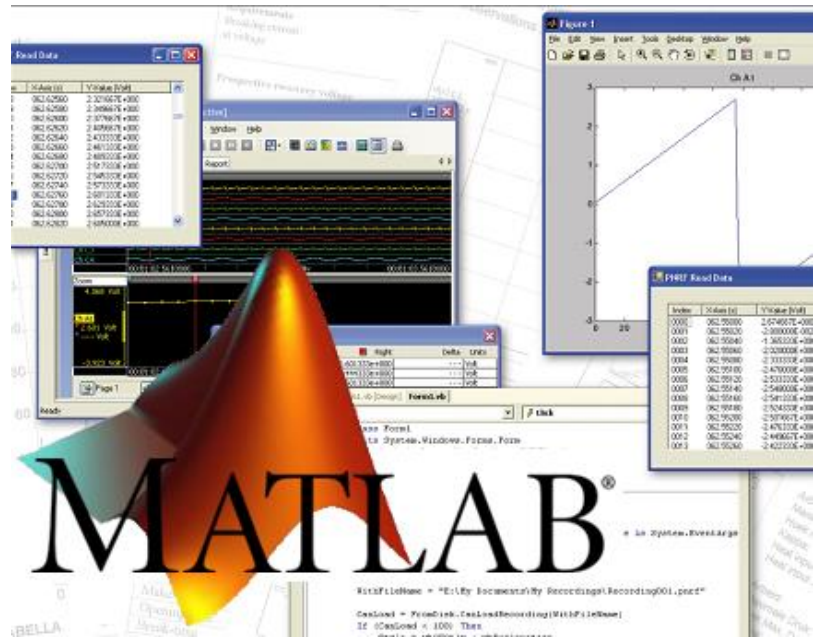


# User Manual

English



## Perception PNRF Reader MATLAB® Examples

Document version 2.0 – November 2020

For HBM's Terms and Conditions visit [www.hbm.com/terms](http://www.hbm.com/terms)

Hottinger Brüel & Kjaer GmbH  
Im Tiefen See 45  
64293 Darmstadt  
Germany  
Tel: +49 6151 80 30  
Fax: +49 6151 8039100  
Email: [info@hbm.com](mailto:info@hbm.com)  
[www.hbm.com/highspeed](http://www.hbm.com/highspeed)

Copyright © 2020

All rights reserved. No part of the contents of this book may be reproduced or transmitted in any form or by any means without the written permission of the publisher.

**LICENSE AGREEMENT AND WARRANTY**

For more information about LICENSE AGREEMENT AND WARRANTY refer to:

[www.hbm.com/terms](http://www.hbm.com/terms)

# Table of Contents

<b>TABLE OF CONTENTS</b> .....	<b>4</b>
<b>1 GETTING STARTED</b> .....	<b>5</b>
1.1 INTRODUCTION .....	5
1.2 INTENDED AUDIENCE .....	5
1.3 REQUIREMENTS AND INSTALLATION .....	5
1.3.1 <i>Installation</i> .....	5
<b>2 EXAMPLE 1</b> .....	<b>5</b>
<b>3 EXAMPLE 2</b> .....	<b>8</b>
<b>4 EXAMPLE 3</b> .....	<b>9</b>

## 1 Getting Started

Welcome to the Perception PNRF Reader MATLAB®.examples document. This document contains a couple of examples showing how you can use the PNRF reader interface of Perception from within MATLAB. For more information on the RPC interface we refer to the user manual called “**PNRF Reader SDK**” (I2697\_1.1en)

### 1.1 Introduction

MATLAB® is a well-known language for technical computing. It integrates computation, visualization, and programming in an environment where problems and solutions are expressed in familiar mathematical notation.

MATLAB provides interfaces to clients or servers communicating via Component Object Model (COM). In this section we will describe how you can interface MATLAB to the PNRF Reader using COM.

### 1.2 Intended audience

This documentation assumes you have sufficient knowledge of MATLAB®, this manual is NOT a tutorial on how to use MATLAB®.

