. раздел 3.1. Строка подключения)

Фрагмент кода	Описание
strQuery = "SELECT * FROM History WHERE nodeId= 'ns=2;s=AnalogOutputPoint_001.Output' AND sourceTimestamp > '2019-04-18 17:16:40' AND sourceTimestamp < '2019-04-18 17:47:41'"	Создание SQL запроса (см. раздел 3.2. Оператор запроса SELECT)
Set resultSet = Connection.Execute(strQuery)	Получение результата. Результат получается в виде курсора на результирующую таблицу. Произвольный доступ к результирующей таблице через курсор недоступен
resultSet.Fields(<номер столбца>). Name	Извлечение имени столбцов
resultSet.Fields(<номер столбца>). Value	Извлечение значений столбцов
resultSet.movenext	Переход на следующую строку результирующей таблицы
resultSet.Close	Окончание работы с результирующей таблицей

Внимание: Переменные DSN, UID, PWD, Ip, Port, nodeId, serverTimestamp, sourceTimestamp, value, statusCode, History чувствительны к регистру - их необходимо записывать так, как представлено в примере.

6. На рисунке приведен пример формирования VBA-скрипта.

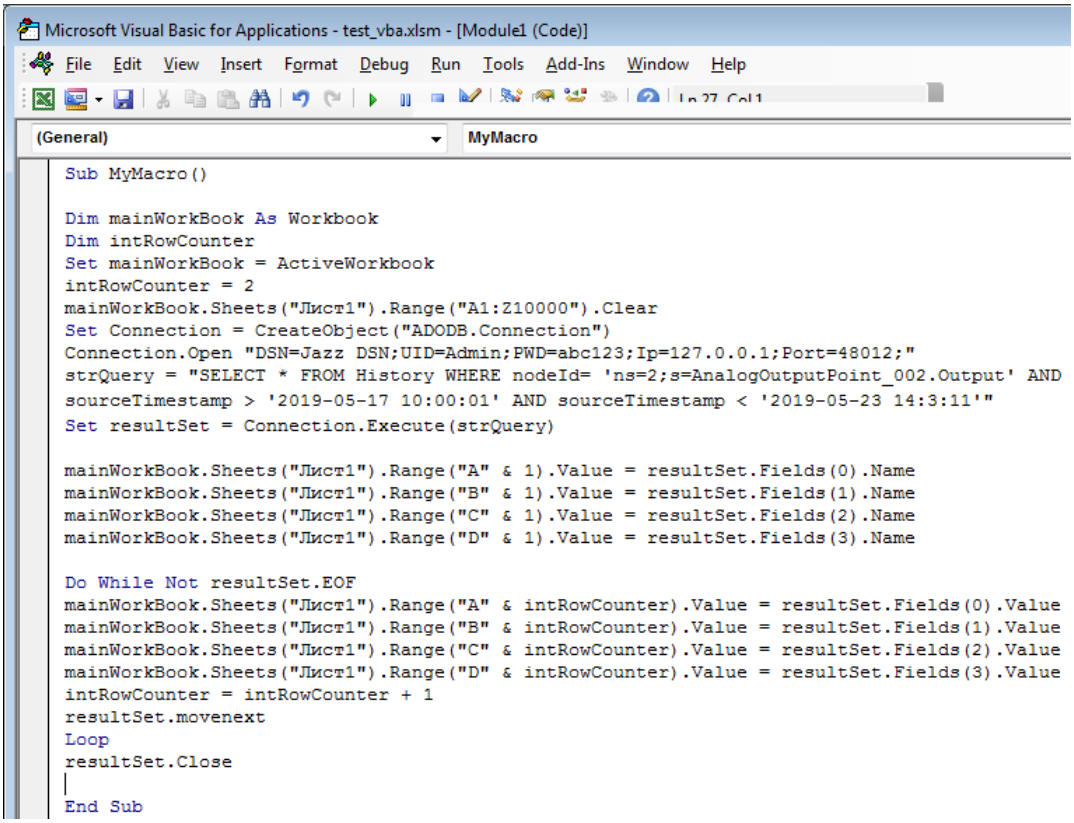


Рисунок 3. Пример VBA-скрипта

7. Запуск макроса по имени приведет к выполнению VBA-скрипта. Для запуска макроса из MS Excel в окне **Макрос** нажмите кнопку **Выполнить**.

