TDTclient for BCI2000

J. Adam Wilson (jawilson@cae.wisc.edu)

University of Wisconsin – Madison
Department of Biomedical Engineering
Neural ***I***nterfaces ***T***echnology and ***R***esearch ***O***ptimization (**NITRO**) Lab

Chapter 1 - Introduction and Installation

1.1 Overview

The TDTclient program is a data acquisition tool for the BCI2000 software suite that uses the Tucker-Davis Technology (TDT) RX5 – Pentusa system. The Pentusa system is capable of multi-channel (up to 64), high data transfer rates (more than 1 GB / s) allowing for high density EEG, ECoG, or even single-unit recordings. The system is highly configurable, allowing the user to use digital filtering and complex analysis on the hardware.

1.2 Requirements

Currently, the TDTclient supports the following system configuration:

- TDT Pentusa/RX5
- GBIT card (no USB)
- ActiveX libraries installed (to install from source)
- BCI2000 Software

The obsolete Medusa system is not currently supported, but if there is enough of a demand, it can probably be implemented (contact jawilson@cae.wisc.edu for more details.) The same computer requirements apply as those for BCI2000.

1.3 Installation (binary)

If you are not installing from the source code, then there is not really any installation that needs to be done. It probably would not hurt to install the TDT ActiveX library from the TDT website (www.tdt.com), although it should not be necessary. However, the program has only been tested on a computer with the ActiveX libraries installed, so this has not yet been verified to be true.

1.4 Installation (source)

If you are installing from source, there are several steps to get the program working.

Install TDT Drivers

The most current TDT drivers should be downloaded and installed from www.tdt.com. Make sure that the microcode on the hardware is updated to at leaset version 58 as well, or the program WILL NOT WORK!!

Install TDT ActiveX Library

Currently, the ActiveX library must be installed on the local machine in order to use the TDTclient. It can be obtained from www.tdt.com, and it requires a password to install.

Import the TDT ActiveX Object into Borland C++Builder (MAY BE UNNECESSARY)

In the `EEGsource\TuckerDavis` directory, open the `TDTclient.bpr` with Borland C++Builder 6. On the menu, go to **Component → Import ActiveX Control…**, and in the list box search for **RPcoX ActiveX Control module**. Click on the **Install** button, and it will allow the object to be used.
(*Note, all paths are relative to the BCI2000 installation path, i.e.* `C:\BCI2000\`*, so the above path would be* `C:\BCI2000\EEGsource\TuckerDavis\`)

Build the Executable

On the menu, select **Project → Make Project**, and the file `TDTclient.exe` will be created. If you are building from the entire BCI2000 project source tree, selecting **Make All Projects** will also build the `TDTclient.exe` file. (Any errors in the build can be reported to jawilson@cae.wisc.edu).

1.5 Quick Start

The BCI2000 system requires for separate components to run: the EEG source module, signal processing module, application module, and operator module. The specific program set will depend on the desired application. To start a 1-or-2-dimensional cursor control task, start
`Operator\operat.exe`,
`EEGsource\TuckerDavis\TDTclient.exe`,
`SignalProcessing\AR\ARSignalProcessing.exe`, and
`Application\D2Box\D2box.exe`.

(Note: a *.bat file that automatically starts each of these can be used as well.)
Configuration notes for the module can be found later in this document.

Chapter 2 – The TDT RPvds chAcquisitionXX.rpx File

Anyone who has used the TDT system should be familiar with the RPvds circuit creation process. This chapter gives an overview of the circuit included with the TDTclient, and requirements if you wish to expand or modify the circuit. The included circuits chAcquire64.rpx and chAcquire16.rpx can be used for EEG or ECoG acquisition.

2.1 Basic Circuit Description

The chAcquire64.rpx and chAcquire16.rpx circuits were designed to be as simple as possible. They allow 16 or 64 channels of simultaneous acquisition, and includes a low-pass filter, high-pass filter, and 60 Hz notch filter. It has a semi-variable sampling rate, which based on the base sample rate of the TDT, which is 24414.0625 Hz. The main processor takes care of the basic timing of the circuit, and each processor handles the acquisition, filtering, and storage of 16 channels. So, a 2 processor system can handle 16 channels, and the 5 processor system can handle 64 channels.

