



ADSANTEC

Ultra High-Speed Mixed Signal ASICs

Advanced Science And Novel Technology Company, Inc.
2790 Skypark Drive, Suite 112, Torrance, CA 90505

Offices: 310-530-9400 Fax: 310-530-9402
www.adsantec.com

ASNT_PRBS45 PRBS Generator USER GUIDE





Overview

The ADSANTEC Inc. ASNT_PRBS45 has a single PRBS Channel. The unit generates NRZ (Non Return to Zero) industry standard PRBS patterns. The following are the selectable PRBS patterns that can be used: PRBS7, PRBS9, PRBS11, PRBS15, PRBS17, PRBS20, PRBS23, PRBS29, and PRBS31. The unit also can generate a custom 256 bit pattern to which the user uploads a text file containing the data through the GUI, and user selectable error injection which can be set up through the GUI or through importing of a text file. A single-ended half-rate clock is required for operation. A single-ended, and differential clock output are provided by the clock input. A single-ended sync output is used to trigger an oscilloscope used to view a PRBS7 pattern or eye diagram. The unit is controlled through a GUI or DLL with example python code. The DLL is a 32-bit and you must use a 32-bit compiler/interpreter for it to work.

Getting Started

Connect the provided +5V AC/DC adapter to the unit. Power the unit on by turning the rocker switch on the back panel to the “ON” position. Observe a **green** light on the front panel labeled PWR illuminate and the fan starting up. Next connect the provided USB cable to the computer and the unit. The USB connection on the back panel is a USB-B jack. Observe a **green** light illuminate on the front panel labeled USB when the USB cable is connected to the computer and unit.





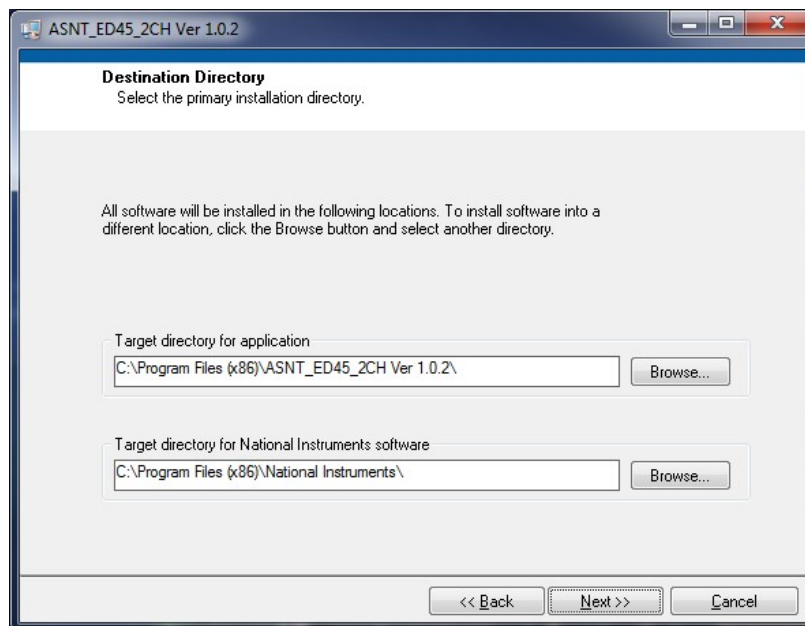
1.0 Software

If the software has already been installed go to **1.8**

1.1 Locate the installation files given as shown below. Double left-click on *setup.exe*

File Name	Modified	Type	Size
bin	4/8/2014 1:32 PM	File folder	
license	4/8/2014 1:32 PM	File folder	
supportfiles	4/8/2014 1:33 PM	File folder	
CDM v2.10.00 WHQL Certified.exe	4/8/2014 12:09 PM	Application	1,913 KB
dp.pmf	4/8/2014 1:33 PM	PMF File	1 KB
nidist.id	4/8/2014 1:33 PM	ID File	1 KB
setup.exe	10/7/2013 4:35 PM	Application	1,394 KB
setup.ini	4/8/2014 1:33 PM	Configuration sett...	5 KB

1.2 Select Target directory for the location of the installation files that will be installed, then left-click on *Next*.