This documentation also assumes you understand the HBM acquisition terminology. Understanding acquisition terminology is vital to understanding digital recordings: trigger, sample rate, pre-/post trigger, etc.

### 1.3 Requirements and installation

We assume you have installed MATLAB and the Perception PNRF Reader SDK. In addition you must install the required software modules as described below.

#### 1.3.1 Installation

The PNRF SDK contains everything you need to install and use the PNRF API. When you received the SDK on a CD you must install the SDK from the CD onto your hard disk; you cannot use the API from the CD.

## 2 Example 1

Opening a pnrf file and showing the data of the first channel of the first recorder. Next to the data the so called “*Data values*” are shown. These “*Data values*” can contain recording specific information like the name of the device under test, test conditions like position or size or brand of device under test. It is up to the operator what information has been stored. The entered information from the Perception information sheet is stored into the “*Data values*”.

```

%-- 22-1-14 12:00 --%
% Create PNRF reader OLE Automation server
FromDisk = actxserver('Perception.Loaders.pNRF');
% FromDisk = actxserver('Perception.Loaders.NRF');
% FromDisk = actxserver('Perception.Loaders.LRF');

% Use the open file dialog to select a recording file
[FileName,PathName] = uigetfile('*.pNRF',...
    'Select a Perception Recording File');
MyRecordingName = [PathName, FileName];
  
```

```

% But you also can do this via code:
% MyRecordingName = 'D:\PNRFReader\Waveforms\MyRecordingFile.pnrf';

% Load your recording, myData is a Recording Interface
MyData = FromDisk.LoadRecording(MyRecordingName);

% Use the recording interface to get an interface to the 1st recorder
MyFirstRecorder = MyData.Recorders.Item(1);

% From the first recorder we take the channel collection
myChannels = MyFirstRecorder.Channels;

% From this channel collection we take the first channel
myFirstChannel = myChannels.Item(1);

% Now we get the DataSource (data source) interface from the first channel
% The parameter 3 means: DataSourceSelect_Mixed
ItfData = myFirstChannel.DataSource(3);

% Get the sweeps collection from the datasource interface
MySweeps = ItfData.Sweeps;

% Get the start and end time of the recorded data of the first channel
dStartTime = MySweeps.StartTime;
dEndTime = MySweeps.EndTime;

% Get the data segments collection between the start and end time
SegmentsOfData = ItfData.Data(dStartTime, dEndTime);

% Take the first data segment
myFirstSegment = SegmentsOfData.Item(1);

% Check the number of samples
NumberOfSamples = myFirstSegment.NumberOfSamples;

% Next check is not needed but when you are dealing with a very
% large data segment this can delay the MATLAB execution.
% You can comment this out if you like.
if (NumberOfSamples > 500000)
    NumberOfSamples = 500000;
end

% Now the actual data is read.
% waveform(DataSourceResultType ResultDataType, int FirstSample,
%           int ResultCount, int Reduction);
%
% where:
% ResultDataType: The requested type of the result data.
%   Possible values:
%     Original = -1,
%     Int16 = 2,
%     Double64 = 5,
% FirstSample: First sample index from the selected segment to be
%              used for the result data
% ResultCount: Total number of samples to be returned
% Reduction: Reduction factor
waveformData = myFirstSegment.waveform(5, 1, NumberOfSamples, 1);

% Caculate the end time of the read data samples