2.2 Sample Rate Generation

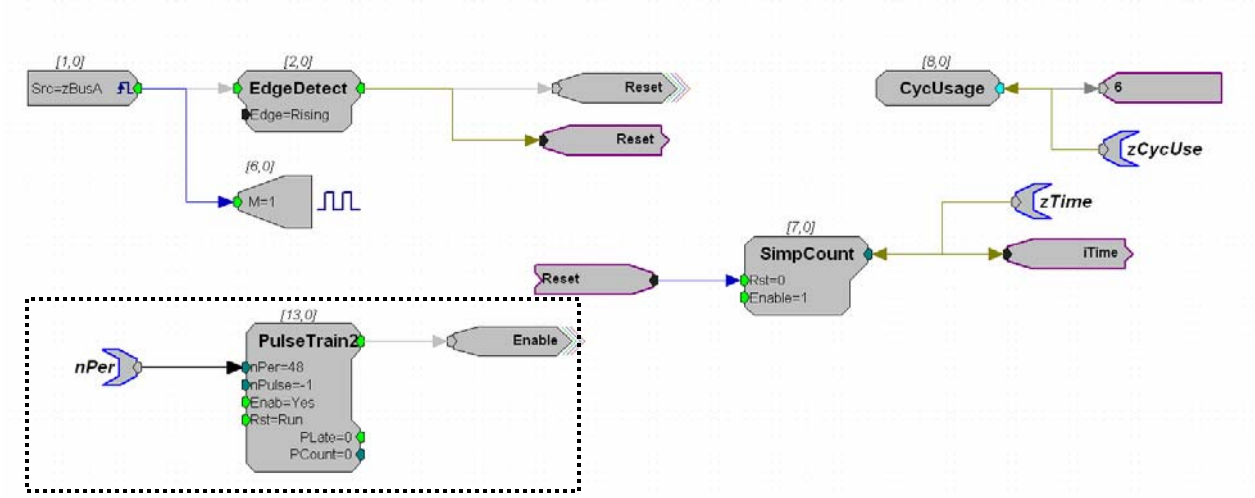

Figure 1 The main processor circuit which controls the sample rate via a pulse train. The period is set according to the closest sample period to the desired sample rate.

The main processor has a PulseGenerator construct, which sets the sample rate of the data buffer. It works by generating a pulse every *n* samples of the base rate, which triggers the buffer to read in a sample of every channel. So, if every 47th sample was used, the sample rate would effectively be 519.448 Hz, close to the more standard 512 Hz. There is a tool included with TDTclient called TDTsampleRate which will calculate the actual sample rate based on the desired rate. The period of the pulse generator is set in the BCI2000 TDTclient code based on the entered sample rate. The tag ***nPer*** is required by the TDTclient program, and will produce an error if not found in the circuit.