На рисунке приведен пример результата запуска вышеуказанного VBA-скрипта.

	A	B	C	D
1	serverTimestamp	sourceTimestamp	value	statusCode
2	22.05.2019 9:50:14	22.05.2019 9:50:12	1,24025	0
3	22.05.2019 9:50:22	22.05.2019 9:50:22	1,54750	2159083520
4	22.05.2019 9:50:25	22.05.2019 9:50:23	2,15780	0
5	22.05.2019 9:50:33	22.05.2019 9:50:33	1,35970	2159083520
6	22.05.2019 9:50:36	22.05.2019 9:50:34	1,68780	0
7	22.05.2019 9:50:44	22.05.2019 9:50:44	1,13978	0
8	22.05.2019 9:50:47	22.05.2019 9:50:45	1,25700	0
9	22.05.2019 9:50:55	22.05.2019 9:50:55	2,23540	0
10	22.05.2019 9:50:58	22.05.2019 9:50:56	2,21543	0
11	22.05.2019 9:51:06	22.05.2019 9:51:06	2,73216	0
12	22.05.2019 9:51:09	22.05.2019 9:51:07	2,58914	0
13	22.05.2019 9:51:17	22.05.2019 9:51:17	2,57730	0
14	22.05.2019 9:51:20	22.05.2019 9:51:18	2,68710	0
15	22.05.2019 9:51:28	22.05.2019 9:51:28	2,77830	0
16	22.05.2019 9:51:31	22.05.2019 9:51:29	2,13147	0
17	22.05.2019 9:51:39	22.05.2019 9:51:39	2,46565	0
18	22.05.2019 9:51:42	22.05.2019 9:51:40	3,79830	0
19	22.05.2019 9:51:50	22.05.2019 9:51:50	0,00014	2159083520
20	22.05.2019 9:51:53	22.05.2019 9:51:51	0,00014	0
21	22.05.2019 9:52:01	22.05.2019 9:52:01	0,00014	2159083520
22	22.05.2019 9:52:04	22.05.2019 9:52:02	0,00014	0
23	22.05.2019 9:52:12	22.05.2019 9:52:12	0,00014	2159083520
24	22.05.2019 9:52:16	22.05.2019 9:52:13	0,00014	0
25	22.05.2019 9:52:24	22.05.2019 9:52:24	0,00014	2159083520
26	22.05.2019 9:52:27	22.05.2019 9:52:24	0,00014	0
27	22.05.2019 9:52:35	22.05.2019 9:52:35	0,00000	2159083520
28	22.05.2019 9:52:38	22.05.2019 9:52:35	0,00000	2159083520

Рисунок 4. Пример таблицы Excel

3.4. Пример создания отчета

В разделе представлен пример VBA-скрипта, предназначенного для формирования отчета.

Отчет предоставляет ежемесячные выходные данные за год по параметрам: TIR, FIR, PIR.

Чтобы задать отчет необходимо выполнить следующие действия:

