

Dr. Gal Ben-David
Digital Signal processing consultant



ALON1

USB automation cards

User's manual

© Copyright 2005 by Dr. Gal Ben-David

Dr. Gal Ben-David
POB 82, Adi 17940, Israel
www.galium.co.il
gal@galium.co.il
Tel: +972-54-4698289
Fax: +972-4-9860401

All rights reserved. No part of this publication and its accompanied hardware and software may be reproduced, stored in a retrieval system, or transmitted by any means, electronic, mechanical, photocopying, recording or otherwise, without prior written permission from *Dr. Gal Ben-David*

Information provided is believed to be accurate and reliable. However, no responsibility is assumed for its uses, nor for any infringements of patents or other rights of third parties which may result from its use.

Hardware and software are sold “as is”. We assume no responsibility for direct or indirect damages, including equipment fault, computer malfunction, loss of data, and loss of income.

We, at *Dr. Gal Ben-David*, hold a policy of continuing improvement of our products. We reserve the right to change and improve any of our products without prior notice. Please check our web site for last software versions.

.NET, Visual C, MFC, Visual Basic and Windows are trademarks of Microsoft Corporation

USB is a trademark of the USB Implementers Forum, Inc.

NAIS is a trademark of Matsushita Electric Works, Ltd., Osaka, Japan.

Table of content

Table of content	3
Revision History	3
Acknowledgment	4
The ALON1 family of USB interface boards	5
System requirements	6
PC as a controller	7
The Universal Serial Bus	8
Power considerations	10
Using the 5V connection at the ALON1 boards	12
Response timing	13
USB Compliance	14
Installation	15
Software programming - SDK	19
Initializing ALON1 board system	20
Acquiring a list of ALON1 boards connected to the computer	21
Input/Output	22

Revision History

Date	Details
17-MAR-2005	Original

Acknowledgment

Dear valued customer,

I would like to thank you for purchasing an ALON1 USB automation card. The card was designed to allow simple and efficient control for various real world applications with the ease and expendability of USB interface. I believe that you will find our boards to be reliable and easy to use.

The most important person in our company is our customer. Our products are as useful as what we know about customer needs, what his/her expectations are and how our products will better serve him/her. I will appreciate having any suggestion, feedback and ideas that will make your designs better. We are a small and agile company, and we fortunate to love what we do. You will be surprised by the short time that we take good idea into good products.

Regards

Dr. Gal Ben-David

gal@galium.co.il

The ALON1 family of USB interface boards

The ALON1 is a family of low-cost USB automation and control electronic boards. The boards enable the design of extendible, easy to program, simple and low cost automation systems.

A typical ALON1 system is shown in the figure 1.