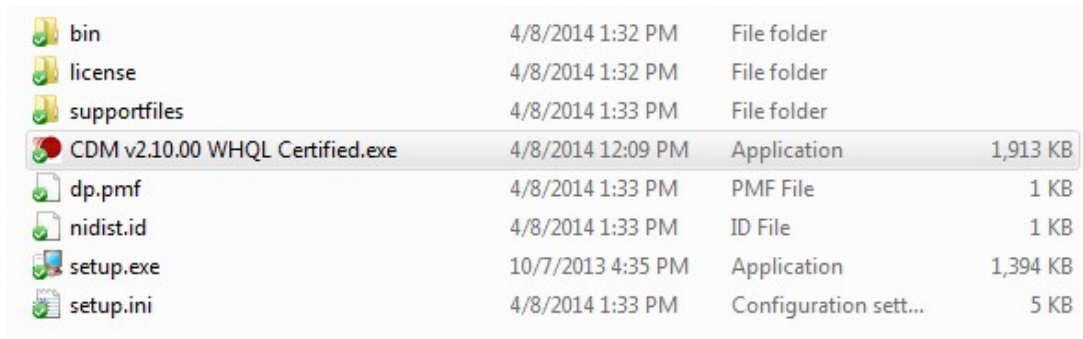
1.3 Left-click on *Next*.

1.4 Wait for the files to be installed.

1.5 Left-click on *Finish*

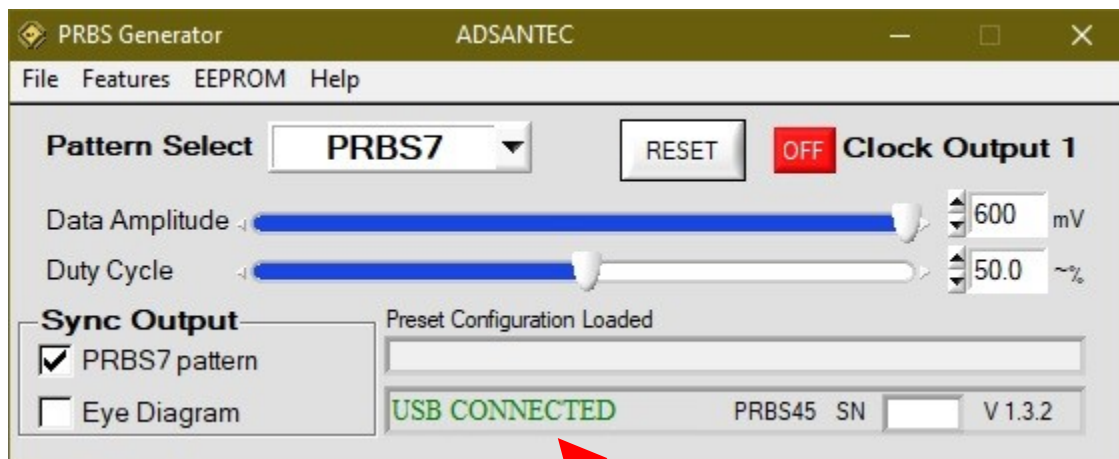
1.6 Left-click on *Restart* if requested.

1.7 Double left-click on the “CDM v2.10.00 WHQL Certified.exe” to install USB drivers. This file can be found in the same directory as the setup file used earlier to install software.



1.8 Double left-click on the icon *ASNT_PRBS45 Vx.x.x* on the desktop to open the control software.

1.9 Wait until the USB indicator has turned **green**. This indicates that the software is connected to the ASNT_PRBS45. If the **green** indicator does not appear, then the USB drivers may not be installed correctly or the USB cable may need to be re-inserted. Make sure that both LED's on the front panel are illuminated.





Front Panel Connections



Clock In

Apply a half-rate clock signal to the SMA female Clock In input on the front panel. The input amplitude must be greater than $100mV$, and less than $1V$ peak to peak. Example of the clock frequency needed: $16GHz$ for $32Gbps$, $8GHz$ for $16Gbps$, etc. The clock input is AC coupled inside the unit; therefore no external DC block is required.