1. В редакторе VB с уже введенной структурой нового макроса введите VBA-скрипт для извлечения данных.
2. Пример формирования VBA-скрипта приведен ниже:

```
Dim connection

Public Const CurrentSheet As String = "sheet"
Public Const FocusedTag1 As String = "A5"
Public Const FocusedTag2 As String = "A6"
Public Const FocusedTag3 As String = "A7"

Public Const FocusedTag4 As String = "A9"
Public Const FocusedTag5 As String = "A10"
Public Const FocusedTag6 As String = "A11"

Public Const FocusedTag7 As String = "A13"
```

```
Public Const FocusedTag8 As String = "A14"
Public Const FocusedTag9 As String = "A15"

Public Const FocusedTag10 As String = "A17"
Public Const FocusedTag11 As String = "A18"
Public Const FocusedTag12 As String = "A19"

Public Const FocusedTag13 As String = "A21"
Public Const FocusedTag14 As String = "A22"
Public Const FocusedTag15 As String = "A23"

'Main Summary Configuration

Public Const AVARAGE_GROUP As String = FocusedTag1 + "," + FocusedTag3 + ","
 + FocusedTag4 + "," + FocusedTag6 + "," + FocusedTag7 + "," + FocusedTag9
 + "," + FocusedTag10 + "," + FocusedTag12 + "," + FocusedTag13 + "," +
 FocusedTag15
Public Const TOTAL_GROUP As String = FocusedTag2 + "," + FocusedTag5 + "," +
 FocusedTag8 + "," + FocusedTag11 + "," + FocusedTag14
Public Const FOCUSED_DATE_RANGE As String = "C3:N3"

Sub Report()

Dim val As Variant

Set connection = CreateObject("ADODB.Connection")
connection.Open "DSN=Jazz
 DSN;UID=Admin;PWD=abc123;Ip=10.155.26.195;Port=48012;"

For Each EndTimeCell In Worksheets(CurrentSheet).Range(FOCUSED_DATE_RANGE)
For Each tag In Worksheets(CurrentSheet).Range(AVARAGE_GROUP)
nextTimePoint = EndTimeCell.Value
startTimePoint = DateAdd("m", -1, nextTimePoint)

'MsgBox Format(startTimePoint, "yyyy-mm-dd hh:mm:ss") + " and " +
 Format(nextTimePoint, "yyyy-mm-dd hh:mm:ss")

Worksheets(CurrentSheet).Cells(tag.Row, EndTimeCell.Column).Value =
 GetAvarageValue(connection, tag.Value + ".Output", Format(startTimePoint,
 "yyyy-mm-dd hh:mm:ss"), Format(nextTimePoint, "yyyy-mm-dd hh:mm:ss"))
Next tag
For Each tag In Worksheets(CurrentSheet).Range(TOTAL_GROUP)
nextTimePoint = EndTimeCell.Value
startTimePoint = DateAdd("m", -1, nextTimePoint)

Worksheets(CurrentSheet).Cells(tag.Row, EndTimeCell.Column).Value =
 GetTotal(connection, tag.Value + ".Output", Format(startTimePoint, "yyyy-
mm-dd hh:mm:ss"), Format(nextTimePoint, "yyyy-mm-dd hh:mm:ss"))
Next tag

Next EndTimeCell

'MsgBox "Расчет отчета окончен"

End Sub

Function GetTotal(connection, TagName, StartTime, EndTime)
On Error GoTo DeleteCustomerError
totalOfTagValues = 0

strQuery = "SELECT value FROM History WHERE nodeId='ns=2;s=" & TagName &
 "' AND sourceTimestamp > '" & StartTime & "' AND sourceTimestamp < '" &
 EndTime & "'"
Set resultSet = connection.Execute(strQuery)

Do While Not resultSet.EOF
totalOfTagValues = totalOfTagValues + resultSet.Fields(0).Value
resultSet.movenext
```

```
Loop

GetTotal = totalOfTagValues
Exit Function
DeleteCustomerError:
ErrorsHandler
GetTotal = totalOfTagValues
End Function

Public Function GetAvarageValue(connection, TagName, StartTime, EndTime)
On Error GoTo DeleteCustomerError
result = 0
strQuery = "SELECT value FROM History WHERE nodeId='ns=2;s=" & TagName &
 "' AND sourceTimestamp > '" & StartTime & "' AND sourceTimestamp < '" &
 EndTime & "'"

Set resultSet = connection.Execute(strQuery)

amount = 0
Do While Not resultSet.EOF
Avarage = Avarage + resultSet.Fields(0).Value
amount = amount + 1
resultSet.movenext
Loop

resultSet.Close
result = Avarage / amount
GetAvarageValue = result
Exit Function
DeleteCustomerError:
ErrorsHandler
GetAvarageValue = result
End Function

Function ErrorsHandler()
Set objError = CreateObject("ADODB.Error")
Dim strError As String

If connection.Errors.Count > 0 Then
For Each objError In connection.Errors
strError = strError & "Error #" & objError.Number & _
" " & objError.Description & vbCrLf & _
"NativeError: " & objError.NativeError & vbCrLf & _
"SQLState: " & objError.SqlState & vbCrLf & _
"Reported by: " & objError.Source & vbCrLf & _
"Help file: " & objError.HelpFile & vbCrLf & _
"Help Context ID: " & objError.HelpContext

Next

End If

'MsgBox strError

End Function
```