```

```

tEnd = myFirstSegment.StartTime + ...
      (NumberOfSamples - 1) * myFirstSegment.SampleInterval;

% Create a time series for the X-Axis
t = myFirstSegment.StartTime: myFirstSegment.SampleInterval : tEnd;

% show the data in a diagram
figure1 = figure('NumberTitle','off','Name','Perception Recording File');

axes1 = axes('Parent',figure1,...
            'Position',[0.05 0.05 0.6 0.9]);

set(figure1, 'Position', [100 100 1200 800]);
plot(t, waveformData);

xlabel(['Time (',ItfData.XUnit,')']);
ylabel([myFirstChannel.Name, ' (', ItfData.YUnit,')']);

% Get the filename without path in our example this will be:
% 'MyRecordingFile' and the extension will be '.pnrf'
[pathstr,name,ext] = fileparts(MyRecordingName);
myTitle = [name, ext];

% Set the title of the diagram
title(myTitle);

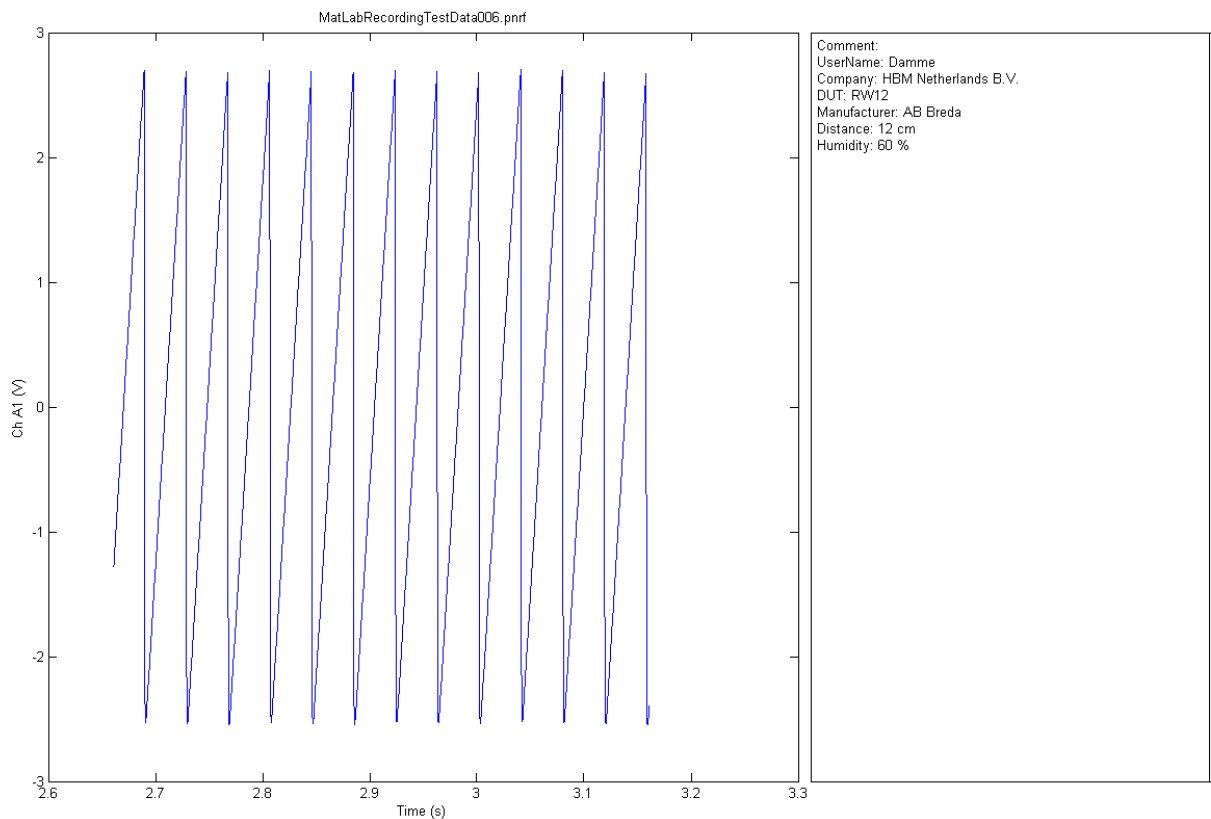
% Now show the data values. These values are entered in Perception via the
% Information sheet. They can contain recording specific information like
% the name of the device under test, test conditions like position or size
% or brand of device under test. It is up to the operator what information
% has been stored.
% The entered information from the information sheet is stored into
% the DataValues
myDataValues = MyData.DataValues;

% The number of data values
nCount = myDataValues.Count;

% Show all the data values
cInfo = '';
for i = 1:nCount
    myDataValue = myDataValues.Item(i);
    cInfo = sprintf('%s%s: %s %s\n', cInfo,myDataValue.Name,...
                    myDataValue.Value, myDataValue.Units);
end

% Create a Textbox
annotation(figure1,'textbox', [0.66 0.05 0.32 0.9], 'String',cInfo,...
          'FitBoxToText','off');

% The UTC time contains the date and time of the start of the recording
[year day time valid] = ItfData.GetUTCtime;
    
```



