Powersim

COM Automation API

DLL Version 1.0.0.3
Document Version 1.0.0.0
26 March 2017

Table of Contents

Summary

The Powersim automation interface allows you to configure and control Powersims for automated test control. Python and C# examples are available upon request.

Interfaces

The functions described below describe the interfaces available for communicating with a Powersim device.

Any invalid parameters or function requests will result in a COM error which can be handled with the ErrorInfo class defined by Microsoft.

Connection

Before any other interfaces can be called, a connection must be made using one of the follow functions. Making a connection will disconnect any previous connection.

SetSocketConnection

Sets a socket connection to an Ethernet connected Powersim using the hostname and port specified. The default port on all Powersim devices is 8080.

Parameters		
hostName	BSTR (string)	Hostname or IP address of the Powersim device
port	UINT	IP port (range: 0 – 65535)
Returns		
Nothing		

SetSerialConnection

Sets a serial connection over a local RS232 connection to a Powersim device using the specified Comm port.

Parameters		
portName	BSTR (string)	Comm port to connect to (ex: “COM3”)
Returns		
Nothing		

Waveforms

These interfaces control the static output and waveform test modes. For static output control, set index 0 of the waveform, harmonic, and test sequence control, and call the default write for each before calling StartDefaultWaveform. To run a waveform test, build up all the waveforms and harmonics, and the list of test sequences, then call the write for each group before calling StartTestSequence.

ClearWaveforms

Clears all waveforms in the Powersim object.

NOTE: This does not write any data to the Powersim device. Use WriteWaveforms to write all Waveforms to the device.

Parameters		
None		
Returns		
Nothing		

SetWaveformVoltage

Sets the voltages of a waveform in Volts.
Phase A angle is always set to 0.

NOTE: This does not write any data to the Powersim device. Use WriteWaveforms to write all Waveforms to the device.

Parameters		
index	ULONG	Index of the waveform [0..99]
vA	FLOAT	Phase A voltage in Volts RMS.
vB	FLOAT	Phase B voltage in Volts RMS.
vC	FLOAT	Phase C voltage in Volts RMS.
angleB	FLOAT	Phase angle of Phase B voltage. [-360..360 degrees]
angleC	FLOAT	Phase angle of Phase C voltage. [-360..360 degrees]
Returns		
Nothing		

SetWaveformCurrent

Sets the currents of a waveform in Amps.
All phase angles are in reference to the Phase A voltage (0 degrees).

NOTE: This does not write any data to the Powersim device. Use WriteWaveforms to write all Waveforms to the device.

Parameters		
index	ULONG	Index of the waveform [0..99]
iA	FLOAT	Phase A current in Amps RMS.
iB	FLOAT	Phase B current in Amps RMS.
iC	FLOAT	Phase C current in Amps RMS.
angleA	FLOAT	Phase A angle reference to phase A Voltage in degrees [0..360.00 degrees]
angleB	FLOAT	Phase B angle reference to phase A Voltage in degrees [0..360.00 degrees]
angleC	FLOAT	Phase C angle reference to phase A Voltage in degrees [0..360.00 degrees]
Returns		
Nothing		

SetDCLevels

Sets the DC voltage levels of a waveform for Channel 7 (level1) and Channel 8 (level2) in Volts.

NOTE: This does not write any data to the Powersim device. Use WriteWaveforms to write all Waveforms to the device.

Parameters		
index	ULONG	Index of the waveform [0..99]
level1	FLOAT	DC voltage applied to channel 7 [0..10V]
level2	FLOAT	DC voltage applied to channel 8 [0..10V]
Returns		
Nothing		

SetFrequency

Sets the sine wave frequency of a waveform in hertz (Hz). This value must be greater than 0 Hz and no more than 80 Hz.

NOTE: This does not write any data to the Powersim device. Use WriteWaveforms to write all Waveforms to the device.

Parameters		
index	ULONG	Index of the waveform [0..99]
frequency	FLOAT	Frequency in Hz (0..80 Hz]
Returns		
Nothing		

WriteWaveforms

Writes all waveforms to the Powersim device.