Data Out

The Data Out differential output is AC coupled inside the unit; therefore no external DC block is required. Both connectors are female K-type 2.92mm. The data output amplitude and duty cycle are adjustable through the GUI. If using only a single-ended output, 50Ω terminate the unused output.

Clock Output

The Clock Out single-ended female SMA output is AC coupled inside the unit; therefore no external DC block is required. This clock output has lower jitter than Clock Out 1. If not using this output, 50Ω terminate.

Clock Output 1

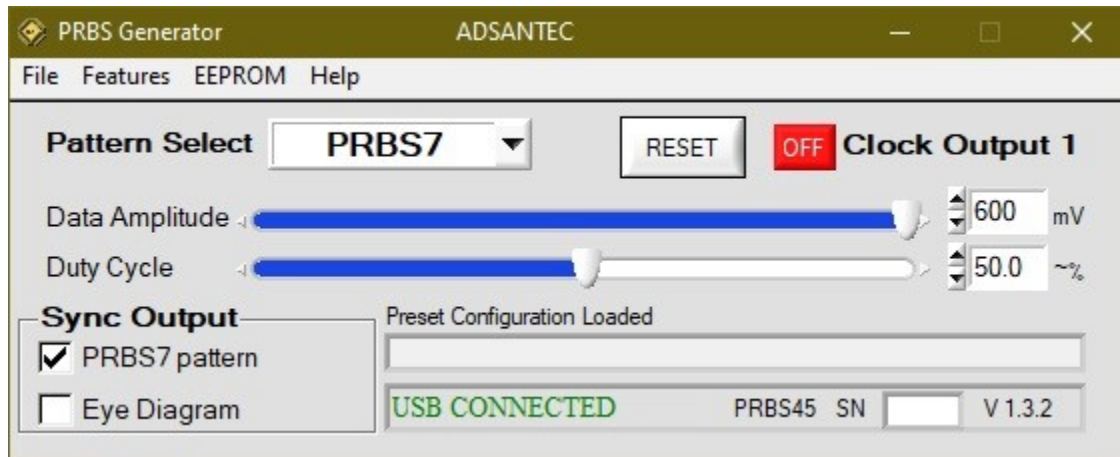
The Clock Out 1 differential output is AC coupled inside the unit; therefore no external DC block is required. Both connectors are female SMA. This clock can be turned on or off via the GUI. If using only a single-ended output, 50Ω terminate the unused output.

Sync

The Sync output is AC coupled inside the unit; therefore no external DC block is required. The connector is a female SMA connector. The Sync output is used to synchronize the data with an oscilloscope.



Operation



The GUI provides control of the unit. Left-click on the pull down box next to the label **Pattern Select** to choose the desired PRBS pattern or custom data. The output data amplitude is controlled through the slider labeled **Data Amplitude**. Moving the slider all the way to the left (0.000 in the numeric box) will turn off the data output. The duty cycle of the data output can be controlled through the slider labeled **Duty Cycle**.

If no output is observed, move the slider for Data Amplitude all the way to 0 (the left), then move up to the desired amplitude to unlock.

The On/Off button inside the box labeled for **Clock Output 1** turns it on and off. The single-ended **Clock Output** cannot be turned on or off.

