TDTRP User Guide

Use Matlab to directly talk to TDT hardware

Updated 2025-02-27

Tucker-Davis Technologies
11930 Research Circle
Alachua, FL 32615 USA
Phone: +1.386.462.9622
Fax: +1.386.462.5365

Notices

The information contained in this document is provided "as is," and is subject to being changed, without notice. TDT shall not be liable for errors or damages in connection with the furnishing, use, or performance of this document or of any information contained herein.

The latest versions of TDT documents are always online at https://www.tdt.com/docs/

Table of Contents

Direct Hardware Access

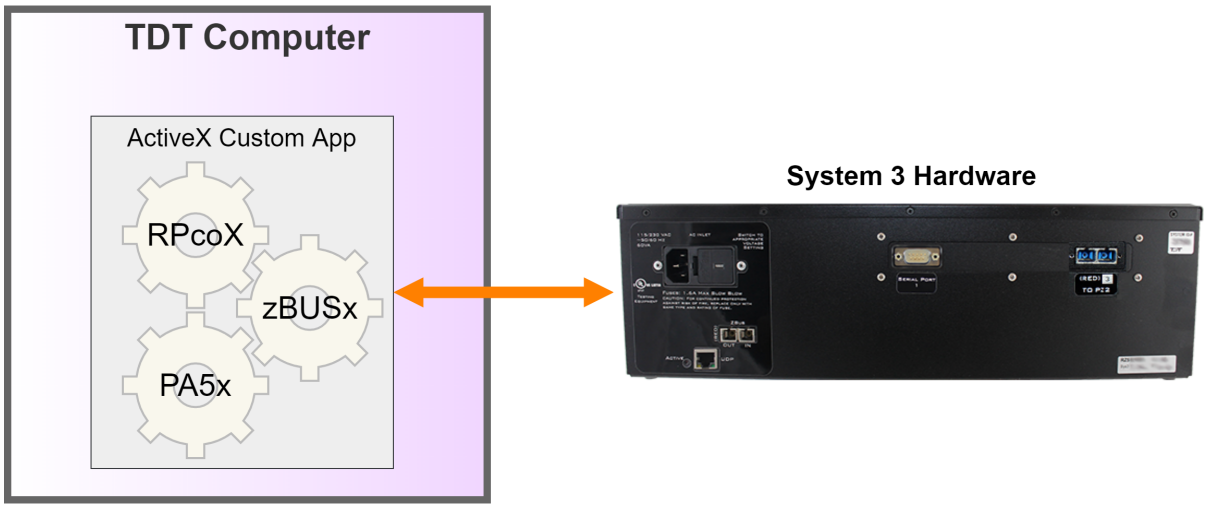
TDTRP User Guide

Use TDTRP to directly communicate with TDT hardware from MATLAB.

Requirements

- TDT Drivers
- TDT ActiveX package

About TDTRP



TDTRP is a MATLAB class that serves as a wrapper for the RPcoX ActiveX control that installs with the TDT ActiveX package. It provides direct access to TDT hardware (connect, load/run circuit, read/write tags on the hardware, send software triggers) and is generally easier to use than calling the RPcoX object directly.

TDTRP installs with the ActiveX controls into:

```
C:\TDT\ActiveX\ActXExamples\Matlab
```

or you can download it in the ActiveX Controls example zip.

TDTRP Functionality

This section includes methods, examples, and comparisons to RPcoX.

> **Note**
>
> See ActiveX User Reference for the complete RPcoX programming guide.

Creating a TDTRP object

```
obj = TDTRP(CIRCUITPATH, DEVICETYPE, 'parameter', value, ...)
```

`obj` is what you want the structure to be called for your code. In this document obj will be 'TDT' after we make our first call to TDTRP below. However, obj can be any unique variable name.

Input Parameter	Values
`'INTERFACE'`	String, interface type. Can be `'USB'`, `'USB3'`, or `'GB'` (default for fiber optic interface)
`'NUMBER'`	Integer, device number as enumerated by zBusMon (default 1)
`'FS'`	Float, sampling rate to run the circuit at (otherwise uses circuit default)

TDTRP Example

```
TDT = TDTRP('C:\TDT\ActiveX\ActXExamples\RP_Files\Continuous_Acquire.rcx',
'RZ6')
```

`TDT` is the returned object. The target device is an RZ6 processor. I only have one, so no need to set the 'NUMBER' parameter, and it has a fiber optic interface built into it, so it uses the default 'GB' interface.

