

# JSB383 API Overview

## Document Purpose

This document provides information for controlling the jsb383 Opto Input module from various programming languages using either the newer managed class library dll or the older standard dll for applications that do not support the managed class library. The sample source code directories contain samples for various languages that demonstrate use of each of the functions in a simple straight-forward manner and can be used as a starting point for user application. The samples applications show the exact calling sequence for each of the function described in this document including the full variable and function declarations. This text is meant to just give an overview of what is available. The user will need to have the programming language software installed on the computer being used, and should be familiar with that language operation and use. This document only contains material related to the unique module dll software and is not intended as a programming language tutorial. All examples on the CD have been built and tested for a given version of programming software. Updated versions of programming software may require slight modifications to code or option settings.

## Hardware Overview

The hardware has 4 or 8 opto-isolated inputs. Inputs are number 1 – 8. When reading the inputs a value of '1' means that voltage is present on the input greater then switch point (either 3.3V or 10V). Refer to the pin-out documents for exact locations.

## Class Library API

The class library is a set of functions that can be called from any programming language supporting calls to a managed class library. It provides the necessary calls to the modules usb driver to do the requested action on the module, providing simple user interface. The class library name is "jsb383CL.dll". Included with the class library is an .xml file that is used by Visual Studio to support IntelliSense help to make using the functions quick and easy. Both files can be found on the CD in the 'Class Library' directory. Local copies may also be in sample code directories to ease the transferring of the sample source to your working directories, as some Visual Studio version do not support relative paths. This Class Library is supported on Visual Studio 2005 and later.

### *Get the number of modules attached*

`NumberOfModules()`

This function returns the number of modules current attached to the system.

### *Get the serial number of a module*

`SerialNumber( )`

This function returns the serial number of for the requested module number (range 1-256). The serial number is used to access multiple modules on the system. When a system is powered up with multiple modules attached the units may come up in a different order, and serial numbers are used to find them.

### *Get the Class Library version*

`ClassLibraryVersion()`

This function returns the dll version being used for the given serial number.

### *Get the driver version*

`DriverVersion()`

This function returns the driver version being used for the given serial number.

### *Get the module firmware version*

`FirmwareVersion()`

This function returns the firmware version for the module.

### *Flash Led on module*

`FlashLed()`

This function flashes the module led.

### *Read the number of inputs on module*

`NumberOfInputs()`

This function returns the number of inputs on the module. Typically 4 or 8.

# JSB383 API Overview

## **Check state of input module**

`IsInputOn()`

This function checks the state of the requested input (value 1-8) and returns true if input is on (> switch point).

## **Read the inputs on module**

`Inputs()`

This function returns the value of all the inputs on the module as a byte. The lsb of the return byte is Input 1 and msb is Input 8. If the bit is a one, then that input has the voltage is greater then switch point, otherwise a 0 is returned.

## **DLL API**

The dll file is jsb383dll and contains a set of callable functions that can be called from any programming language supporting calls to a standard windows dll. It provides a high level interface between a user application and the provided module usb drivers. The files are located under the "DLL" directory on the CD and may also appear in sample code directories to ease build. Also included is the file jsb383Dll.h which has function prototypes for languages requiring declarations.

## **Get the number of modules attached**

`jsb383GetNumberOfModules()`

This function returns the number of modules current attached to the system.

## **Get the serial number of a module**

`jsb383GetSerialNumber()`

This function returns the serial number of for the requested module number (range 1-256). The serial number is used to access multiple modules on the system. When a system is powered up with multiple modules attached the units may come up in a different order, and serial numbers are used to find them.

## **Get the DLL version**

`Jsb383DllVersion()`

This function returns the dll version being used for the given serial number. `Djsb23GetDllVersion`

## **Get the driver version**

`Jsb383DriverVersion()`

This function returns the driver version being used for the given serial number.

## **Get the module firmware version**

`jsb383FirmwareVersion()`

This function returns the firmware version for the module.

## **Number of Inputs**

`Jsb383MaxInputs()`

This function returns the number of inputs on module. Typically 4 or 8.

## **Check state of input module**

`Jsb383IsInputOn()`

This function checks the state of the requested input (value 1-8) and returns true if input is on (> switch point).

## **Read the inputs on module**

`Jsb383Input()`

This function returns the value of all the inputs on the module as a byte. The lsb of the return byte is Input 1 and msb is Input 8. If the bit is a one, then that input has the voltage is greater then switch point, otherwise a 0 is returned.

## **Flash Led on module**

`Jsb383FlashLed(serial number, port number)`

This function flashes the module led.