

**DR. BOB DAVIDOV**

## **МатЛАБ OPC HDA клиент**

*Цель работы:* освоение правил накопления OPC данных HDA сервером и их отображение MatLAB клиентом

*Задача работы:* чтение и отображения HDA данных OPC сервера в среде MatLAB.

*Приборы и принадлежности:* персональный компьютер, MatLAB с OPC Toolbox, OPC DA/HDA сервер компании ИнСАТ (демонстрационная версия на 32 тега).

### **ВВЕДЕНИЕ**

Современные процессы обязывают специалистов как можно быстрее проводить анализ больших объемов данных с использованием мощных аналитических алгоритмов. Доступ к сохраненным OPC HDA данным при помощи OPC Toolbox MatLAB позволяет эффективно производить анализ данных и их визуализацию, генерировать отчеты в MATLAB, автоматизировать выполнение общих процедур и записывать результаты работы обратно на сервер.

### **ОБЩИЕ СВЕДЕНИЯ**

Целью OPC HDA сервера является предоставление клиентской программе единого интерфейса для обмена данными с любыми хранилищами данных, в качестве которых может выступать нестандартный файл с данными, стандартная СУБД, OPC DA сервер или другой OPC HDA сервер. Стандарт распространяется только на интерфейсы COM объекта и методы его использования для взаимодействия HDA сервера с клиентскими программами и не устанавливает способов получения или хранения данных.

OPC клиент может подсоединяться к нескольким OPC HDA серверам разных производителей и быть установлен на разных компьютерах в сети Ethernet.

Существует два типа HDA серверов:

- простой сервер данных предыстории для построения графиков (трендов);
- сервер для хранения данных в упакованном виде с возможностью их обработки и анализа. К функциям обработки и анализа данных относятся нахождение среднего, минимального и максимального значения и др.

Работа с данными заключается в чтении, записи или изменении данных.

## ПОРЯДОК ВЫПОЛНЕНИЯ РАБОТЫ

Конфигурация OPC сервера, необходимая для выполнения работ: **Simulator.mbc**

**Задание 1.** Установка необходимого программного обеспечения.

1. Если OPC Foundation Core Components пакета МатЛАБ не установлен введите команду OPCREGISTER для их установки. Команду вводите через “Окно команд” МатЛАБ.

Синтаксис команды OPCREGISTER:

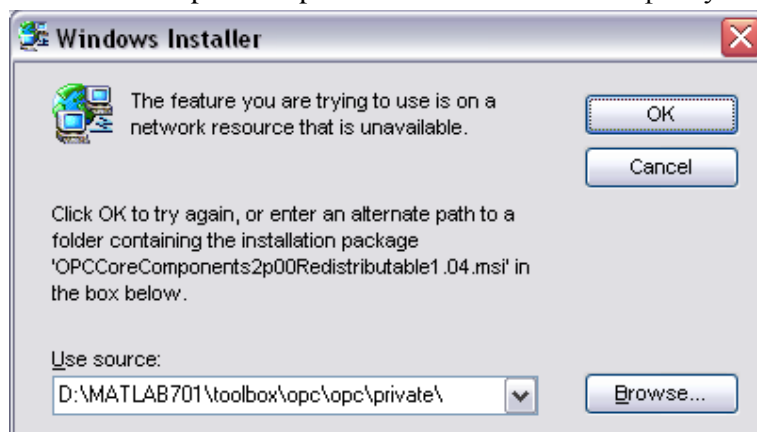
```
>> opcregister
>> opcregister('repair')
>> opcregister('remove')
```

Если после ввода команды появляется следующее сообщение,

Warning: Could not find an exact (case-sensitive) match for 'OPCREGISTER'.

C:\MATLAB701\toolbox\opc\opc\opcregister.m is a case-insensitive match and will be used instead. You can improve the performance of your code by using exact name matches and we therefore recommend that you update your usage accordingly. Alternatively, you can disable this warning using warning('off', 'MATLAB:dispatcher:InexactMatch').