### 3 Example 2

Opening a pnrf file and showing the trigger information

```

%-- 22-1-14 12:00 --%
% Create PNRF reader OLE Automation server
FromDisk = actxserver('Perception.Loaders.pNRF');
% FromDisk = actxserver('Perception.Loaders.NRF');
% FromDisk = actxserver('Perception.Loaders.LRF');

% Use the open file dialog to select a recording file
[FileName,PathName] = uigetfile('*.pNRF', ...
    'Select a Perception Recording File');
MyRecordingName = [PathName, FileName];

% But you also can do this via code:
% MyRecordingName = 'D:\PNRFReader\Waveforms\MyRecordingFile.pnrf';

% Load your recording, myData is a Recording Interface
MyData = FromDisk.LoadRecording(MyRecordingName);

% Use the recording interface to get an interface to the 1st recorder
MyFirstRecorder = MyData.Recorders.Item(1);

% Get the trigger collection
myTriggers = MyFirstRecorder.Triggers;

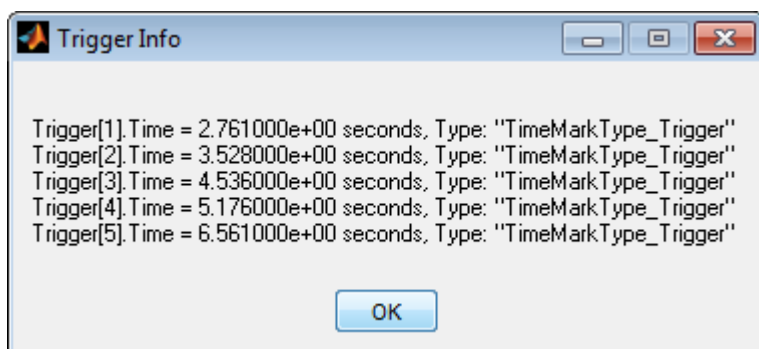
cInfo = '';
if (isempty(myTriggers))
    msgbox('No Triggers found', 'Triggers')
else

```



```

nTriggerCount = myTriggers.Count;
for i = 1:nTriggerCount
    myTrigger = myTriggers.Item(i);
    cInfo = sprintf('%sTrigger[%i].Time = %s seconds, Type: "%s"\n',...
        cInfo, i, myTrigger.Time, myTrigger.MarkType);
end
msgbox(cInfo, 'Trigger Info');
end
    
```



## 4 Example 3

Opening a pnrf file containing asynchronous data.

