

## How to Control Tabor AWGs with MATLAB

### Example4: Creating arbitrary waveform from binary data

This example will demonstrate how to create an arbitrary waveform from binary data created using MATLAB code. Use this code with its detailed comments to understand how binary data should be sent to the Tabor AWG waveform memory. It is a good programming example for all other languages, because the same procedure that applies here, will apply to other programming languages as well.

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 example 'CreateArbitraryWaveformManually.m' can also 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'.*



Here is the full code used to create a 500KHz sine wave using Tabor WX2184C:

```

close all;
clear all;
clc;
% determind the Size of output buffer
blocksize = 10000;
% -----Create 500KHz sine wave for testing-----
% Number of waveform samples
numSamples = 4000;
% Ensure the numnber of samples > 192 samples and is a multiple of 16
numSamples = max(192, ceil(numSamples/16)*16);
% Setting the output frequency
fs = 2e9;% If you wish to change the SCLK, this parameter needs to be changed also in the SCPI
%command below
f = fs/numSamples;% Output frequency is a result of: f = SCLK / # of samples.
dt = 1/fs;
t = 0:dt:(numSamples/fs)-dt;
% Data array is in [0,1] values. When creating a waveform as binary data,
% one must normalize the data points in the array to be between -1 and 1
data = 0.5*(sin(2*pi*f*t)+1);
% Convert data to 14 bit, as described in manual section 4-66
data = uint16(((2^13-1)*data)+2^13); % 14 bit data
% Convert data to byte array [low byte 0, high byte 0, low byte 1, ...]
bytes = zeros(2*numSamples,1, 'uint8');
% Low byte
bytes(1:2:2*length(data)) = uint8(bitand(data, uint16(hex2dec('00FF'))));
% High byte
bytes(2:2:2*length(data)) = uint8(bitshift(bitand(data, uint16(hex2dec('FF00'))), -8));
% -----creating VISA-USB object-----
obj1 = instrfind('Type', 'visa-usb', 'RsrcName', 'USB0::0x168C::0x2184::0000215470::0::INSTR', 'Tag',
'');
% Create the VISA-USB object if it does not exist
% otherwise use the object that was found.
if isempty(obj1)
obj1 = visa('NI', 'USB0::0x168C::0x2184::0000215470::0::INSTR');
else
fclose(obj1);
obj1 = obj1(1)
end

% -----Communicating with instrument object-----
% Set output buffer size to blocksize

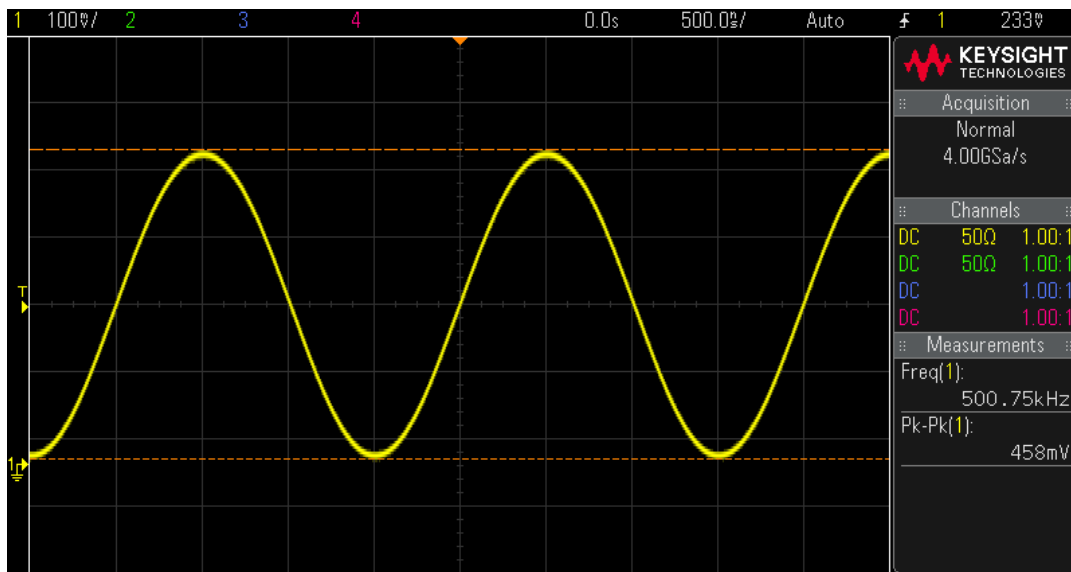
```

```

set(obj1, 'OutputBufferSize', blocksize);
% Connect to instrument object, obj1.
fopen(obj1);
% Asking for device identification details
idn = query(obj1, '*IDN?');
    if isempty(strfind(idn, 'WX2184')),
        error('Instrument not supported.\n\nplease go to Tabor front panel->Utility->Remote interface
and select your remote interface.\n');
    else
        fprintf('Instrument IDN: %s', idn);
    end
% Clearing all the old errors from error's buffer.
% This is done to ensure error's aren't from previous sessions.
fprintf(obj1, '*CLS');
% Resetting the device.
fprintf(obj1, '*RST');
% Couple channel 1&2 with 3&4
fprintf(obj1, ':INST:COUP:STAT ON');
% Select function type
fprintf(obj1, ':FUNC:MODE USER');
% Set continuous mode on
fprintf(obj1, ':INIT:CONT 1');
% Select channel 1 as the active channel
fprintf(obj1, ':INST:SEL CH1');
% Set arbitrary waveform SCLK to 2Gs/s
fprintf(obj1, ':FREQ:RAST 2e9');
% Select internal source for the SCLK
fprintf(obj1, ':FREQ:RAST:SOUR INT');
% Selects DC couple as output.
fprintf(obj1, ':OUTP:COUP DC');
% Setting the amplitude and offset
fprintf(obj1, ':VOLT:LEV:AMPL 1');
fprintf(obj1, ':VOLT:LEV:OFFS 0');
% Open channel 1(CH1 is the active channel)
fprintf(obj1, ':OUTP ON');
% Delete all segments from memory
fprintf(obj1, ':TRAC:DEL:ALL');
% Select the source as BUS
fprintf(obj1, ':TRAC:SEL:SOUR BUS');
% Define trace
fprintf(obj1, [':TRAC:DEF 1,' num2str(numSamples)]);
cLength = strcat(num2str(length(num2str(2*numSamples))), num2str(2*numSamples));
samples = sprintf('%s', cLength);
% Select trace
fprintf(obj1, ':TRAC:SEL 1');
% Write data blockwise
fprintf(obj1, '%s', [':TRAC:DATA' samples]);
n = numel(bytes);
num_blocks = ceil(n/blocksize);
for k=1:num_blocks
    pause(0.2)
    fprintf('downloading chunk %d\n', k);
    fwrite(obj1, bytes((k-1)*blocksize+1:min(n,k*blocksize)), 'uint8');
end
% Checking for Errors
query(obj1, ':SYST:ERR?')
% Wait for completion
query(obj1, '*OPC?');
% Enable
fprintf(obj1, ':ENAB');
% -----Disconnect from instrument object-----
if ~isempty(obj1),
    fclose(obj1);
    delete(obj1);
end
% ----- END OF EXAMPLE -----

```

As can be seen on scope, a 500KHz sine wave was created:



The outputted sine wave.

## 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](mailto:info@tabor.co.il). More information can be found at our website at [www.taborelec.com](http://www.taborelec.com)

© Proprietary of Tabor Electronics Ltd.