Необходимо правильно указать путь “..\opc\private\” к OPCCoreComponents2p00Redistributable1.04.msi файлу в окне “Use source”



Установка ModBUS OPC DA/HDA сервера компании ИнСАТ

2. Установите демонстрационную версию OPC сервера на 32 бита компании ИнСАТ MODBUS OPC SERVER SETUP DEMO32TAGS.exe

**Задание 2.** Отработка команд связи с OPC сервером.

1. Найдите OPC Historical Data Access (HDA) сервера на вашем компьютере. У хост (host) компьютера можно запросить информацию о всех доступных OPC HDA серверах.

```
>> hostInfo = opc.hda.getServerInfo('localhost')
```

```
hostInfo =
```

```
1x3 OPC HDA ServerInfo array:
```

index	Host	ServerID	HDASpecification	Description
-------	------	----------	------------------	-------------

```

1 localhost MasterSCADA.OPCHistoryServer.1 HDA1      MasterSCADA OPC History Server
2 localhost DarkGrayOpcHard.OpcHdaServer  HDA1      COpcHdaServer Object
3 localhost InSAT.ModbusOPCServer.HDA     HDA1

```

## 2. Создайте объект клиента OPC HDA

После определения имени хоста (например, 'localhost') и идентификатора OPC сервера к которому необходимо подключиться (например, 'InSAT.ModbusOPCServer.HDA') следует создать объект `opc.hda.Client`.

Используйте `opc.hda.Client` метод конструктора, указав имя хоста и аргументы Server ID.

```
>>hdaClient = opc.hda.Client('localhost', 'InSAT.ModbusOPCServer.HDA')
```

```
hdaClient =
```

```

OPC HDA Client localhost/InSAT.ModbusOPCServer.HDA:
      Host: localhost
      ServerID: InSAT.ModbusOPCServer.HDA
      Timeout: 10 seconds

```

```
      Status: disconnected
```

```

      Aggregates: -- (client is disconnected)
      ItemAttributes: -- (client is disconnected)

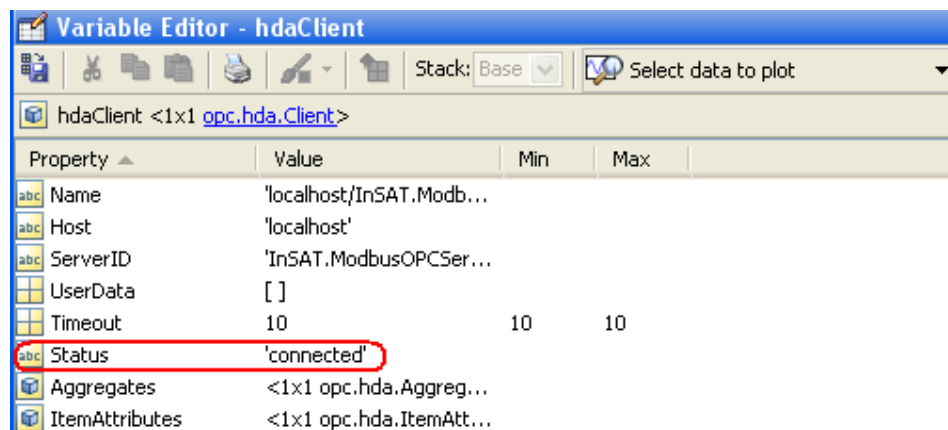
```

```
Methods
```

## 3. Подключите клиент МатЛАБ к OPC HDA серверу

Объект OPC HDA клиента не подключается автоматически к серверу когда создается. Используйте следующую команду подключения `opc.hda.Client` объекта к серверу.

```
>> connect(hdaClient)
```