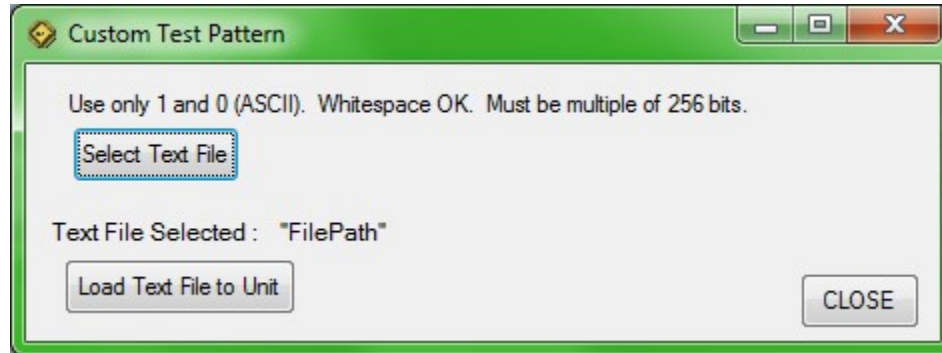
The GUI has three options to select from to control the **Sync Output**: **PRBS7 Pattern**, **Eye Diagram**, and **Manual Divide Ratio**. If **PRBS7 Pattern** box is selected, then the **Sync Output** will provide a divided by 127 clock to observe the PRBS7 pattern on an oscilloscope. **Eye Diagram** box provides a divided by 255 to observe an eye diagram on an oscilloscope. **Manual Divide Ratio** box allows any divide ratios from 1 to 256 (output is further divided by 4) to be typed into the box. The value is programmed after the **Set** button is clicked.

There is a small chance that the PRBS pattern will be incorrect. If so, left click on the **RESET** button on the GUI shown above to reset the unit.

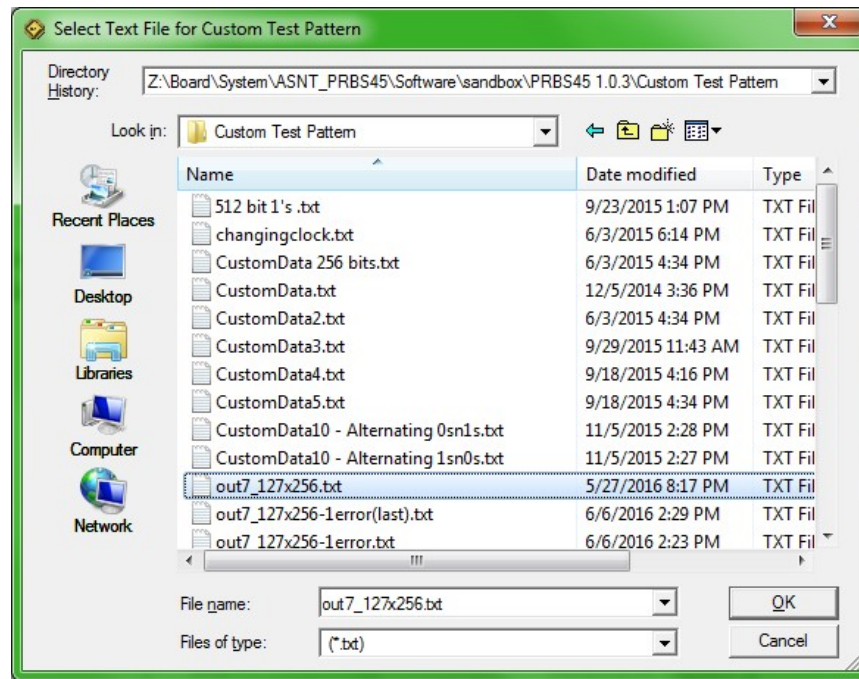


Custom Data

The Custom Data selection under Pattern Select allows up to 1 Mbit of binary test pattern data to be loaded into the unit. An ASCII text (.txt) file with 1's and 0's is required (other characters ignored). The length must be a multiple of 256 bits, up to 1,048,576 bits. To set up a custom test pattern, select Custom Data in the Pattern select drop down box. A window will pop up as shown below.



