How to Control Tabor AWGs with MATLAB

Example3: Synchronize between two Tabor AWGs

This example will demonstrate how work with two synchronized WX2184C AWGs. It will show how to create & download a sequence built from 3 different waveforms files into the AWGs' waveform memory. While keeping the two AWGs still synchronized, download and output a new sequence table every 0.6 seconds.

If you haven't been acquainted with controlling Tabor AWGs using MATLAB's Instrument Control Toolbox, please refer to Tabor's website for the previous tutorials in the series ""How to Control Tabor AWGs with MATLAB":

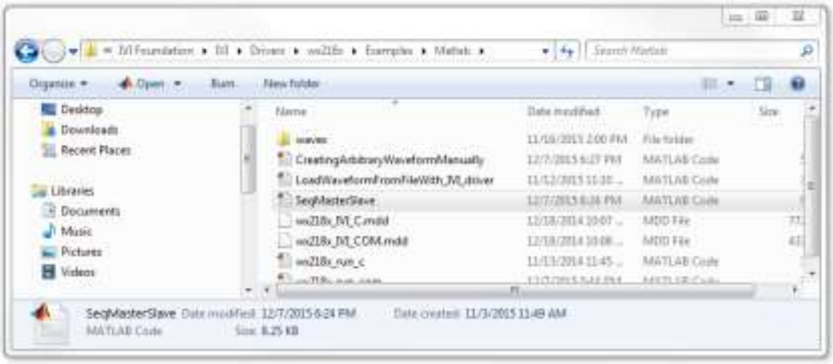
- 'Getting started'.
- 'Using SCPI commands'.
- 'Using the IVI driver'.

In order to control instruments using MATLAB, the instrument control toolbox is required. Please note that the Instrument Control Toolbox is an additional application that needs to be added. For more information you can visit the **Mathworks** website.

For this tutorial, we will use MATLAB version R2014a 32bit and a USB interface. To ensure you successfully established all the necessary settings for remote control over the Tabor instrument using LAN/USB/GPIB, please go over the **connectivity tutorials** on the Tabor's website.

The 'SeqMasterSlave.m' itself as well as the waveforms (in the 'waves' folder), can be found after installation of the latest Tabor IVI driver through this path on your computer:

'C:\Program Files (x86)\IVI Foundation\IVI\Drivers\wx218x\Examples\Matlab'.



Name	Date modified	Type	Size
waves	11/10/2015 2:00 PM	File folder	
CreatingArbitraryWaveformManually	12/7/2015 6:27 PM	MATLAB Code	
LoadWaveformFromFileWith_IVI_driver	11/12/2015 11:10 ...	MATLAB Code	
SeqMasterSlave	12/7/2015 6:24 PM	MATLAB Code	
wx218x_IVI_C.mdd	12/18/2014 10:07 ...	MDD File	77
wx218x_IVI_COM.mdd	12/18/2014 10:08 ...	MDD File	41
wx218x_run_c	11/13/2014 11:45 ...	MATLAB Code	

SeqMasterSlave Date modified: 12/7/2015 6:24 PM Date created: 11/3/2015 11:49 AM
MATLAB Code Size: 8.25 KB