## 4. Для накопления HDA данных запустите OPC DA сервер.

Перед чтением данных OPC DA сервера с конфигурацией `Simulator.mbc` (см. Рис 1), например, сигналов `Sin` и `Saw`, их надо накопить. Сервер HDA хранит данные только тех переменных, которые были активированы и считаны из промышленной сети или созданы программными средствами сервера OPC DA. Вот почему для работы с HDA сервером, сначала следует запустить DA сервер который создает и автоматически сохраняет данные в HDA сервере. Для этого:

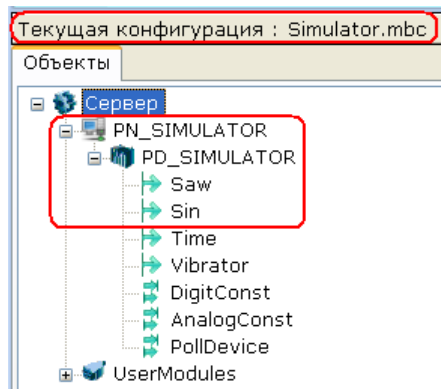


Рис 1. Конфигурация Simulator.mbc OPC сервера.

4.1. Создайте экземпляра объекта управления OPC DA сервера.

```
>> da = opcda('localhost', 'InSAT.ModbusOPCServer.DA');
```

4.2. Подключите клиент объекта к серверу.

```
>> connect(da);
```

4.3. Раскройте список методов, поддерживаемых обработчиком объекта da.

```
>> methods(da);
Methods for class opcda:
Contents          disconnect          removepublicgroup
addgroup         disp              saveobj
cleanup           getnamespace      serveritemprops
clearlogfile     loadobj          serveritems
connect          obj2mfile        showopcevents
copyobj          opcda
delete           opcserverinfo
```

4.4. Используя метод из списка выше создайте группу переменных (тегов)

```
>> grp = addgroup(da);
```

4.5. Раскройте список методов, поддерживаемых обработчиком объекта grp.

```
>> methodsview(grp) или
```

```
>> methods(grp)
```

```
Methods for class dagroup:
Contents      dagroup      getdata      refresh      write
additem     delete      loadobj      saveobj      writeasync
cancelasync  disp        makepublic  start
cleanup      flushdata  peekdata   stop
clonegroup   genslread  read        trend
copyobj      genslwrite readasync  wait
```

4.6. Получите список переменных OPC DA сервера

```
>> serveritems(da)
```

```
ans =
'PN_SIMULATOR'
'PN_SIMULATOR.PD_SIMULATOR'
'PN_SIMULATOR.PD_SIMULATOR.Saw'
'PN_SIMULATOR.PD_SIMULATOR.Sin'
'PN_SIMULATOR.PD_SIMULATOR.Time'
'PN_SIMULATOR.PD_SIMULATOR.Vibrator'
'PN_SIMULATOR.PD_SIMULATOR.DigitConst'
'PN_SIMULATOR.PD_SIMULATOR.AnalogConst'
```

```

'PN_SIMULATOR.PD_SIMULATOR.PollDevice'
'UserModules'
'UserModules.I-7017'
'UserModules.I-7017.AI'
'UserModules.I-7017.AI.CH01'
'UserModules.I-7017.AI.CH02'
'UserModules.I-7017.AI.CH03'
'UserModules.I-7017.AI.CH04'
'UserModules.I-7017.AI.CH05'
'UserModules.I-7017.AI.CH06'
'UserModules.I-7017.AI.CH07'
'UserModules.I-7017.AI.CH08'
'UserModules.I-7017.Error'
'UserModules.I-7017.PollDevice'

```

4.7. Используя методы из списка выше (см. п.4.5) добавьте в группу переменные OPC сервера (см. п.4.6).

```

>> itm1 = additem(grp, 'PN_SIMULATOR.PD_SIMULATOR.Sin');
>> itm2 = additem(grp, 'PN_SIMULATOR.PD_SIMULATOR.Saw');