2.3 Data Acquisition, Filtering, and Storage

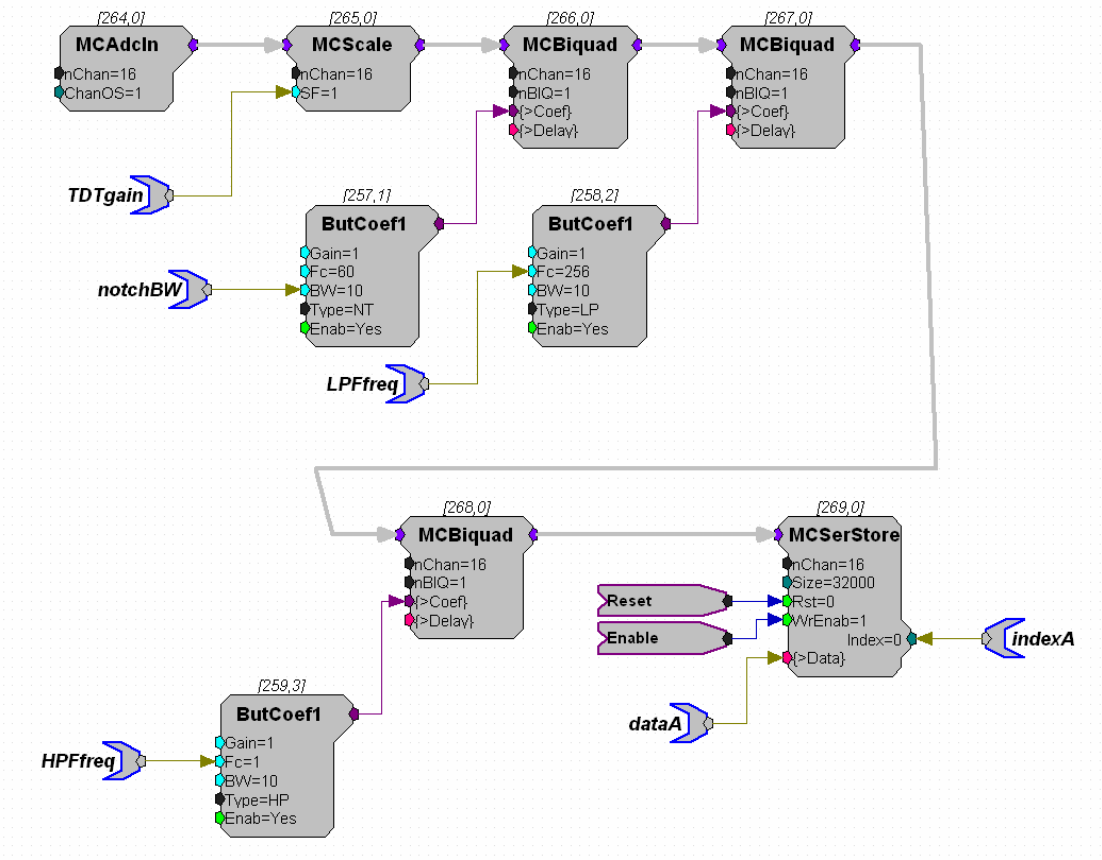

Figure 2 The data acquisition and filtering circuit.

The Pentusa makes use of the new multichannel components in the RPvds software, which allows designing processes for multiple channels easily. The MCAdcIn component is the input from the amplifier. This is then multiplied by the MCScale component, which is set by the **TDTgain** tag. (Note: all of the tags in **bold** are set by the TDTclient software, and are required for use.) The signal then goes through a series of Butterworth filters, the first of which is a 60 Hz notch filter. Instead of altering the corner frequencies, the bandwidth is set by the **notchBW** tag. If this is set at 10, then the corner frequencies will be at 55 and 65 hz for the notch. The low-pass filter is set next, and the corner frequency is set by the **LPFfreq** tag. Finally, the signal is high-pass filtered, which is set by the **HPFfreq** tag. These are all first order filters. The filter order can be increased by increasing the nBIQ number in the MCBiquad components, but the 1st order system seems to work find. In addition, keep in mind that there is a 1.5 Hz high pass filter on the Medusa preamps, so setting HPFfreq any lower will not accomplish much.

Finally, the signal is stored in a data buffer (MCSerStore). The Enable tag is set high by the pulse generator every **nPer** samples. Each buffer will have a different set of tags associated with them as well. On processor 1, the necessary tags are **dataA** and **indexA**. **dataA** is a pointer to the data in the buffer, which is used by TDTclient to read the data to the PC. **indexA** is the offset in the buffer of the current data sample, and is used by TDTclient to know where to start reading the data in the buffer. Similar tags are present on the other processors as well, although they are named **dataB** and **indexB** for processor 2, etc. The size for the MCSerStore SIZE parameter MAY be changed dynamically (there seems to be a disagreement of this fact

between the documentation and TDT customer support). It is currently set at 32000 bytes, although this may change in the future. Consult the TDT documentation for more information on each of these components.