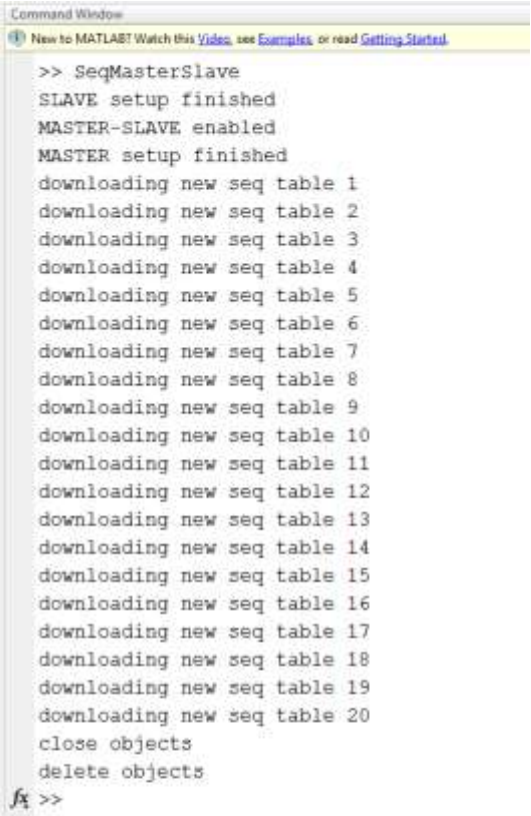
For information regarding the steps, please read the detailed comments in the code itself.

```
% SeqMasterSlave.m----------------------------------------------------
% This Example Shows How To:
% 1. Create a sequence
% 2. Download it to 2 instruments
% 3. Setting them as master and slave
% 4. Outputting the sequence from the synchronized instruments
% 5. Downloading a new sequence table to each instrument
% 6. outputting the new sequence from the synchronized instruments
%
% Setup Instructions:
% 1. AWG_MASTER:CH1->SCOPE:CH1
% 2. AWG_SLAVE:CH1->SCOPE:CH2
% 3. AWG_MASTER:SYNC_OUT->SCOPE:CH3
% 4. SCOPE: set trigger source and trigger level to CH3
% 5. Make sure to clear the workspace before each run.
%---------------------------------------------------------------------
MasterAddress = 'USB0::0x168C::0x2184::0000215470::0';
SlaveAddress  = 'USB0::0x168C::0x2184::0000215993::0';
%% ----------------------------------------SLAVE setup--------------------------------------------
ACTIVE_CHANNEL = 'Channel1';
% Open connection to instrument with IP/USB address
%dev = icdevice('wx218x_IVI_COM.mdd',
%'TCPIP0::192.168.0.185::5025::SOCKET','optionstring','simulate=false,DriverSetup=Trace=True,TraceArray=true
%');
devSLAVE = icdevice('wx218x_IVI_COM.mdd', SlaveAddress,'optionstring','simulate=false');
try
  % Connecting to device object
  connect(devSLAVE);
  % Getting the instrument model
  groupId = get(devSLAVE, 'Identity');
  InstrModel = groupId.InstrumentModel;
  % Reset device
  groupCnf = get(devSLAVE, 'Utility');   % The reset operation take a few seconds so  if not needed,
%running without a reset will significantly improve program running time.
  invoke(groupCnf, 'Reset')
  % Set the sample clock
  groupArb = get(devSLAVE, 'Arbitrary');
  groupArb.SampleRate = 80E6;
  % Create three segments in the channel A and load waves in them
  groupArb = get(devSLAVE, 'ArbitraryWaveform');
  % Doing this only for WX2184
  if (((strcmp(InstrModel, 'WX2184')) == 1) || ((strcmp(InstrModel, 'WX1284')) == 1) || ((strcmp(InstrModel,
'WX2184C')) == 1) || ((strcmp(InstrModel, 'WX1284C')) == 1))
      groupArb.TraceMode = 'WX218xTraceModeDuplicate';
  end
  % Downloading waveform files
  wavesdir = [pwd '\waves\'];
  wfmHandle1 = invoke(groupArb, 'LoadArbWfmFromFile', ACTIVE_CHANNEL, [wavesdir 'pulse_2048.wav']);
  wfmHandle2 = invoke(groupArb, 'LoadArbWfmFromFile', ACTIVE_CHANNEL, [wavesdir 'sinc_8192.wav']);
  wfmHandle3 = invoke(groupArb, 'LoadArbWfmFromFile', ACTIVE_CHANNEL, [wavesdir 'square_1024.wav']);
  wfmHandles = [wfmHandle1; wfmHandle2; wfmHandle3];
  loop = [1; 1; 1 ];
  flag = [0; 0; 0 ];
  % downloading sequence as follows:
  %  Step #              Segment #               Repeats Count
  %  [ 1 ]    [ Segment 1 ( pulse_2048.wav  )]         [ 1 ]
  %  [ 2 ]    [ Segment 2 ( sinc_8192.wav   )]         [ 1 ]
  %  [ 3 ]    [ Segment 3 ( square_1024.wav )]         [ 1 ]
  groupSeq = get(devSLAVE, 'Arbitrarysequence');
  seqHandle = invoke(groupSeq, 'CreateSequenceAdv', int32(wfmHandles), int32(loop), uint8(flag));
  % Create the sequence in the active channel
  groupOutput = get(devSLAVE, 'Output');
  % Set the Output Mode to Sequence
  groupOutput.OutputMode = 'WX218xOutputModeSequence';
  % Output enabled
  invoke(groupOutput, 'Enabled', ACTIVE_CHANNEL, 1)
  % Output SYNC signal enable
  invoke(get(devSLAVE, 'Outputsync'), 'Enabled', ACTIVE_CHANNEL, 1)
  % Setting as SLAVE.
  %invoke(devSLAVE, 'SendCmd', ':XINS:MODE SLAVE');
  groupObj = get(devSLAVE, 'Xinstrument');
  groupObj = groupObj(1);
  invoke(groupObj, 'ConfigureXInstrumentModeAdv', 'WX218xXInstrumentModeAdvSlave');
catch aException
  disp(aException.message);
end
```

```
%% ------------------MASTER setup-----------------------------------------------
ACTIVE_CHANNEL = 'Channel1';
% Open connection to instrument with IP/USB address
%dev = icdevice('wx218x_IVI_COM.mdd',
%'TCPIP0::192.168.0.185::5025::SOCKET','optionstring','simulate=false,DriverSetup=Trace=True,TraceArr
%ay=true%');
devMASTER = icdevice('wx218x_IVI_COM.mdd', MasterAddress,'optionstring','simulate=false');
try
  connect(devMASTER);
  groupId = get(devMASTER, 'Identity');
  InstrModel = groupId.InstrumentModel;
  % Reset device
  groupCnf = get(devMASTER, 'Utility');
  invoke(groupCnf, 'Reset')  % The reset operation take a few seconds so if not needed, running
%without a reset will significantly improve program running time.
  % Set the sample clock
  groupArb = get(devMASTER, 'Arbitrary');
  groupArb.SampleRate = 2000E6;
  % Create three segments in the channel A and load waves in them
  groupArb = get(devMASTER, 'ArbitraryWaveform');
   %Doing this only for WX2184
  if (((strcmp(InstrModel, 'WX2184')) == 1) || ((strcmp(InstrModel, 'WX1284')) == 1) ||
((strcmp(InstrModel, 'WX2184C')) == 1) || ((strcmp(InstrModel, 'WX1284C')) == 1))
      groupArb.TraceMode = 'WX218xTraceModeDuplicate';
  end
  % Downloading waveform files
  wavesdir = [pwd '\waves\'];
  wfmHandle1 = invoke(groupArb, 'LoadArbWfmFromFile', ACTIVE_CHANNEL, [wavesdir 'pulse_2048.wav']);
  wfmHandle2 = invoke(groupArb, 'LoadArbWfmFromFile', ACTIVE_CHANNEL, [wavesdir 'sinc_8192.wav']);
  wfmHandle3 = invoke(groupArb, 'LoadArbWfmFromFile', ACTIVE_CHANNEL, [wavesdir 'square_1024.wav']);
  wfmHandles = [wfmHandle1; wfmHandle2; wfmHandle3];
  loop = [1; 1; 1 ];
  flag = [0; 0; 0 ];
  % Sequence Description:
  %  Step #              Segment #               Repeats Count
  %  [ 1 ]    [ Segment 1 ( pulse_2048.wav ) ]         [ 1 ]
  %  [ 2 ]    [ Segment 2 ( sinc_8192.wav )  ]         [ 1 ]
  %  [ 3 ]    [ Segment 3 ( square_1024.wav )]         [ 1 ]
  %
  groupSeq = get(devMASTER, 'Arbitrarysequence');
  seqHandle = invoke(groupSeq, 'CreateSequenceAdv', int32(wfmHandles), int32(loop), uint8(flag));
  % Create the sequence in the active channel.
  groupOutput = get(devMASTER, 'Output');
  % set the Output Mode to Sequence
  groupOutput.OutputMode = 'WX218xOutputModeSequence';
  % Output enable
  invoke(groupOutput, 'Enabled', ACTIVE_CHANNEL, 1)
  % Output SYNC signal enable
  invoke(get(devMASTER, 'Outputsync'), 'Enabled', ACTIVE_CHANNEL, 1)
  % setting as MASTER.
  %invoke(devMASTER, 'SendCmd', ':XINS:MODE MASTER');
  % Execute device object function(s).
  groupObj = get(devMASTER, 'Xinstrument');
  groupObj = groupObj(1);
  invoke(groupObj, 'ConfigureXInstrumentModeAdv', 'WX218xXInstrumentModeAdvMaster');
  % MASTER-SLAVE enabled
  %invoke(devMASTER, 'SendCmd', ':XINS:STAT ON');
  invoke(groupObj, 'ConfigureXInstrumentEnabled', 1);
  %------------------delay-------------------------------------
  % Instrument takes about 2 seconds to sync. We are working to make this
  % time shorter.
  pause(2);
  %------------------------------------------------------------
  disp('MASTER-SLAVE enabled');
catch aException
  disp(aException.message);
end
disp('MASTER setup finished');
```

```
%% --------------------------change sequence table--------------------------
for i=1:20
try
loop = [i; 2*i+1; 3*i+2 ];
flag = [0; 0; 0 ];
fprintf('downloading new seq table %d\n', i);
  groupSeq = get(devMASTER, 'Arbitrarysequence');
  seqHandle = invoke(groupSeq, 'CreateSequenceAdv', int32(wfmHandles), int32(loop), uint8(flag));
%--------------delay---------------------------------------------------
% delay will change according to the size of download.
pause(0.3);
%----------------------------------------------------------------------
  groupSeq = get(devSLAVE, 'Arbitrarysequence');
  seqHandle = invoke(groupSeq, 'CreateSequenceAdv', int32(wfmHandles), int32(loop), uint8(flag));
%-------------------delay----------------------------------------------
% delay will change according to the size of download
pause(0.3);
%-----------------------------------------------------------------------
catch aException
  disp(aException.message);
end
end
%% close objects ---------------------------------
disconnect(devSLAVE);
disconnect(devMASTER);
disp('close objects');
%% delete objects ----------------------------------------
%answer=input('delete objects?\n1-> yes\n0-> no\n');
%if (answer)
delete(devMASTER);
delete(devSLAVE);
clear all;
disp('delete objects');
%end
```