```

4.8. Определите, что можно сделать с объектом `grp`.

```

>> get(grp)
General Settings:
  DeadbandPercent = 0
  GroupType = private
  Item = []
  Name = G-003.bfb0b
  Parent = [1x1 opcda]
  Tag =
  TimeBias = 0
  Type = dagroup
  UpdateRate = 0.5000
  UserData = []

Callback Function Settings:
  CancelAsyncFcn = @opccallback
  DataChangeFcn = []
  ReadAsyncFcn = @opccallback
  RecordsAcquiredFcn = []
  RecordsAcquiredFcnCount = 20
  StartFcn = []
  StopFcn = []
  WriteAsyncFcn = @opccallback

Subscription and Logging Settings:
  Active = on
  LogFileName = opcdatalog.olf
  Logging = off
  LoggingMode = memory
  LogToDiskMode = index
  RecordsAcquired = 0
  RecordsAvailable = 0
  RecordsToAcquire = 120
  Subscription = on

```

4.9. Задайте время накопления данных `logDuration` в сек., период записи `logRate` в сек. и вычислите количество записей `numRecords`.

```

>> logDuration = 2*60;
>> logRate = 0.2;
>> numRecords = ceil(logDuration./logRate); % 600 instead of 120

```

4.10. Установите новые параметры объекта grp.

```
>> set(grp, 'UpdateRate',logRate,'RecordsToAcquire',numRecords);
```

4.11. Проверьте установку новых параметров.

```
>> get(grp, 'UpdateRate')
ans =
    0.2
>> get(grp, 'RecordsToAcquire')
ans =
    600
```

4.12. Запустите OPC сервер с накоплением данных

```
>> start(grp);
>> wait(grp);
```

4.13. Убедитесь, что после накопления данных OPC сервера в списке параметров объекта grp имеется запись о том, что он готов к новому запуску и у него имеется 600 записей, которые можно получить командой GETDATA или PEEKDATA.

```
>> grp
...
Status          : Waiting for START.
                  600 records available for GETDATA/PEEKDATA
```

5. Считайте в окно переменных МатЛАБ (workspace) HDA данные сервера.

Для чтения HDA данных выполните команду

```
>> HDA_data = getdata(grp)
```

```
HDA_data =
600x1 struct array with fields:
    LocalEventTime
```

Получена структура данных <600x1 struct>. Каждый элемент структуры содержит поля LocalEventTime, например, [2013,10,18,19,43,35.7500] и Item <2x1 struct>.

Структура Item имеет свои поля:

Field	Value
ItemID	'PN_SIMULATOR.PD_SIMULATOR.Sin'
Value	0.4158
Quality	'Good: Non-specific'
TimeStamp	[2013,10,18,19,43,35.6870]

Field	Value
ItemID	'PN_SIMULATOR.PD_SIMULATOR.Saw'
Value	5
Quality	'Good: Non-specific'
TimeStamp	[2013,10,18,19,43,35.6870]