Using Real time formulas in Perception it is possible to save asynchronous data, each data point has a value and a time stamp and most likely the data is not equidistant. This example shows you how to open a pnrf file containing asynchronous data using MATLAB.

```

%-- 05-12-2020 12:00 --%
% Create PNRF reader OLE Automation server
FromDisk = actxserver('Perception.Loaders.pnrf');
% FromDisk = actxserver('Perception.Loaders.NRF');
% FromDisk = actxserver('Perception.Loaders.LRF');

% Use the open file dialog to select a recording file
[FileName,PathName] = uigetfile('*.pnrf', 'Select a Perception Recording File');
MyRecordingName = [PathName, FileName];

% But you also can do this via code:
% MyRecordingName = 'D:\PNRFReader\waveforms\MyRecordingFile.pnrf';

% Load your recording, myData is a Recording Interface
MyData = FromDisk.LoadRecording(MyRecordingName);

% Loop through all recorders and channels to get the names of all channels
% and store the names in a string array called channelNames
indexer = 0;
for r = 1: MyData.Recorders.Count

    % Use the recording interface to get an interface to the 1st recorder
    MyFirstRecorder = MyData.Recorders.Item(r);

    % From the first recorder we take the channel collection
    myChannels = MyFirstRecorder.Channels;
    isAsyncData = false;
    % From this channel collection we take the first channel
    for i = 1:myChannels.Count
        myChannel = myChannels.Item(i);
    
```

```

        Name = myChannel.Name;
        indexer = indexer + 1;
        channelNames(indexer) = string(Name);
        channel(indexer) = myChannel;
    end
end

% Show a modal selection list with all the channels so the operator can
% select a channel
[indx,tf] = listdlg('Liststring',channelNames);
myChannel = channel(indx);
SelectedChannelName = myChannel.Name;

% Now we get the DataSource (data source) interface from the selected channel
% The parameter 3 means: DataSourceSelect_Mixed
ItfData = myChannel.DataSource(3);

% findout if we have synchronous or asynchronous data by using the TimeInfo
% property
myTimeInfo = ItfData.TimeInfo;
if (myTimeInfo == 'DataSourceTimeInfo_Implicit')
    isAsyncData = false;
else
    isAsyncData = true;
end

% Get the sweeps collection from the datasource interface
MySweeps = ItfData.Sweeps;

% Get the start and end time of the recorded data of the first channel
dStartTime = MySweeps.StartTime;
dEndTime = MySweeps.EndTime;

% Get the data segments collection between the start and end time
SegmentsOfData = ItfData.Data(dStartTime, dEndTime);

% Take the first data segment
myFirstSegment = SegmentsOfData.Item(1);

% Check the number of samples
NumberOfSamples = myFirstSegment.NumberOfSamples;

% Next check is not needed but when you are dealing with a very
% large data segment this can delay the MATLAB execution.
% You can comment this out if you like.
if (NumberOfSamples > 500000)
    NumberOfSamples = 500000;
end

% show the data in a diagram
if isAsyncData
    title = append('Perception Recording File - Channel: ', SelectedChannelName, ' (Not-
    Equidistant data points)');
else
    title = append('Perception Recording File - Channel: ', SelectedChannelName, '
    (Equidistant data points)');
end
figure1 = figure('NumberTitle','off','Name', title);
axes1 = axes('Parent',figure1,'Position',[0.05 0.05 0.6 0.9]);
set(figure1, 'Position', [100 100 1200 800]);

% Now the actual data is read, for asynchronous data:
%
% WaveformWithTimes(DataSourceResultType ResultDataType, int FirstSample,
%                    int ResultCount, int Reduction);

