Corpus System 3 Hardware Emulator

Corpus is a software program that runs on any Windows PC. The Corpus program provides robust and seamless emulation of TDT's System 3 hardware via the standard driver interface. Corpus allows users to run any TDT or user program that relies on System 3 hardware without the physical hardware connected.

The Corpus version is synchronized with the TDT driver package releases. Corpus is available in TDT Drivers v86 and above. Corpus is free to all TDT customers and is automatically installed with TDT Drivers.

Hardware Emulation and PC Processor Capacity

Corpus operates via the same driver interface used to control standard TDT System 3 hardware. If the Corpus program is running when a program attempts to connect to System 3 hardware, the interface will be redirected to the software emulator and any connected hardware will not be accessed.

For Synapse users, Corpus will emulate the hardware configuration specified in the Rig Editor. The *.synrig file that contains the Rig Editor configuration can be found here: C:\Users\{username}\AppData\Local\Tucker-Davis Technologies\Synapse\.

For non-Synapse users, if this file is not found the rig file located at C:\TDT\Corpus\Corpus.synrig file will be loaded. The default file will emulate a PZ5-32 connected to an RZ2 equipped with 8 standard DSPs.

Corpus emulation capabilities are dependent on the processing power of a user's PC. The emulator will automatically assign cores and threads to realize optimal performance on a given platform. Corpus can be used efficiently on almost any machine, from laptops to high-end desktops, without being limited to the type of TDT hardware being emulated.

The core usage on a user's computer is not one-to-one with the emulation of standard RZDSPs. Indeed, Corpus would allocate the same resources if it was emulating one DSP or eight quad DSPs (QZDSPs).

Corpus can simulate hardware function in "pseudo" real-time. The emulator will block process chunks of discrete time samples, as would real hardware. Thus, if something does not work in Corpus, it likely will not work using real hardware. Unlike real hardware, however, Corpus will slow down processing and stretch time as its processing limits are exceeded. Status indicators, as shown in the Corpus window below, give feedback on processor load and speed, as well as other mode and error indicators.

The Corpus Window

Corpus Main Window
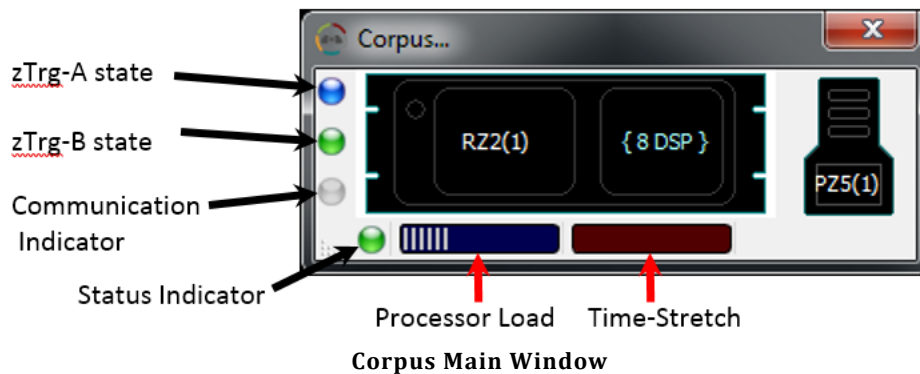
The System 3 devices that are supported by Corpus are listed below.

Device Name	Corpus Emulation Support
RZ2/RZ5x	Fully supported, but some hardware features not simulated or emulated
RZ6	Fully supported, excluding audio features
PZn	Can generate simulated neural signals, but not all device features fully emulated
UDP	Fully simulated, packets will be sent and received via host computer's Ethernet port
RS4/PO8e	Device is recognized but cannot be emulated
IZ2	Device is recognized but cannot be emulated
RV2	Device is recognized but cannot be emulated
RXn	Device is not supported by Corpus
RP2.1	Device is not supported by Corpus
RA16BA	Device is not supported by Corpus

Most DSP components have been implemented, however some components are either not appropriate for emulation or are not yet supported. If an unsupported RPvdsEx component is detected, the Status Indicator will turn red. Clicking on the indicator will show a debug window that lists the offending component(s).

Inputting Fake Data

Corpus allows you to load data files onto the emulated hardware and play them out, sample for sample, as if they can in on the actual hardware inputs.

A complete list of accepted file extensions and their associated hardware devices is shown in table 2.
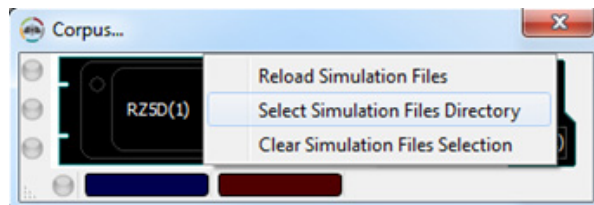
Hardware	Component	File Prefix	File Extension
RZn	ADC	adc	mat, wav
RZn	ADC	adcN	mat, wav, f32, i32, i16, txt
RZn	Digital I/O	Bits	mat, wav, f32, i32, i16, txt
PZn	PZ	chan	mat
PZn	PZ	chanN	mat, f32, txt

The scale factor for each data type is uniquely dependent on the hardware device being emulated. A table of scale factors for each type of hardware and device is shown below.

File Type	Hardware	Scale
mat	RZn	1e-1
	PZn	PZ5: 1e9/4 Other PZn devices: 1e9
wav	RZn	1e-1
	PZn	N/A
f32	RZn	1e-1
	PZn	PZ5: 1e9/4 Other PZn devices: 1e9
i32	RZn	ADC: $(2^{31})/10$, DIO: 1 (bits), 2^{16} (words)
	PZn	N/A
i16	RZn	ADC: $(2^{15})/10$, DIO: 1 (bits), 2^{8} (words)
	PZn	N/A
txt	RZn	1e-1
	PZn	PZ5: 1e9/4 Other PZn devices: 1e9

For certain file prefixes, the user may, and sometimes must, specify the specific channel N that the data will be loaded on. For example, *.f32 files must have a specified channel onto which they are loaded onto an RZ device. Thus, the file must be named "adc1.f32" if the user wishes to load the data into channel 1 of an RZn ADC port.

To load fake data files, the user must direct corpus to an appropriate directory that contains the intended hardware device and the files to the simulated on that device. The folder tree should be formatted as C:\TDT\Corpus\SimulationFiles\<experiment>\<hardware_device_name>. The user can load the appropriate directory by right-clicking in Corpus and choosing "Select Simulation Files Directory," and browsing to the highest folder in the desired directory (typically the <experiment> folder). A new window should appear when corpus is loading and completes loading the data.

The maximum files size for loading simulation data into Corpus is 256 million samples. If a user's file is too large, Corpus will not load any of the data and instead provide a message in the loading dialog box.

Learn More About Corpus

TDT support offers a suit of Synapse-based instructional videos, which includes one dedicated to Corpus, on their website found here: https://www.tdt.com/training-videos/