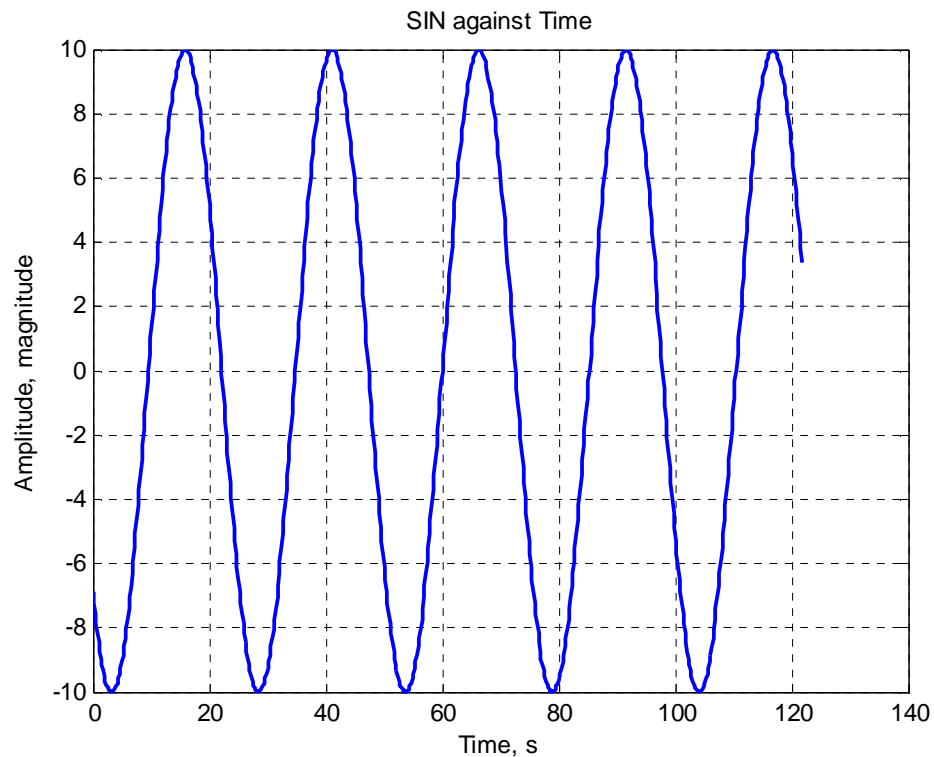
6. Рассмотрите полученную структуру данных. Определите время начала и конца образования структуры данных, среднее значение и нестабильность интервала заполнения структуры данными.

```
>> HDA_data(1).Items(1).TimeStamp
ans =
  1.0e+03 *
  2.0130  0.0100  0.0190  0.0080  0.0280  0.0254
```

```
>> HDA_data(600).Items(1).TimeStamp
ans =
  1.0e+03 *
  2.0130  0.0100  0.0190  0.0080  0.0300  0.0271
```

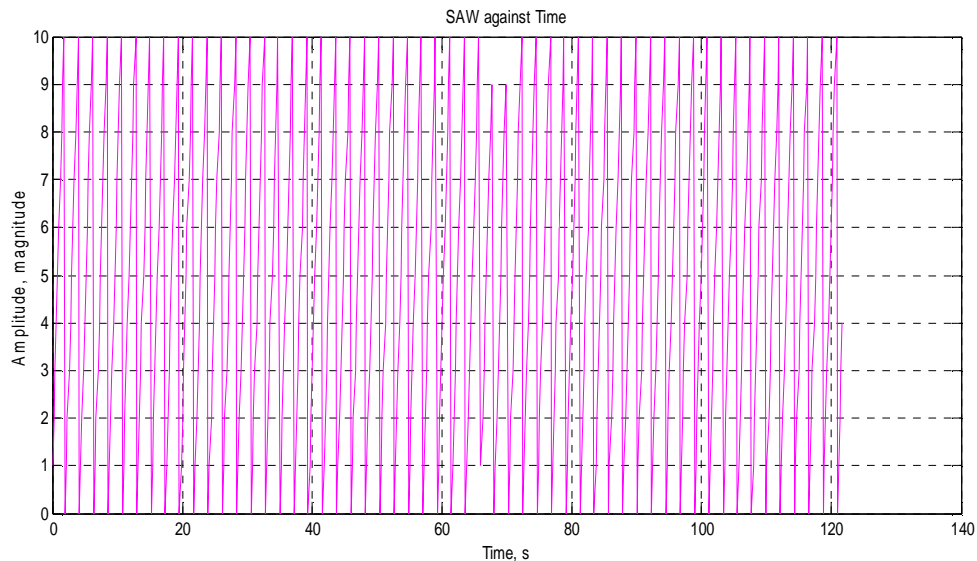
Примечание. Формат переменной **TimeStamp**: [год, месяц, день, час, минуты, секунды]