```

```

%
% For synchronous data:
% Waveform(DataSourceType ResultDataType, int FirstSample,
%           int ResultCount, int Reduction);
%
% Where:
% ResultDataType: The requested type of the result data.
% Possible values:
%   Original = -1,
%   Int16 = 2,
%   Double64 = 5,
% FirstSample: First sample index from the selected segment to be
%              used for the result data
% ResultCount: Total number of samples to be returned
% Reduction: Reduction factor
if isAsyncData
    [WaveformDateWithTimes, Times] = myFirstSegment.WaveformWithTimes(5, 1, NumberOfSamples,
    1);
    plot (Times, WaveformDateWithTimes);
else
    WaveformData = myFirstSegment.Waveform(5, 1, NumberOfSamples, 1);

    % Caculate the end time of the read data samples
    tEnd = myFirstSegment.StartTime + (NumberOfSamples - 1) * myFirstSegment.SampleInterval;

    % Create a time series for the X-Axis
    t = myFirstSegment.StartTime: myFirstSegment.SampleInterval : tEnd;
    plot (t, WaveformData);
end
legend(SelectedChannelName);
xlabel(['Time (' ,ItfData.XUnit ,')']);
ylabel([SelectedChannelName, ' (' , ItfData.YUnit ,')']);

% Get the filename without path in our example this will be:
% 'MyRecordingFile' and the extension will be '.pnrf'
[pathstr,name,ext] = fileparts(MyRecordingName);
myTitle = [name, ext];

% Set the title of the diagram
title (myTitle);

% Now show the data values. These values are entered in Perception via the
% Information sheet. They can contain recording specific information like
% the name of the device under test, test conditions like position or size
% or brand of device under test. It is up to the operator what information
% has been stored.
% The entered information from the information sheet is stored into
% the DataValues
myDataValues = MyData.DataValues;

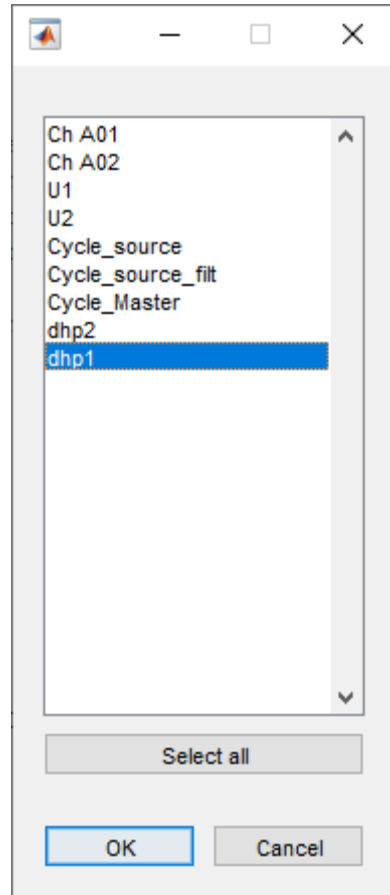
% The number of data values
nCount = myDataValues.Count;

% Show all the data values
cInfo = '';
for i = 1:nCount
    myDataValue = myDataValues.Item(i);
    cInfo = sprintf('%s%s: %s %s\n', cInfo,myDataValue.Name,...
    myDataValue.Value, myDataValue.Units);
end

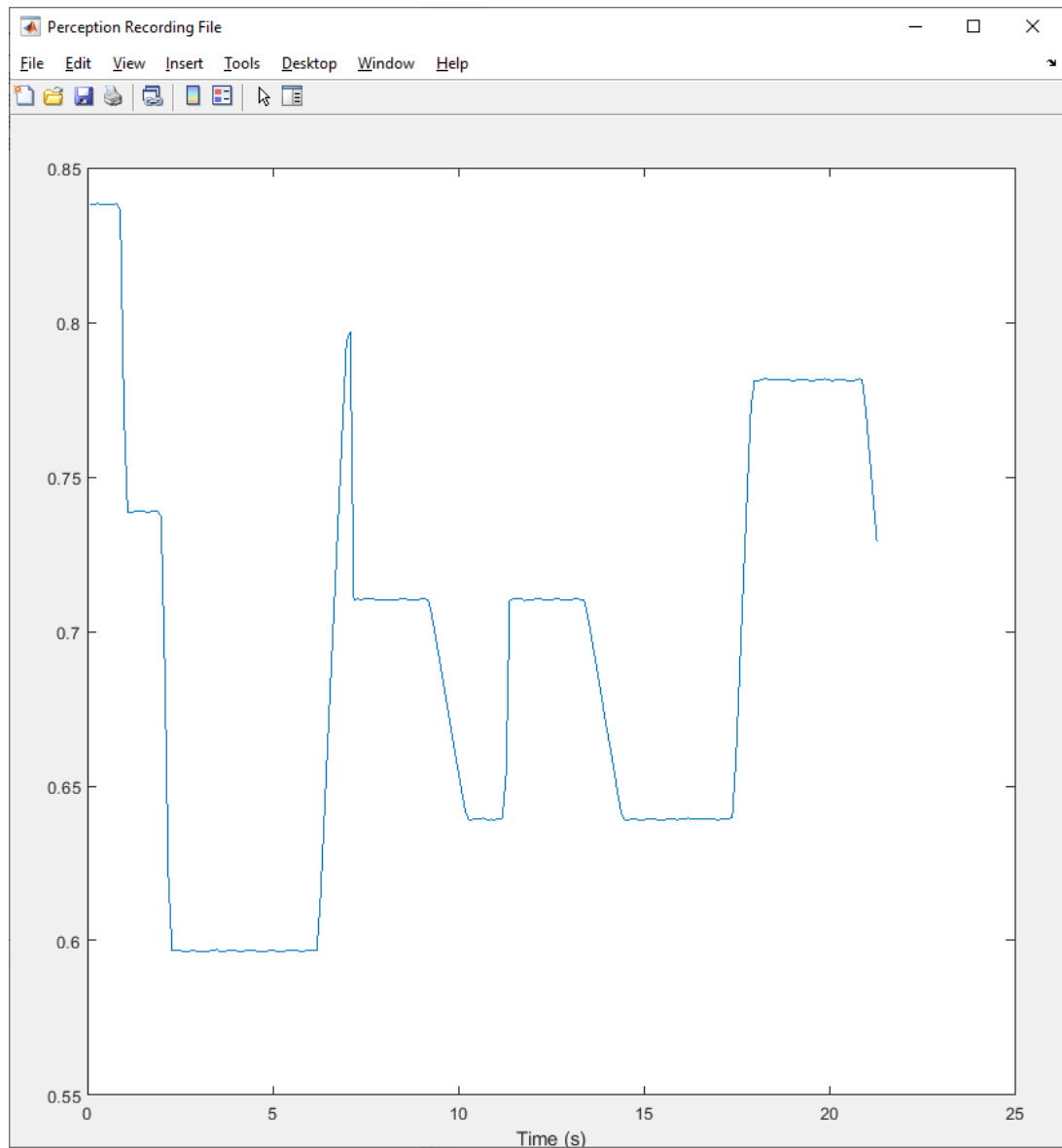
% Create a Textbox
annotation(figure1,'textbox', [0.66 0.05 0.32 0.9], 'string',cInfo,...
    'FitBoxToText','off');
    
```

```
% The UTC time contains the date and time of the start of the recording  
[year day time valid] = ItfData.GetUTCtime;
```