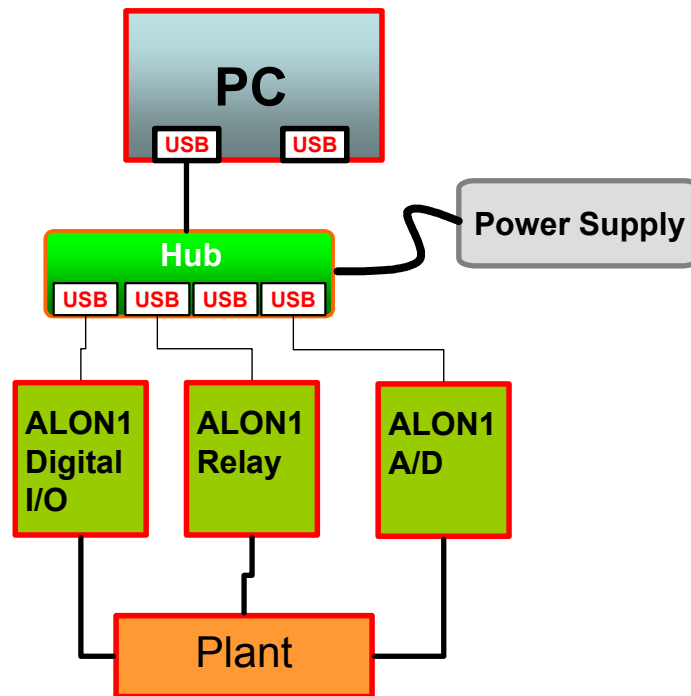


Figure 1 – A typical ALON1 system

The term *plant* represents a sub-system that requires some degree of automation. Typical applications are: Automatic test equipment, laboratory test setup, Industrial shop floor, etc. The plant is controlled by activating switches, relays, motors, heating elements, etc. The status of the plant is observed by examining various mechanical switches, photo-switches, sensors, etc.

ALON1 boards are capable of observing the system status via digital and analog inputs, while controlling the plant using relays, digital output and analog outputs.

All ALON1 boards connect to a Personal Computer via a USB interface. Up to 127 USB devices may be connected and work simultaneously, using self powered hubs or direct computer connection.

Custom PC software should be written by the user based on the free ALON1 SDK (Software Developer Kit). The SDK is based on a general Dynamic Link Library (DLL) that interfaces with all boards. The software may be written in any high level Windows programming language (e.g., Visual C and Visual Basic). We are improving our software on a regular basis. Please visit our site, www.galium.co.il, for the latest software version.

System requirements

The ALON1 boards require the following configuration

Pentium III, 700MHz

128 MByte RAM

10 MByte of free disk space

Windows 98 (Windows 2000/XP highly recommended)

USB 1.1 or 2.0 port

Self powered hub is highly recommended

PC as a controller

The personal computer running Windows has many advantages over the traditional embedded controller:

1. Large memory space, both volatile (RAM) and non volatile (Disk)
2. High processing power
3. Good graphical capabilities
4. Variety of off-the-shelf peripherals
5. Easy data analysis and algorithm development
6. Easy report printing
7. General programming tool

These advantages encourage the use of a PC as controlling equipment. Nevertheless, the commercial Windows operating system was not design to be “mission critical”. Windows systems, especially those released before Windows 2000 are not “bullet proof” and sometimes the computer just hangs. This limitation, together with common sense and safety awareness should always be taken into consideration.

The Universal Serial Bus

The Universal Serial Bus (USB) connection was defined in the mid 90s by a group of leading companies that saw the need for a unified, fast, simple and low cost computer connection.

The USB standard, version 2.00, presents three possible connection speeds:

1. Low speed – 1.5 Mbps
2. Full Speed – 12 Mbps
3. High speed – 480 Mbps

USB offers high simplicity to the user; cables and connectors are the same for all devices and are nonreversible. USB devices may be attached and detached at any time (“hot insertion”).

USB peripheral are self-identifying (Plug and Play) and do not need any manual allocation of resources, such as memory addresses, interrupt vectors and DMA channels. The process of automatic identifying and resource allocation is called *enumeration*.

As a result of its advantages, the USB protocol replaces older communication standards in the vicinity of the personal computer. Some modern computers, for example, do not offer legacy RS232 ports.

The USB cable has four wires, two provide power and two provide a bidirectional differential signaling pair (in fact - differential only most of the time). Cables are polarized by using different connectors at either end. The connector at the PC side is called “type A”, while the device uses a “type B” connector (device end may be made captive). Hubs can be used to expand the number of available connections to a host. Devices can be attached and configured automatically.

A block diagram of the interface of a USB device is show in figure 2.