7. Постройте графики HDA сигналов. Определите амплитуды и периоды сигналов. Найдите причину нестабильности амплитуды пилообразного сигнала показанного на рисунке выше.



```
for i = 1:600
    sin_time(i) = datenum(HDA_data(i).Items(1).TimeStamp);
    sin_val(i) = HDA_data(i).Items(1).Value;
end
sin_time = 24*3600*(sin_time - sin_time(1));

figure
plot(sin_time, sin_val, 'b', 'linewidth',2');
grid on
xlabel('Time, s');
ylabel('Amplitude, magnitude');
title('SIN against Time');
```



```

for i = 1:600
    saw_time(i) = datenum(HDA_data(i).Items(2).TimeStamp);
    saw_val(i) = HDA_data(i).Items(2).Value;
end
saw_time = 24*3600*(saw_time - saw_time(1));

figure
plot(saw_time, saw_val, 'm', 'linewidth',1');
grid on
xlabel('Time, s');
ylabel('Amplitude, magnitude');
title('SAW against Time');

```

Примечание. Функция **datenum** преобразует числовой массив (год, месяц, день, час, минуты, секунды) в эквивалентное количество дней

- Найдите причину нестабильности амплитуды пилообразного сигнала (SAW сигнал) показанного на рисунке выше.

### Задание 3. Получение справочной информации о формате метода.

- Снова запустите OPC сервер с накоплением данных.

```
>> start(grp);
```

```
>> wait(grp);
```

- Убедитесь, что накопление выполнено и данные доступны.

```
>> grp
```

```

...
Status      : Waiting for START.
              600 records available for GETDATA/PEEKDATA

```

- Попытайтесь считать данные методом **peekdata** объекта **grp**:

```
>> HDA_data_2 = peekdata(grp)
```

**Not enough input arguments.** (Сообщение МатЛАБ: недостаточно аргументов)



12. Повторите ввод:

```
>> HDA_data_2 = peekdata(grp,")
```

Сообщение МатЛАБ:

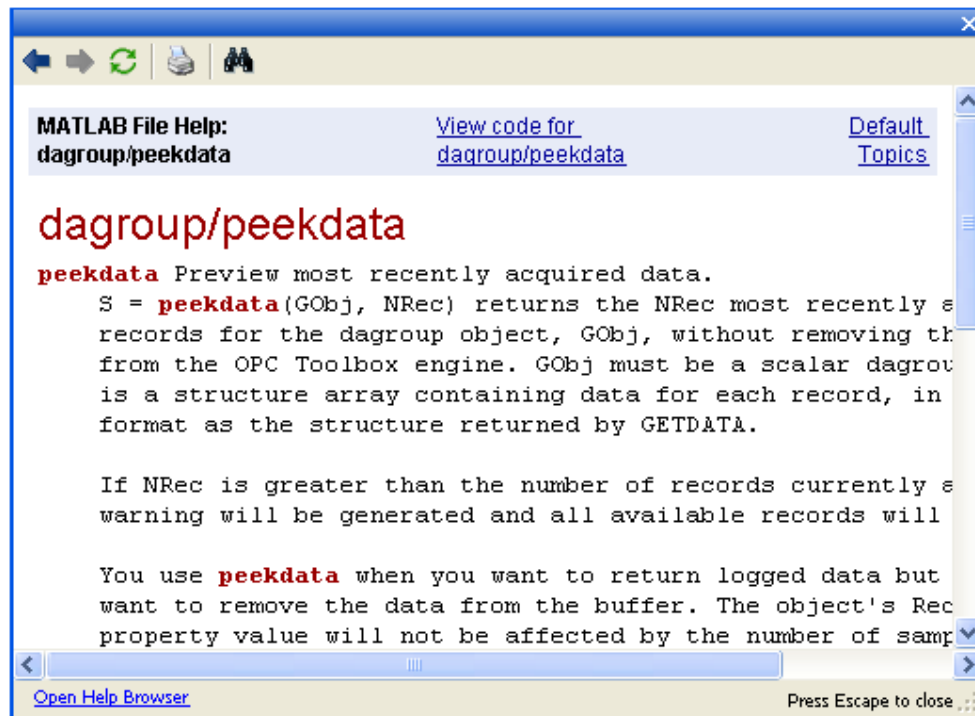
Operands to the || and && operators must be convertible to logical scalar values.