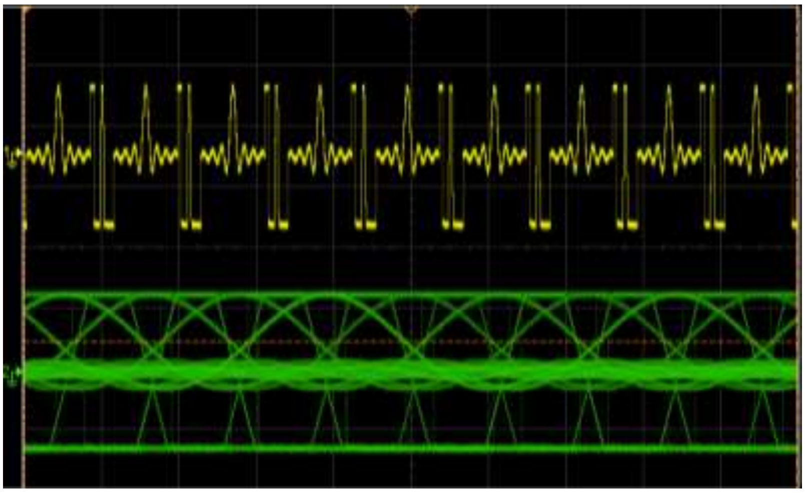
After the program finishes, your command window should look as follows:



New to MATLAB? Watch this Video, see Examples, or read Getting Started.

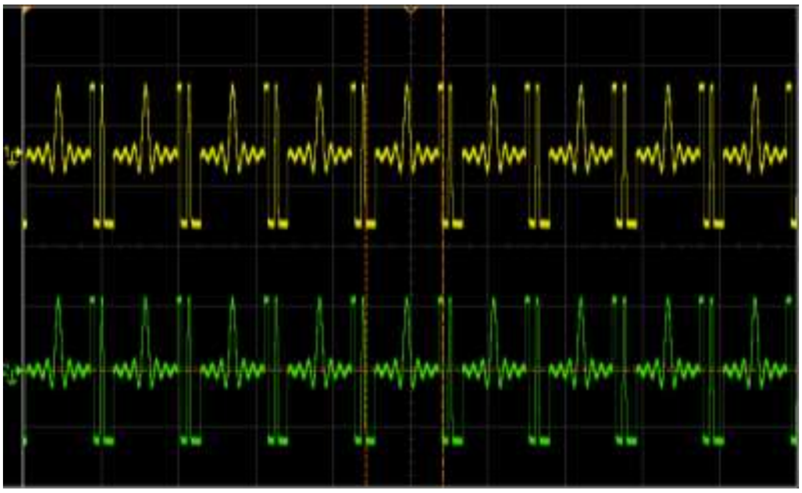
```
>> SeqMasterSlave
SLAVE setup finished
MASTER-SLAVE enabled
MASTER setup finished
downloading new seq table 1
downloading new seq table 2
downloading new seq table 3
downloading new seq table 4
downloading new seq table 5
downloading new seq table 6
downloading new seq table 7
downloading new seq table 8
downloading new seq table 9
downloading new seq table 10
downloading new seq table 11
downloading new seq table 12
downloading new seq table 13
downloading new seq table 14
downloading new seq table 15
downloading new seq table 16
downloading new seq table 17
downloading new seq table 18
downloading new seq table 19
downloading new seq table 20
close objects
delete objects
fx >>
```