The channel selection dialog:



The output plot can look like:



Head Office

**HBM**

Im Tiefen See 45  
64293 Darmstadt  
Germany  
Tel: +49 6151 8030  
Email: info@hbm.com

France

**HBM France SAS**

46 rue du Champoreux  
BP76  
91542 Mennecey Cedex  
Tél: +33 (0)1 69 90 63 70  
Fax: +33 (0) 1 69 90 63 80  
Email: info@fr.hbm.com

Germany

**HBM Sales Office**

Carl-Zeiss-Ring 11-13  
85737 Ismaning  
Tel: +49 89 92 33 33 0  
Email: info@hbm.com

UK

**HBM United Kingdom**

1 Churchill Court, 58 Station Road  
North Harrow, Middlesex, HA2 7SA  
Tel: +44 (0) 208 515 6100  
Email: info@uk.hbm.com

USA

**HBM, Inc.**

19 Bartlett Street  
Marlborough, MA 01752, USA  
Tel : +1 (800) 578-4260  
Email: info@usa.hbm.com

PR China

**HBM Sales Office**

Room 2912, Jing Guang Centre  
Beijing, China 100020  
Tel: +86 10 6597 4006  
Email: hbmchina@hbm.com.cn

© Hottinger Baldwin Messtechnik GmbH. All rights reserved.  
All details describe our products in general form only.  
They are not to be understood as express warranty and do  
not constitute any liability whatsoever.

**measure and predict with confidence**