RPcoX Method Comparison

```
RP = actxserver('RPco.X');
RP.ConnectRZ6('GB',1)
RP.LoadCOF('C:\TDT\ActiveX\ActXExamples\RP_Files\Continuous_Acquire.rcx');
e = RP.Run

if e == 0
    error('error running circuit')
else
    disp('Circuit ready to run')
end
```

Setting the mode with TDTRP

```
TDT.halt % stop any existing processing chain

TDT.load % load the processing chain

TDT.run % run the processing chain
```

TDTRP Example

```
>> circuit = 'C:\TDT\ActiveX\ActXExamples\RP_Files\Continuous_Acquire.rcx';

>> TDT.halt;

>> TDT.load(circuit);

>> TDT.run;

% Calling TDTRP() does the equivalent of calling halt, load, and run
% load and run the circuit onto the specified device
```

RPcoX Method Comparison

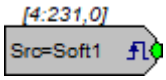
```
RP.Halt
circuit ='C:\TDT\ActiveX\ActXExamples\RP_Files\Continuous_Acquire.rcx';
e = RP.LoadCOF(circuit); % Loads circuit

if e == 0
    error('Error loading circuit')
else
    disp('Circuit ready to run')
end

e = RP.Run

if e == 0
    error('Error running circuit')
else
    disp('Circuit ready to run')
end
```

Software Triggers



Use software triggers to pulse a `TrigIn` component in RPvdsEx for one sample.

```
TDT.trg(TRIGNUM) % TRIGNUM is the software trigger index (1-10)
```

TRIGNUM determined by number after 'Soft' in the `TrigIn` component.

TDTRP Example

```
TDT.trg(1);
```

RPcoX Method

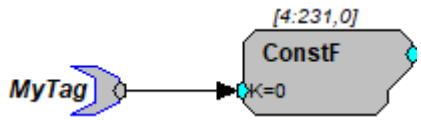
```
RP.SoftTrg(1);
```

Writing to Parameter Tags

```
TDT.write(TAGNAME, VALUE, 'parameter', value, ...)
```

`TAGNAME` is a string. `VALUE` is the single value or array that you want to push onto the tag. Whether the tag is pointed to a scalar component or buffer will determine what is a valid write. Examples of these paradigms are below in the examples below.

Input Parameter	Values
`'FORMAT'`	String, destination format (array only) options are 'F32' (default), 'I32', 'I16', 'I8'
`'OFFSET'`	Scalar, offset into buffer (array only). Default is 0



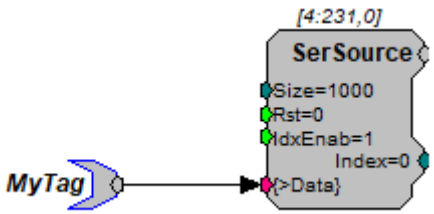
Parameter Tag pointing to a scalar value

TDTRP Example

```
% writing a single scalar value to a ConstF component
TDT.write('MyTag', 3.14);
```

RPcoX Method

```
RP.SetTagVal('MyTag', 3.14); % for single value
```



Parameter Tag pointing to memory buffer

TDTRP Example

```
% writing an array of values to a SerSource, which is a serial buffer
TDT.write('MyTag', 1:10);
```

RPcoX Method

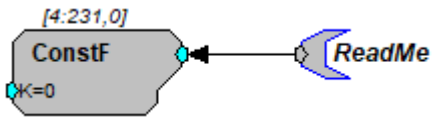
```
RP.WriteTagVEX('MyTag', 0, 'F32', 1:10); % for array of values
```

Reading from Parameter Tags

```
TDT.read(TAGNAME,'parameter',value,...)
```

Similar to the TDT.write method, what the tag is pointed to will determine what a valid read is. A scalar tag must be on a scalar component. An array must be on a buffer.

Input Parameter	Values
`'SOURCE'`	String, source format (array only) options are 'F32' (default), 'I32', 'I16', 'I8'
`'DEST'`	String, destination format (array only) options are 'F64', 'F32' (default), 'I32', 'I16', 'I8'
`'SIZE'`	Scalar, number of words to read (array only). Default is the entire buffer
`'OFFSET'`	Scalar, offset into buffer (array only). Default is 0
`'NCHAN'`	Scalar, number of channels in buffer (array only). Used for de-interlacing data. Default is 1



Parameter Tag pointing to a scalar value

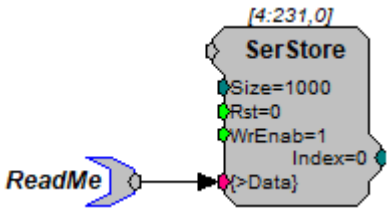
Parameter tags always point *towards* a component port.

TDTRP Example

```
% read single value parameter tags
ScalarVal = TDT.read('ReadMe');
```

RPcoX Method

```
RP.GetTagVal('ReadMe'); % for single value
```



Parameter Tag pointing to memory buffer

TDTRP Example

```
% This will read all values stored in the SerStore serial buffer
ArrVals = TDT.read('ReadMe');
```

RPcoX Method

```
RP.ReadTagVEX('ReadMe',0,1000,'F32','F32',1); % for 1 chan array of 1000
values
```

Examples Using TDTRP

You can find examples of ActiveX code using TDTRP installed with ActiveX in:

```
C:\TDT\ActiveX\ActXExamples\Matlab
```

or you can download it in the ActiveX Controls example zip.