At first, the two WX2184C aren't synchronized and one WX outputs generates a sequence with SCLK=2Gs/s and the second WX runs the same sequence with SCLK=80Ms/s:



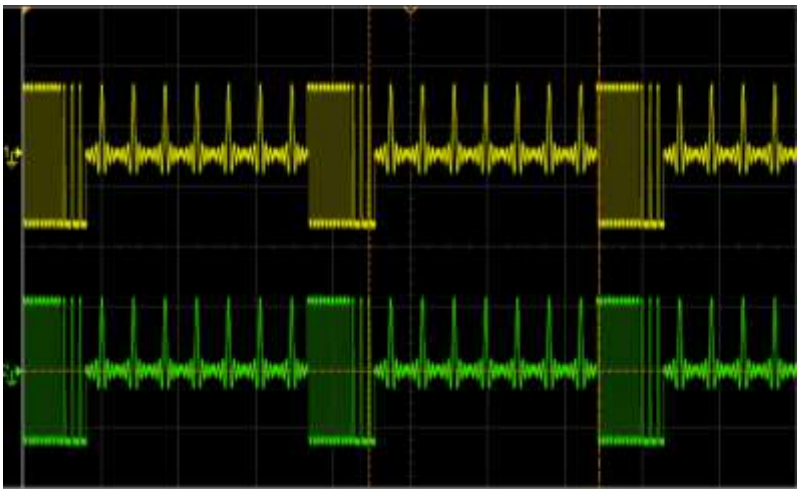
The two WX2184C before synchronization took place.

Once you run the enable sync command, the MASTER & SLAVE's outputs are now in sync:



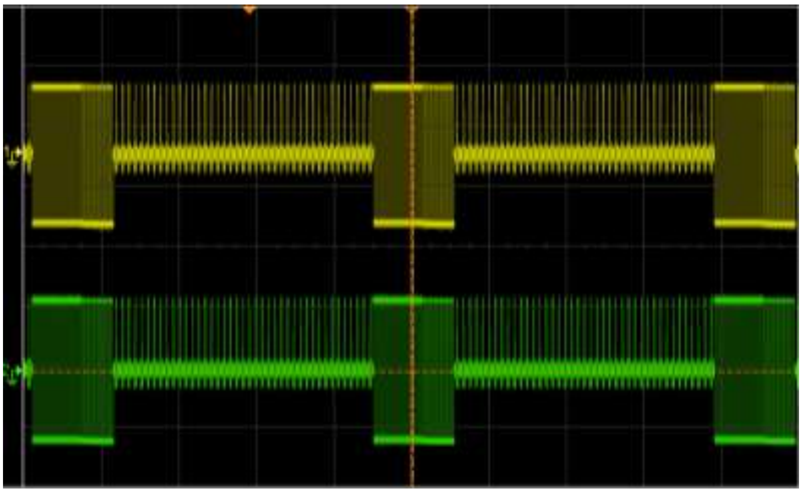
The two WX2184C after synchronization completed.

And a new sequence table is downloaded & outputted every 0.6 seconds:



The two WX2184C after downloading a new sequence table.

Here is the 20th iteration as the last sequence table is outputted:



The two WX2184C after downloading a new sequence table in the 20th iteration.

In the last tutorial, we will show one more example:

- Example #4: Create an arbitrary waveform from binary data using SCPI commands.

For More Information

To learn more about how to use MATLAB with Tabor instruments, visit our website Support & Tutorials zone. For more of Tabor's solutions or to schedule a demo, please contact your local Tabor representative or email your request to info@tabor.co.il. More information can be found at our website at www.taborelec.com