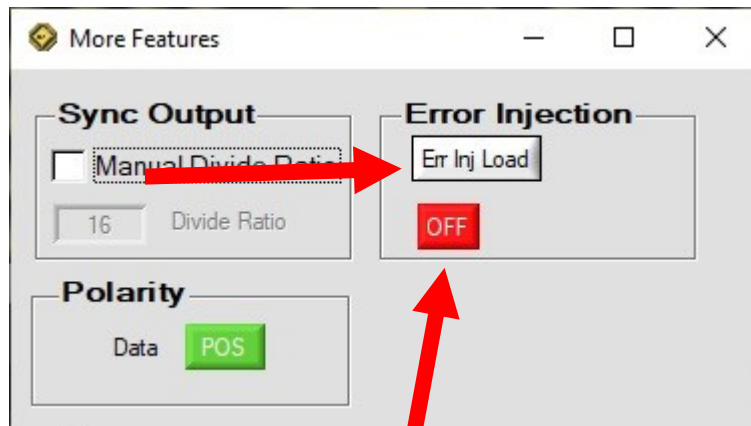
Select “Select Text File” button, and a select text file window will appear similar to what is shown below. Select .txt file and press “OK”. Next select “Load Text File to Unit”, and the text file will be loaded into the unit. If the unit is powered off, the custom data will have to be reloaded.





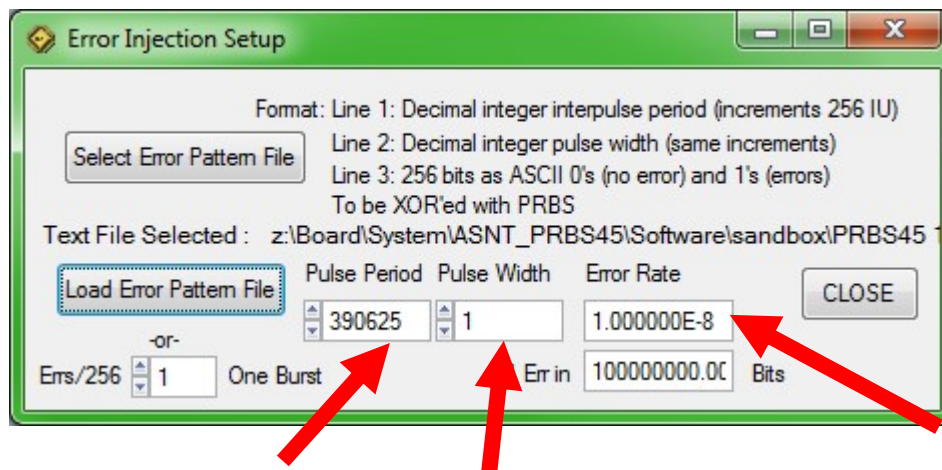
Error Injection/Output Polarity

To enable error injection, begin by clicking the “Features” tab within the top section of the Main GUI window. Error injection can be enabled here. Above the “Error Inj On” check box, the Error Injection Load can be specified by clicking it which opens the Error Injection Setup window.



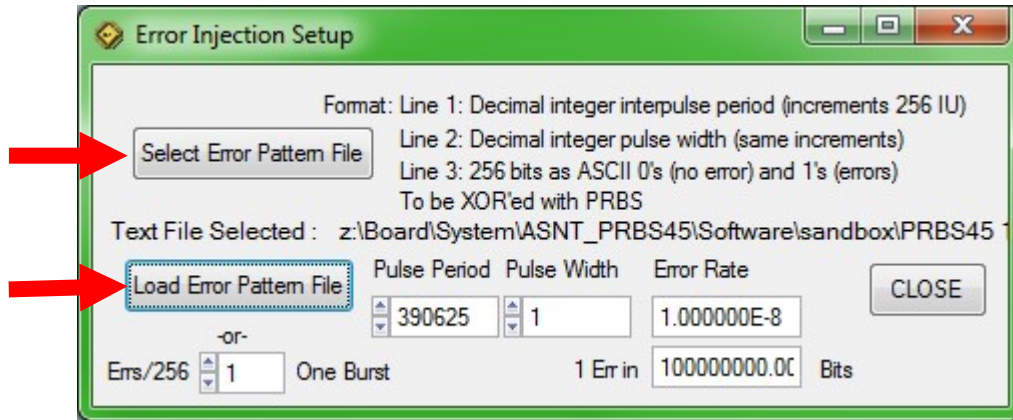
Through GUI

Here the Pulse period, Pulse width and error rate can be specified. It will provide a burst of errors based on the settings selected.