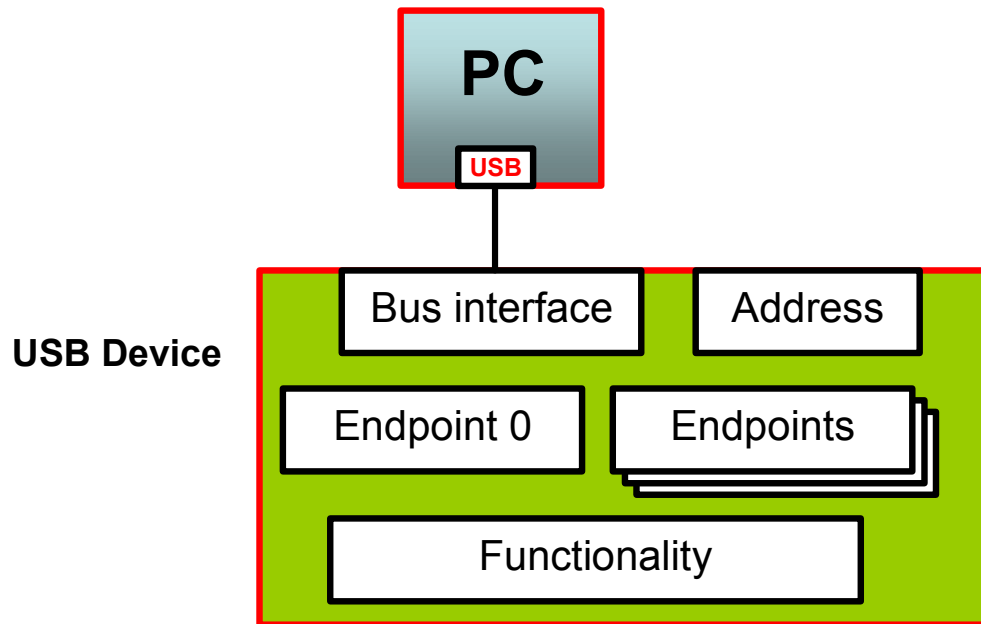


Figure 2 – A USB device

The USB is a master-slave protocol, where the PC is always the master. All transactions are initiated by the PC. When information is sent from the PC to the device, the transaction is called OUT transaction. IN transaction refers to information flow from the device to the PC.

Each USB device has an 7 bit address register. The register is zeroed on reset (or attachment) and an address is allocated during enumeration. A USB device can obtain one of 127 device addresses on the shared USB bus. Each device has one or more 'endpoints' (addressable FIFOs), which can be addressed from the host for read or write. A special endpoint (endpoint-0) is used during enumeration.

Power considerations

USB devices may be either self powered (powered by own power supply) or bus powered (using the host or hub power supply). The ALON1 boards were designed as bus powered, in order to eliminate the need for a wall brick power supply for each board.

Each USB port may supply up to 500mA at 4.75V to 5.25V. All ALON1 boards consume less than 500mA. When enumerated, the boards declare their power consumption as somewhat higher than the real consumption. This allows the user to have a limited source of power. Check board specification for more details.

The boards may be connected directly to the PC as shown in the figure 3.

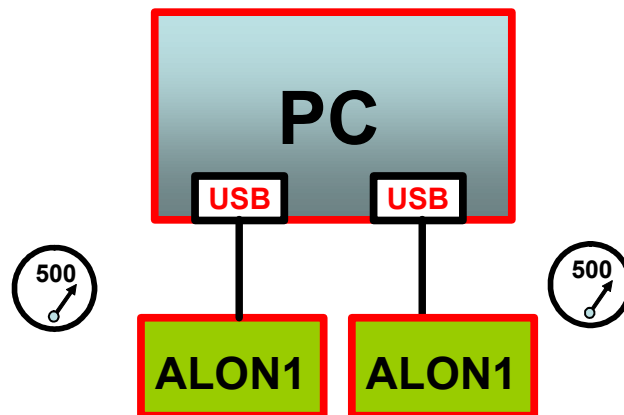


Figure 3 - Option A Connecting directly to PC

A better option would be connecting the devices via a self powered hub as shown in the figure 4. Using the hub allows more devices to be attached, while adding the hub as a mean of PC protection.