Parameters		
None		
Returns		
Nothing		

WriteDefaultWaveform

Writes only the first waveform to the Powersim device (index 0). This is the default waveform.

Parameters		
None		
Returns		
Nothing		

ClearHarmonics

Clears all harmonic entries in the Powersim object.

NOTE: This does not write any data to the Powersim device. Use WriteHarmonics to write all Harmonics to the device.

Parameters

None		
Returns		
Nothing		

SetHarmonic

Sets a harmonic entry at the given index to the Powersim object.

Each harmonic entry can have up 15 harmonic waveforms added to it. Call this function multiple times with a different harmonicNumber [0..14] to set multiple harmonics for one entry.

*NOTE: This does not write any data to the Powersim device. Use **WriteHarmonics** to write all Harmonics to the device.*

Parameters		
index	ULONG	Index of the harmonic entry [0..29]
harmonicIndex	ULONG	Harmonic waveform index [0..14]
harmonicNumber	ULONG	Harmonic[2..100]
vA	FLOAT	Phase A voltage in Volts RMS.
vB	FLOAT	Phase B voltage in Volts RMS.
vC	FLOAT	Phase C voltage in Volts RMS.
iA	FLOAT	Phase A current in Amps RMS.
iB	FLOAT	Phase B current in Amps RMS.
iC	FLOAT	Phase C current in Amps RMS.
Returns		
Nothing		

WriteHarmonics

Writes all harmonics to the Powersim device.

Harmonics are set/cleared using ***SetHarmonic*** and ***ClearHarmonics***.

Parameters		
None		
Returns		
Nothing		

WriteDefaultHarmonic

Writes only the first harmonic to the Powersim device (index 0). This is the default harmonic.

Parameters		
None		
Returns		
Nothing		

ClearSequences

Clears all test sequences in the Powersim object.

NOTE: This does not write any data to the Powersim device. Use WriteSequences to write all test sequences to the device.

Parameters		
None		
Returns		
Nothing		

SetSequence

Set a test sequence entry into the Powersim device.

NOTE: This does not write any data to the Powersim device. Use WriteSequences to write all test sequences to the device.

Parameters		
index	ULONG	Index of the test sequence [0..999]
waveformIndex	ULONG	Index of the waveform entry to use [0..99]
harmonicIndex	ULONG	Index of the harmonic entry to use [0..29]
numCycles	ULONG	Number of cycles to run this waveform in the test.
Returns		
Nothing		

WriteSequences

Writes all test sequences to the Powersim device.
Sequences are set/cleared using SetSequence and ClearSequences.

Parameters		
None		
Returns		
Nothing		

StartDefaultWaveform

Starts the default waveform on the Powersim device indefinitely.
The default waveform can be set using index 0 and written to the device using WriteDefaultWaveform.

To stop the waveform output, use StopWaveform.

Parameters		
None		
Returns		
Nothing		

StartTestSequence

Outputs the waveforms using the test sequences defined using SetSequence and WriteSequences.

The output will stop when the test sequence is complete or can be stopped using StopWaveform.

Parameters		
None		
Returns		
Nothing		

StopWaveform

Stops any output currently active on the Powersim device.

Parameters		
None		
Returns		
Nothing		

Events

These interfaces are used to build and control event tests. To add events, you will modify an individual event record buffer by setting the trigger time, and parameters for the type of event it is, then you add the event to the list (nothing is saved to the list until it is added) using AddEvent. After all events are added, they can be written to the device using WriteEvents, or saved to a file with SaveEventsToFile.

SetOperationalMode

Sets the operational mode of the Powersim device.

Parameters		
runInEventMode	BOOL	This indicates that the device will run in event mode. This needs to be set for the initial state of service config to be set at the beginning of the test
alwaysEnergize	BOOL	This will set the outputs to the service config at powerup.
runTestRelativeToStart	BOOL	This indicates that the event timestamps are relative to the start of the test, rather than UTC time
Returns		
Nothing		

SetServiceType

Sets the base type of the service.

NOTE: This does not write any data to the Powersim device. Use WriteServiceConfig to write the service config to the device.