2.4 Required Tags

This gives an overview of the tags that MUST be present in order for TDTclient to work correctly.

- nPer
- TDTgain
- notchBW
- LFPfreq
- HPFfreq
- dataA
- dataB
- dataC
- dataD
- indexA
- indexB
- indexC
- indexD

Experienced users may want to expand or modify these circuits with their own filters or other tools. This is perfectly fine as long as these parameter tags are present in the file.

Chapter 3 – BCI2000 Configuration

This chapter describes each of the parameters in the Source configuration tab in the BCI2000 config program.

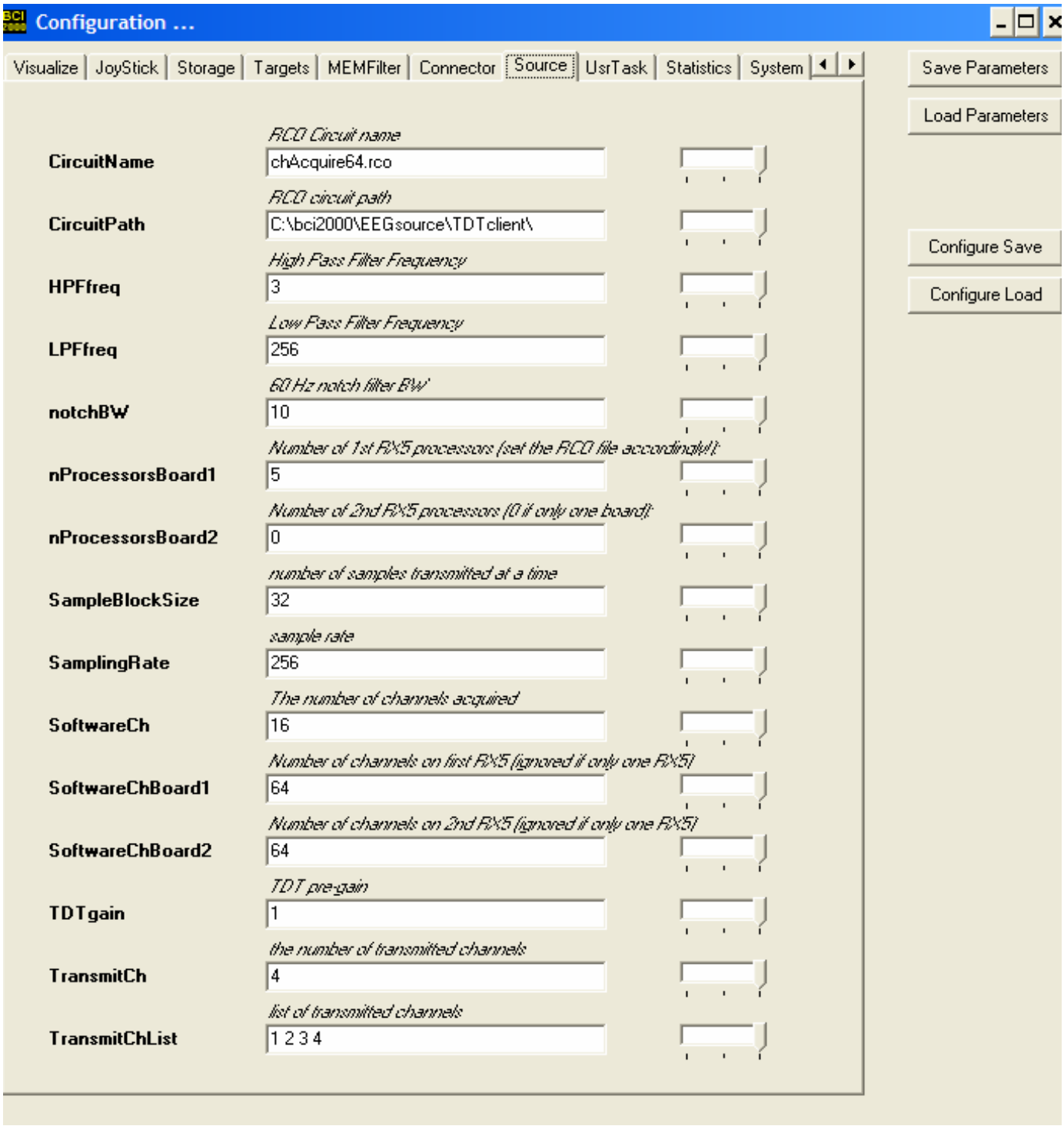

Figure 3 The source configuration screen in BCI2000.

Figure 3 shows the configuration screen that will be present when using the TDTclient. Several of the parameters are used in every acquisition program, while most are unique to the TDTclient. A list of each parameter and its description follows.

Circuit Name and Circuit Path – The *Circuit Path* is the directory path which contains the TDT *.rco file to be used. *CircuitName* is the actual *.rco file which will be used. The two default options are chAcquire64.rco and chAcquire16.rco. chAcquire64.rco should be used for 5 processor systems capable of 64 channel acquisition, while the chAcquire16.rco should be used for 2 processor systems capable of 16 channel acquisition.

HPFfreq – The corner frequency of the digital high-pass filter in the circuit. This can be set to 0 to not be used, but be aware that the Medusa pre-amp have an analog high-pass filter at 1.5 Hz built in.

LPFfreq – The corner frequency of the digital low-pass filter. This can be set to any value, but it is recommended that the highest value used is at half of the sampling frequency (i.e. 256 for a sample rate of 512).

notchBW – This is the bandwidth of the 60 Hz notch filter. If set at 10, the bandstop filter will have corner frequencies of 55 and 65 Hz.

nProcessorsBoard1 – The number of processors on your RX5. This can be either 5 or 2. 5 is the default value. *NOTE that if this is changed, the RCO file should be changed as well!!!*

nProcessorsBoard2 – The number of processors on the second RX5. This is set to 0 if there is only 1 system, and can be set to 0, 2, or 5. Currently, both systems will use the same RCO file, so if one system has 2 processors and the other 5, it can only use the 16 channel version of the RCO file.