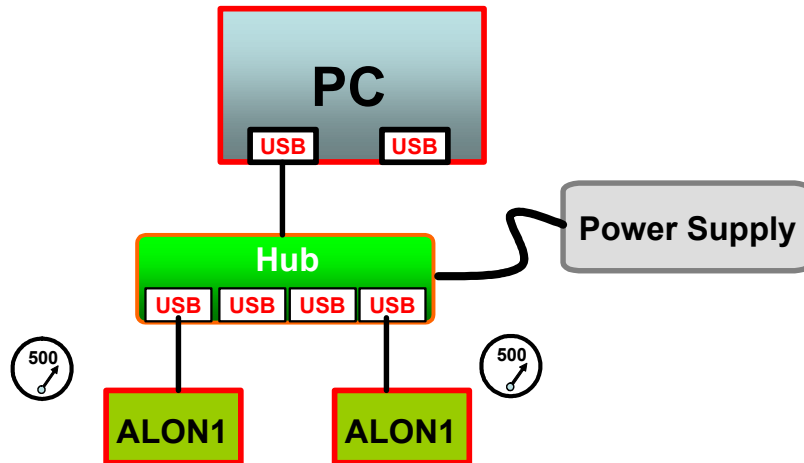


Figure 4 - Option B (recommended) – Using self powered hub

Using bus powered hubs should be avoided. Since each USB port may supply up to 500mA, each hub connection may deliver 100mA that is not enough for the ALON1 boards.

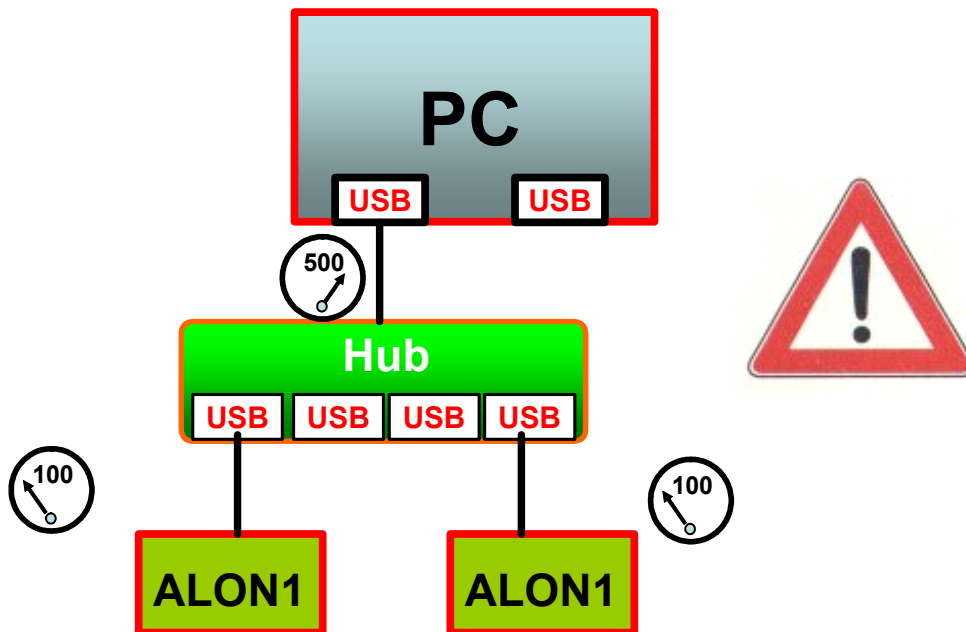


Figure 5 - Option C (do not!!!) – Using bus powered hub

Using the 5V connection at the ALON1 boards

ALON1 boards have a 5V connection for custom use. Since USB device may use 500mA, the user should check board specification for board own use and for available power.