На рисунке приведен результат запуска вышеуказанного VBA-скрипта:

	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O
1										Годовой отчет					1 ноября 2019 г.
2															
3	Тег	Описание	Декабрь	Январь	Февраль	Март	Апрель	Май	Июнь	Июль	Август	Сентябрь	Октябрь	Ноябрь	Расчет
4	Группа оборудования насоса H1														
5	TIR001	Температура H1	0	0	0	0	0	0	0	0	0	0	0	61,4985888	5,124882401
6	FIR002	Расход на всасе H1	0	0	0	0	0	0	0	0	0	0	0	848554	848554
7	PIR003	Давление на всасе H1	0	0	0	0	0	0	0	0	0	0	0	49,99865	4,166554163
8	Группа оборудования насоса H2														
9	TIR004	Температура H2	0	0	0	0	0	0	0	0	0	0	0	61,4982081	5,124850673
10	FIR005	Расход на всасе H2	0	0	0	0	0	0	0	0	0	0	0	847618	847618
11	PIR006	Давление на всасе H2	0	0	0	0	0	0	0	0	0	0	0	49,9965323	4,166377695
12	Группа оборудования насоса H3														
13	TIR007	Температура H3	0	0	0	0	0	0	0	0	0	0	0	61,4981105	5,124842543
14	FIR008	Расход на всасе H3	0	0	0	0	0	0	0	0	0	0	0	847408	847408
15	PIR009	Давление на всасе H3	0	0	0	0	0	0	0	0	0	0	0	49,9956949	4,166307911
16	Группа оборудования насоса H4														
17	TIR010	Температура H4	0	0	0	0	0	0	0	0	0	0	0	61,4980145	5,124834544
18	FIR011	Расход на всасе H4	0	0	0	0	0	0	0	0	0	0	0	847439	847439
19	PIR012	Давление на всасе H4	0	0	0	0	0	0	0	0	0	0	0	49,9962452	4,166353765
20	Группа оборудования насоса H5														
21	TIR013	Температура H5	0	0	0	0	0	0	0	0	0	0	0	61,4983787	5,12486489
22	FIR014	Расход на всасе H5	0	0	0	0	0	0	0	0	0	0	0	846887	846887
23	PIR015	Давление на всасе H5	0	0	0	0	0	0	0	0	0	0	0	49,9959584	4,16632987

Рисунок 5. Годовой отчет

4. Программный интерфейс приложения

4.1. API gRPC

Внимание: Порт **gRPC** сервера - 44800.

gRPC[1] - это фреймворк для удаленного вызова процедур (RPC). **gRPC** доступен для использования во всех популярных языках программирования[2].

Для доступа к серверу **gRPC** NaftaVision сгенерируйте код клиента[3][4], используя прилагаемый proto-файл "Nafta.proto". Сгенерированные классы содержат простейшие методы доступа ко всем полям типа get/set, а также методы для сериализации и десериализации вашей структуры данных в/из массива байтов. Ниже приведен пример proto-файла:

```
syntax = "proto3";

option java_multiple_files = true;
option java_package = "nft.dcs.grpc";
option java_outer_classname = "NaftagRPCProto";
option objc_class_prefix = "Nafta";

service Nafta
{
// NaftaVision API
// Retrieves a current user information.
// Получает информацию о текущем пользователе
    rpc getCurrentUserInfo(CurrentUserInfoRequest) returns (CurrentUserInfo) {};
}

// NaftaVision API
// Request a current user information.
// Запрос информации о текущем авторизованном пользователе
message CurrentUserInfoRequest {}

// NaftaVision API
// Reply as: a current user information.
// Ответ в виде: информации о текущем авторизованном пользователе
message CurrentUserInfo
{
    string userName = 1;
    string userGroup = 2;
    int32 accessLevel = 3;
}
```

[1] https://grpc.io/docs/what-is-grpc/introduction/
[2] https://grpc.io/docs/languages/
[3] https://grpc.io/docs/languages/java/quickstart/
[4] https://grpc.io/docs/languages/java/basics/