Parameters		
serviceType	BYTE	1 – Single Phase 2 – Two Phase 3 – Three Phase 4 – 2 single phase 5 – 3 single phase
frequency	FLOAT	Frequency of service (typically 50 or 60) (0..80 Hz]
Returns		
Nothing		

SetServiceVoltage

Sets the service voltage config.

NOTE: This does not write any data to the Powersim device. Use WriteServiceConfig to write the service config to the device.

Parameters		
vA	FLOAT	Phase A voltage in Volts RMS.
vB	FLOAT	Phase B voltage in Volts RMS.
vC	FLOAT	Phase C voltage in Volts RMS.
angleB	FLOAT	Phase angle of Phase B voltage. [-360..360 degrees]
angleC	FLOAT	Phase angle of Phase C voltage. [-360..360 degrees]
Returns		
Nothing		

SetServiceCurrent

Sets the current for the base service.

NOTE: This does not write any data to the Powersim device. Use WriteServiceConfig to write the service config to the device.

Parameters		
iA	FLOAT	Phase A current in Amps RMS.
iB	FLOAT	Phase B current in Amps RMS.
iC	FLOAT	Phase C current in Amps RMS.
angleA	FLOAT	Phase A angle reference to phase A Voltage in degrees [-360..360 degrees]
angleB	FLOAT	Phase B angle reference to phase A Voltage in degrees [-360..360 degrees]
angleC	FLOAT	Phase C angle reference to phase A Voltage in degrees [-360..360 degrees]
Returns		
Nothing		

WriteServiceConfig

Writes the service configuration to the device.

Parameters		
None		
Returns		
Nothing		

ClearEvents

Clears out all the events in the device so that new test events can be written.

*NOTE: This does not write any data to the Powersim device. Use **WriteEvents** to clear all events in the device.*

Parameters		
None		
Returns		
Nothing		

SetEventTimeRecurring

Sets the event time of the current event being edited the specified recurring time.

*NOTE: This does not write any data to the Powersim device. Add the event with **AddEvent** then write all events with **WriteEvents**.*

Parameters		
recurringType	BYTE	0 – period since anchor time 1 – offset in day 2 – offset Sunday 3 – offset Monday 4 – offset Tuesday 5 – offset Wednesday 6 – offset Thursday 7 – offset Friday 8 – offset Saturday
offset	ULONG	Offset in seconds from the start point defined by recurring time
period	ULONG	Period between events after initial offset (zero is single event)
Returns		
Nothing		

SetEventTimeNonRecurring

Sets the event time of the current event being edited to the specified non-recurring time.

*NOTE: This does not write any data to the Powersim device. Add the event to the Powersim object with **AddEvent** then write all events to the device with **WriteEvents**.*

Parameters

time	ULONG	Time when the event will occur. This is in UTC unless the operation made is set to have events run relative to start. In that case it is just the time since the start of the test. See SetOperationalMode.
Returns		
Nothing		

SetWaveformEvent

Sets the parameters of the current event being edited to a waveform type event with the following parameters.

NOTE: This does not write any data to the Powersim device. Add the event to the Powersim object with AddEvent then write all events to the device with WriteEvents.

Parameters		
eventType	BYTE	0 – voltage (units in Volts RMS) 1 – current (units in Amps RMS) 2 – current angle (units in [-360..360 degrees]) 3 – voltage angle (units in [-360..360 degrees])
setA	BOOL	Sets the value, a, for phase A
setB	BOOL	Sets the value, b, for phase B
setC	BOOL	Sets the value, c, for phase C
a	FLOAT	Value for A
b	FLOAT	Value for B
c	FLOAT	Value for C
duration	ULONG	Length of time in cycles that the change should stay in effect. 0 if permanent.
Returns		
Nothing		

SetHarmonicsEvent

Sets the parameters of the current event being edited to a harmonic event with the following parameters.

NOTE: This does not write any data to the Powersim device. Add the event to the Powersim object with AddEvent then write all events to the device with WriteEvents.