Through Text File Import

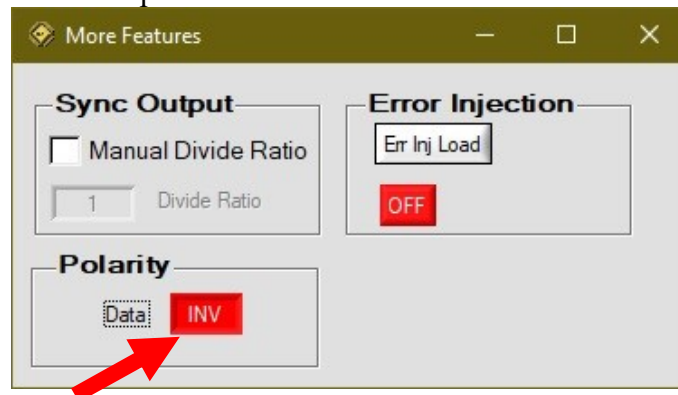


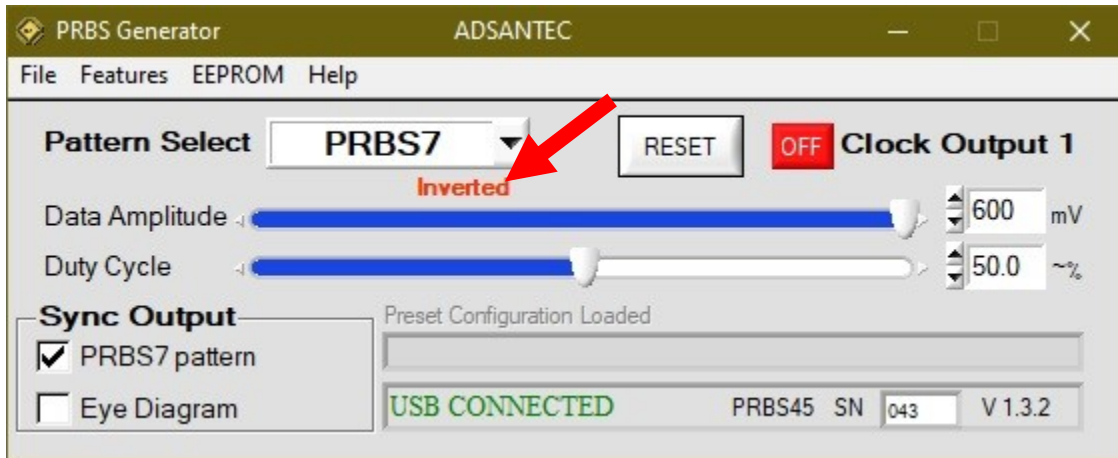


Error injection can also be achieved through the use of an error pattern file (text file), where the settings are configured as specified within the above window, with the first line specifying the pulse period, the second line specifying the pulse width, and the remainder of the file specifying the 256-bit error burst pattern. Zeroes (ASCII) denote no error and ones denote errors. When the text file is formatted and configured as desired, it can be selected through the “Select Error Pattern File” button, followed by “Load Error Pattern File”, which can give greater control and allow the location of errors to be specified within each 256 bits.

When error injection is enabled, the PRBS selection dropdown menu will change in color to warn that the pattern being generated is not exact.