Error in [dagroup/peekdata](#) (line 58)

if nRec<1 || (rem(nRec, 1)>0),

13. Щелкните по выделенной гиперссылке [dagroup/peekdata](#) сообщения МатЛАБ.

```
>> HDA_data_2 = peekdata(grp,")
```



В появившемся окне содержится справочная информация о формате и коде команды **peekdata** из которой следует, что вторым аргументом команды должно быть количество считываемых записей.

14. Повторите ввод в правильном формате, например,

```
>> HDA_data_2 = peekdata(grp, 100);
```

Убедитесь, что в окне **workspace** появилась структурная переменная **HDA\_data\_2** из 100 записей.

15. Удалите объекты OPC DA и HDA серверов после их использования. Обратите внимание, что при удалении объекта автоматически удаляются и наследственные объекты (children) из панели toolbox engine.

<pre>&gt;&gt; disconnect(hdaClient)       Status: disconnected  &gt;&gt; delete(hdaClient)  &gt;&gt; clear hdaClient data</pre>	<pre>&gt;&gt; disconnect(da)       Status: disconnected  &gt;&gt; delete(da)  &gt;&gt; clear hdaClient data</pre>
---	---

#### Задание 4. Отображение текущих переменных DA сервера на графопостроителе МатЛАБ.

1. Создайте следующий текстовый документ в редакторе МатЛАБ.

```
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
% OPC_DA_read_SIN.m   v1.0A
% Matlab v7.0 (R14) SP 1
% Bob Davidov
%
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
clear all

hostInfo = opcserverinfo('localhost');
for i = 1:length(hostInfo.ServerID)
    hostInfo.ServerID(i)
end
hostInfo = opc.hda.getServerInfo('localhost')
hdaClient = opc.hda.Client('localhost', 'InSAT.ModbusOPCServer.HDA')
connect(hdaClient)

da = opcda('localhost', 'InSAT.ModbusOPCServer.DA');
connect(da);
grp = addgroup(da);
itm1 = additem(grp, 'PN_SIMULATOR.PD_SIMULATOR.Sin');
itm2 = additem(grp, 'PN_SIMULATOR.PD_SIMULATOR.Saw');
logDuration = 2*60;
logRate = 0.2;
numRecords = ceil(logDuration./logRate); % 600 instead of 120
set(grp, 'UpdateRate',logRate,'RecordsToAcquire',numRecords);

figure
N = 45;
set(gca, 'XLim',[0 N], 'YLim',[-10 10])
grid
xlabel('Time, s');
ylabel('PN_SIMULATOR.PD_SIMULATOR.Sin, magnitude');
title({'InSAT.ModbusOPCServer.DA ... Sin against Time'})
for l = 0:N
    r = read(itm1);
    hold on;
    plot(l, r.Value, 'xb');
    pause (0.5);
end

disconnect(da)
delete(da)
clear da data

%end of m-File.m
```

2. Поставьте точки останова и запустите файл командой <F5>
3. Рассмотрите результаты выполнения команд представленной программы, определите их назначение.

4. Внесите изменения в программу для отображения переменной OPC DA сервера:  
`PN_SIMULATOR.PD_SIMULATOR.Saw'`
5. Проверьте отображение переменной.