Parameters		
eventType	BYTE	6 – voltage harmonic 7 – current harmonic
setA	BOOL	Sets the value, a, for phase A
setB	BOOL	Sets the value, b, for phase B
setC	BOOL	Sets the value, c, for phase C
a	FLOAT	Value for A harmonic
b	FLOAT	Value for B harmonic
c	FLOAT	Value for C harmonic

harmonic	ULONG	Harmonic number
duration	ULONG	Length of time in cycles that the change should stay in effect. 0 if permanent.
Returns		
Nothing		

SetRelayEvent

Sets the parameters of the current event being edited to a relay event with the following parameters.

*NOTE: This does not write any data to the Powersim device. Add the event to the Powersim object with **AddEvent** then write all events to the device with **WriteEvents**.*

Parameters		
closed	BOOL	If true, relay closes, otherwise it opens
duration	ULONG	Length of time in cycles to leave relay in new state. 0 if permanent.
Returns		
Nothing		

SetAuxOutputEvent

Sets the parameters of the current event being edited to an aux output event with the following parameters.

*NOTE: This does not write any data to the Powersim device. Add the event to the Powersim object with **AddEvent** then write all events to the device with **WriteEvents**.*

Parameters		
set1	BOOL	Modify aux1
set2	BOOL	Modify aux2
aux1	FLOAT	Value for aux1
aux2	FLOAT	Value for aux2
duration	ULONG	Length of time in line cycles to leave the change in place. 0 if permanent.
Returns		
Nothing		

AddEvent

Adds the event currently being edited to the list of events.

*NOTE: This does not write any data to the Powersim device. Write all events to the device with **WriteEvents**.*

Parameters		
None		

Returns		
Nothing		

WriteEvents

Writes all the events in the Powersim object to the device.

Parameters		
None		
Returns		
Nothing		

SetTime

Sets the time in the device for use in event test mode. This is unnecessary if the device is networked with NTP enabled.

Parameters		
time	ULONG	UTC time
Returns		
Nothing		

StartTest

Starts the event test.

Parameters		
None		
Returns		
Nothing		

StopTest

Stops the event test.

Parameters		
None		
Returns		
Nothing		

Universal Methods

These functions are used to read the status and event log from the most recent test. The most recent test is the one currently running, or the last test ran if the test is already stopped.

ReadStatus

Reads the test status out of the currently connected Powersim device and returns is as a StatusDefn structure.

Parameters		
None		

Returns		
Status	StatusDefn	Status information. See StatusDefn.

SetRelay

Sets the value of the relay

Parameters		
on	BOOL	Closed if true, open otherwise
Returns		
Nothing		

Data Storage and Retrieval

These functions are used for storing and loading data to and from files.

LoadWaveformsFromFile

Loads all waveforms from an xml formatted file (*.wvf). These waveforms are loaded into the Powersim object but not yet written to the device.

To write the waveforms to the device, use the WriteWaveforms function.

Waveform files can be created using the SaveWaveformsToFile function or with the Powersim application.

Parameters		
filename	BSTR (string)	Waveform filename. (file extension: *.wvf)
Returns		
Nothing		

SaveWaveformsToFile

Saves the current waveforms in the Powersim object to an xml formatted file (*.wvf).
This file can be loaded using the LoadWaveformsFromFile function or the Powersim application.
If the file already exists, it will be overwritten.

Parameters		
filename	BSTR (string)	Waveform filename (include the extension *. wvf)
Returns		
Nothing		

LoadEventsFromFile

Loads all events from an xml formatted file (*.evt). These events are loaded into the Powersim object but not yet written to the device.

To write the waveforms to the device, use the WriteEvents function.

Event files can be created using the SaveEventsToFile function or with the Powersim application.

Parameters		
filename	BSTR (string)	Waveform filename. (file extension: *.evt)
Returns		
Nothing		

SaveEventsToFile

Saves the current events in the Powersim object to an xml formatted file (*.evt).
This file can be loaded using the LoadEventsFromFile function or the Powersim application.
If the file already exists, it will be overwritten.

Parameters		
filename	BSTR (string)	Waveform filename (include the extension *. evt)
Returns		
Nothing		

Calibration

The process for calibrating the unit for your meter starts with configuring the output range for each output using SetSpanConfig. Then you generate the rough calibration values by setting the apparent voltage and current at full scale output by calling SetFullScaleConfig. You then enable the output by calling StartCalibration with the values you would like to do your calibration on (typically mid-scale values). At this point you read the values that your meter is showing, and call SetCalibration with those values to calculate new calibration values, and to update the output. You can repeatedly call this function to refine the calibration until you reach the desired level of precision.