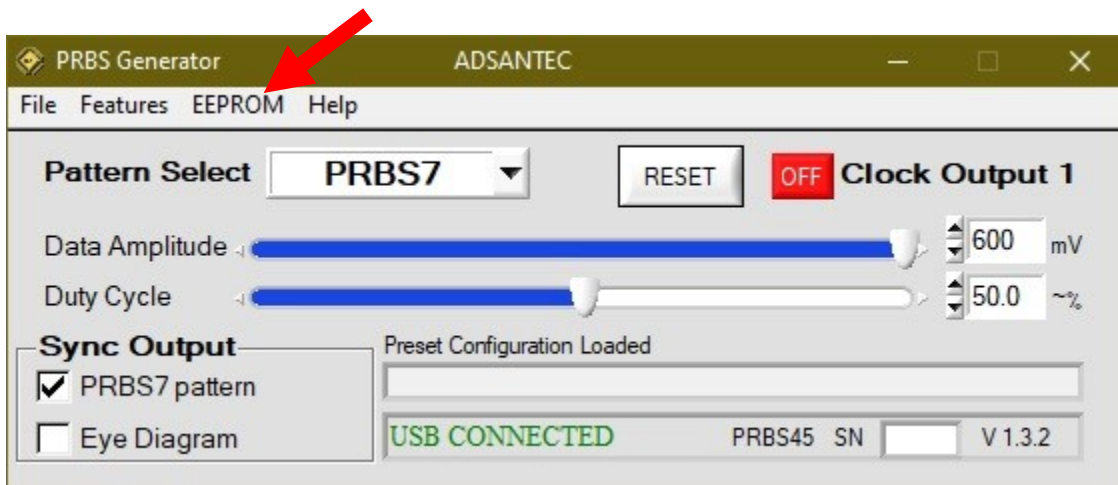
Note that under the Features menu, the option also exists to invert output polarity, which can sometimes be handy. By default, the output polarity is as specified on the unit’s front panel, but this can be altered by clicking the polarity square labeled POS within this menu. Note that the output GUI will change to reflect the output is inverted as shown below on the following page.





EEPROM Settings Function

While by default, the ASNTPRBS45 software will save settings to the installation directory inside a settings.cfg file, it may sometimes be advantageous to save (or load) settings directly to/from the unit itself; this can be particularly handy when switching the PC that is connected via USB. To make use of this function, select the EEPROM menu shown above to save/load settings as needed.





DLL

The 32-bit DLL is compiled from CVI Labwindows. Its functions are in C style arguments and they return integers. The DLL can be used as an alternative to the GUI. This is useful for using the unit to automate control. A python example is provided as an example on how to use the DLL. The DLL may be used by any program that can deal with C type functions. The DLL and python example file can be found in the installation directory of the GUI. **Note: 32-bit version of Python must be used.**

int usb_init(void)

Description: Finds USB device. Must be called first in program.

Argument: none

Returns: '1' USB is connected, '0' USB is not connected

int usb_close(void)

Description: Closes USB devices. Must be called to clear USB handle so that another program can control the unit.

Argument: none

Returns: '1' USB closed, '0' Error (handle may be closed already)

int set_divide_ratio(int divide_ratio)

Description: Set Sync Output divider from 4 to 1024. Output Sync divide ratio = divide_ratio * 4

Argument: integer 1 to 256

Returns: '1' write successful, '0' write unsuccessful

int ClockOutput1(int ONoff)

Description: Turn clock output 1 ON or OFF

Argument: integer 1 or 0

Returns: '1' write successful, '0' write unsuccessful

int set_dutycycle(float voltSetting)

Description: Adjust output data duty cycle

Argument: float 1.8 to 3.3

Returns: '1' write successful, '0' write unsuccessful

int set_amplitude(float voltSetting)

Description: Adjust output data amplitude

Argument: float 1.8 to 2.7

Returns: '1' write successful, '0' write unsuccessful



int prbs_select(int prbsSetting)

Description: Select output data PRBS pattern

Argument: integer

1 = PRBS7

2 = PRBS9

3 = PRBS11

4 = PRBS15

5 = PRBS17

6 = PRBS20

7 = PRBS23

8 = PRBS29

9 = PRBS31

10 = Half-Rate Clock

Returns: '1' write successful, '0' write unsuccessful

int RST_PRBS(void)

Description: Resets PRBS generator. Note: **This functions needs to be called whenever input clock frequency is changed. This includes turning on the unit or removing/applying a clock input.**

Argument: none

Returns: '1' write successful, '0' write unsuccessful



REVISION HISTORY

Revision	Date	Changes
1.4.2	11-2024	Updated for New Software Features
1.3.2	07-2019	Updated Letterhead
1.3.1	07-2016	Added DLL Feature
1.2.1	06-2016	Added Error Injection Feature
1.1.1	04-2016	Added unit Reset feature on GUI Rearranged document Added more pictures Clarified descriptions
1.0.1	10-2015	Initial Release