**Задание 5.** Отображение накопленных значений HDA сервера.

1. Создайте следующий текстовый документ в редакторе m-файлов МатЛАБ.

```

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
% OPC_read_SIN_HDA.m  v1.0A
% Matlab v7.0 (R14) SP 1
% Bob Davidov
%
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
clear all

hostInfo = opcserverinfo('localhost');
for i = 1:length(hostInfo.ServerID)
    hostInfo.ServerID(i)
end
hostInfo = opc.hda.getServerInfo('localhost')
hdaClient = opc.hda.Client('localhost', 'InSAT.ModbusOPCServer.HDA')
connect(hdaClient) % MasterSCADA is loaded

da = opcda('localhost', 'InSAT.ModbusOPCServer.DA');
connect(da);
grp = addgroup(da);
itm1 = additem(grp, 'PN_SIMULATOR.PD_SIMULATOR.Sin');
itm2 = additem(grp, 'PN_SIMULATOR.PD_SIMULATOR.Saw');
numRecords = 50;
set(grp, 'RecordsToAcquire', numRecords);
start(grp)
wait(grp)

data = hdaClient.readRaw({'PN_SIMULATOR.PD_SIMULATOR.Sin',
'PN_SIMULATOR.PD_SIMULATOR.Saw'}, now-100, now)

figure
plot(24*3600*(data(1).TimeStamp-data(1).TimeStamp(1)),data(1).Value,'b','linewidth',2)
grid on
xlabel('Time, s');
ylabel('Amplitude, magnitude');
title('HDA SIN against Time');

figure
plot(24*3600*(data(2).TimeStamp-data(2).TimeStamp(1)),data(2).Value,'m','linewidth',1)
grid on
xlabel('Time, s');
ylabel('Amplitude, magnitude');
title('HDA SAW against Time');

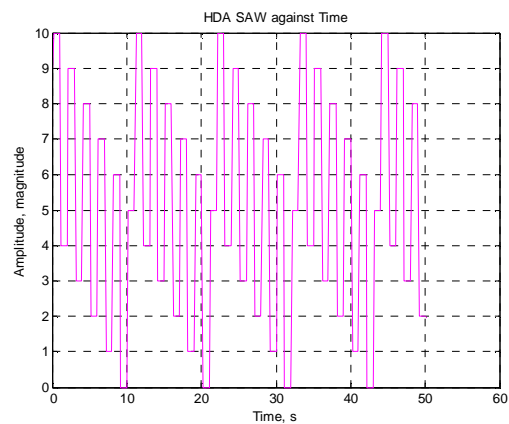
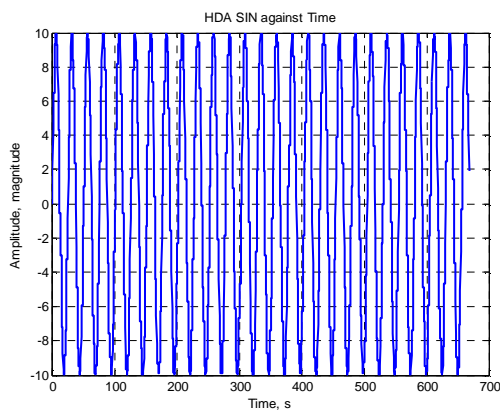
```

```
disconnect(da)
delete(da)
clear da data
```

```
disconnect(hdaClient)
delete(hdaClient)
clear hdaClient data
```

```
%end of m-File.m
```

2. В режиме отладки по шагам выполните программу. Определите назначение команд программы.
3. Сверьте полученный результат со следующими графическими данными.



## КОНТРОЛЬНЫЕ ВОПРОСЫ

1. Для чего предназначен OPC HDA сервер?
2. Как установить режим накопления данных HDA сервером?
3. Как изменить скорость накопления данных?
4. Почему отличаются размеры массивов переменных SIN и SAW задания 5?
5. Почему на последнем рисунке справа график не похож на пилообразный сигнал SAW OPC сервера ?

## БИБЛИОГРАФИЧЕСКИЙ СПИСОК

1. Раздел Help МатЛАБ.
2. OPC Toolbox, User's guide. R2013b [http://www.mathworks.com/help/pdf\\_doc/opc/opc.pdf](http://www.mathworks.com/help/pdf_doc/opc/opc.pdf)