SampleBlockSize – The number of sample points collected per channel at a time.

SampleRate – The sampling rate of the system. **Use the TDTsampleRate program to calculate the value to enter here!** The TDT has a fixed rate of 24414.0625 Hz, so the requested sample rate must be some integer fraction of this value. For example, a sample rate of 512 Hz is not possible, because it is impossible to find an integer that will divide 24414.0625 to get 512. The closest rate to 512 Hz that can be used is 519.448, because 24414.0625 / 519.448 = 47. This value (47) is used to set the period of the PulseTrain in figure 1.

SoftwareCh – The total number of channels collected. This should equal *softwareChBoard1 + softwareChBoard2*, the number of channels acquired on each system. Note that setting this to a smaller value will not reduce the amount of data transferred, because the entire amount of channels are always collected. This value tells the program how many channels to keep.

SoftwareChBoard1 – The number of channels collected from the first RX5. This is ignored if only one system is used.

SoftwareChBoard2 – The number of channels collected from the second RX5. This is ignored if only one system is used.

TDTgain – The amount of gain to add to the signal. Generally, between 10000 and 100000 is a good value to use, but test it to see what gives good signals.

TransmitCh – The number of channels which will be passed to the signal processing module of BCI2000.

TransmitChList – The list of channels which will be passed to the signal processing module of BCI2000.

Chapter 4 – TDTSampleRate

As mentioned previously, the TDT can only use sampling rates that are integer divisors of the base sampling rate of 24414.0625 Hz. The TDTsampleRate program takes the desired sample rate and calculates the closest sample rate that will be compatible with the TDT. To start it, go to the EEGSource/TuckerDavis/TDTsampleRate folder, and start TDTsampleRate.exe. Enter the desired sample rate in the appropriate box, press Calculate, and copy and paste the calculated rate into the BCI2000 configuration screen.