SetSpanConfig

Sets the output span configuration for each output channel.

Parameters		
vA	BYTE	0 – 0-5V, 1 – 0-10V, 2 – +/- 5V, 3 – +/- 10V, 4 – +/- 2.5V
vB	BYTE	0 – 0-5V, 1 – 0-10V, 2 – +/- 5V, 3 – +/- 10V, 4 – +/- 2.5V
vC	BYTE	0 – 0-5V, 1 – 0-10V, 2 – +/- 5V, 3 – +/- 10V, 4 – +/- 2.5V
iA	BYTE	0 – 0-5V, 1 – 0-10V, 2 – +/- 5V, 3 – +/- 10V, 4 – +/- 2.5V
iB	BYTE	0 – 0-5V, 1 – 0-10V, 2 – +/- 5V, 3 – +/- 10V, 4 – +/- 2.5V
iC	BYTE	0 – 0-5V, 1 – 0-10V, 2 – +/- 5V, 3 – +/- 10V, 4 – +/- 2.5V
aux1	BYTE	0 – 0-5V, 1 – 0-10V, 2 – +/- 5V, 3 – +/- 10V, 4 – +/- 2.5V
aux2	BYTE	0 – 0-5V, 1 – 0-10V, 2 – +/- 5V, 3 – +/- 10V, 4 – +/- 2.5V
Returns		
Nothing		

SetFullScaleConfig

Generates a starting point calibration value where the entered voltage and current values will be the full-scale output. Used as the first step of calibration.

Parameters		
voltage	FLOAT	Voltage at full scale output
current	FLOAT	Current at full scale output

Returns		
Nothing		

StartCalibration

Outputs a waveform on all channels with the specified voltage, current and frequency. This is the second step of calibration.

Parameters		
voltage	FLOAT	Voltage to output
current	FLOAT	Current to output
frequency	FLOAT	Frequency of waveforms
Returns		
Nothing		

SetCalibration

Updates the calibration values using the values that are seen by the meter. Also updates the output values with is what is set in StartCalibration. This is the last step in calibration, and can be called repeatedly to dial in the calibration values.

Parameters		
vA	FLOAT	Value for phase A voltage as reported by meter
vB	FLOAT	Value for phase B voltage as reported by meter
vC	FLOAT	Value for phase C voltage as reported by meter
iA	FLOAT	Value for phase A current as reported by meter
iB	FLOAT	Value for phase B current as reported by meter
iC	FLOAT	Value for phase C current as reported by meter
Returns		
Nothing		

Identification

These functions are used to identify the currently connected Powersim device.

GetFixtureSerialNumber

Gets the serial number of the currently connected Powersim device as a string.

Parameters		
None		
Returns		
	BSTR (string)	Serial number (ex: “001002301”)

GetFixtureVersion

Gets the version of the currently connected Powersim device as a string.

Parameters		
None		
Returns		

	BSTR (string)	Version number as string (ex: “1.0”)

GetFixtureName

Gets the fixture name of the currently connected Powersim device as a string.

Parameters		
None		
Returns		
	BSTR (string)	Fixture name (ex: “Lab Fixture 1”)

Data Structures

StatusDefn

The StatusDefn structure is returned by the ReadStatus function.

```
typedef struct
{
   VARIANT_BOOL isRunningSequence;    // true if currently running test sequence
   VARIANT_BOOL isRunningDefault;     // true if currently running the default waveform (index 0)
   VARIANT_BOOL isRunningWaveform;    // true if currently running waveform
   ULONG currentWaveform;             // index of the waveform currently running
   ULONG cyclesRemaining;             // cycles remaining for the current waveform (0 => forever)
   VARIANT_BOOL isInvalidFrequency;   // true if a waveform has a frequency of 0 or > allowed (80)
   VARIANT_BOOL isOverflow;           // true if a waveform has a calculated value > 10V
} StatusDefn;
```