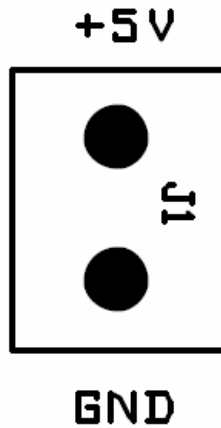


Figure 6 - 5V connector for custom use

Caution: The 5V supply is taken directly from the PC/Hub power supply. Care must be taken in order not to consume more power than a total of 500mA per card, including card's own function. High currents and short circuits may damage the PC/Hub. Do not make connections when the board is connected to the PC and PC is powered.

Response timing

The USB protocol is based on packets sent between the PC host and the device. Packets sent by the device and read by the PC are called IN packets, while packets sent by the PC to the device are OUT packets. At a recurring interval, the host sends an OUT packet and/or asks for an IN packet. Note that the device cannot send packet unless requested to. IN and OUT packets sent at recurring intervals are obtained using USB packet scheme called INTERRUPT packets. The term interrupt may be misleading, as the communication idea is closer to polling. USB low speed devices (1.5M bit per second) are polled every 10mS.

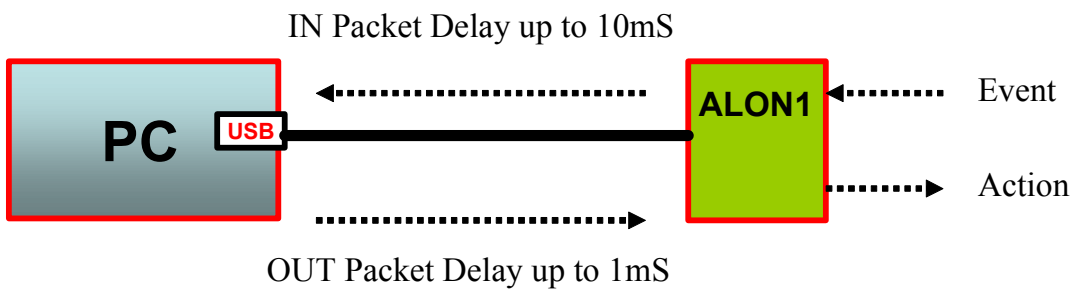


Figure 7 - ALON1 – response timing

System design should take this interval into consideration. Change in the state of the system will be recognized by the host software after up to 10ms. Command sent to the device will be received and acted upon may also be delayed up to 1mS.

USB Compliance

The ALON1 family of USB controllers was design to meet both USB1.1 and USB2.0 specifications at low speed (1.5M bit per second). We tested the boards in a variety of systems and combination together with additional ALON1 boards and various other USB devices. Tests were done under Windows 98, 2000 and XP.

Nevertheless, two features required by the USB specifications are not met as we believe are not applicable to automation applications

Captive cable – Low speed devices must not have a USB type B connector. The cable should be permanently attached to the device. Extension cables are also forbidden by the USB specifications. The ALON1 boards make use of a connector in order to allow various lengths of cables without extensions. Cable length are limited to 5 meters.

Power down – USB specifies that a device must enter suspend state after 3mS of no USB activity. When suspended, the device must consume less than 500 microampere. The ALON1 board would have to power down LEDs, relays etc. We believe that control applications are better freeze in case of trouble, rather than disconnect everything.

Installation

All ALON1 boards use the standard USB Human Interface Device (HID) interface. Windows recognizes the boards automatically and no special driver is needed.

Windows 98/SE/ME

After the first attachment of the board to the computer, a Add New hardware wizard will be opened automatically.



Figure 8 – Window 98 Add hardware wizard dialog

.Note that the board was identified as Human Interface Device. Press Next.

Allow windows to look for the driver



Figure 9 – Window 98 Search for driver dialog

Press Next. Continue following the instructions (two more dialogs, press Next), and at the end of the installation process Windows identifies the device.



Figure 10 – Windows 98, Installation done

Press finish.

In order to verify proper installation, right click the “My Computer” icon and select properties. A dialog will be opened. Now, select hardware and the device manager.

The board will appear as a Human Interface Device

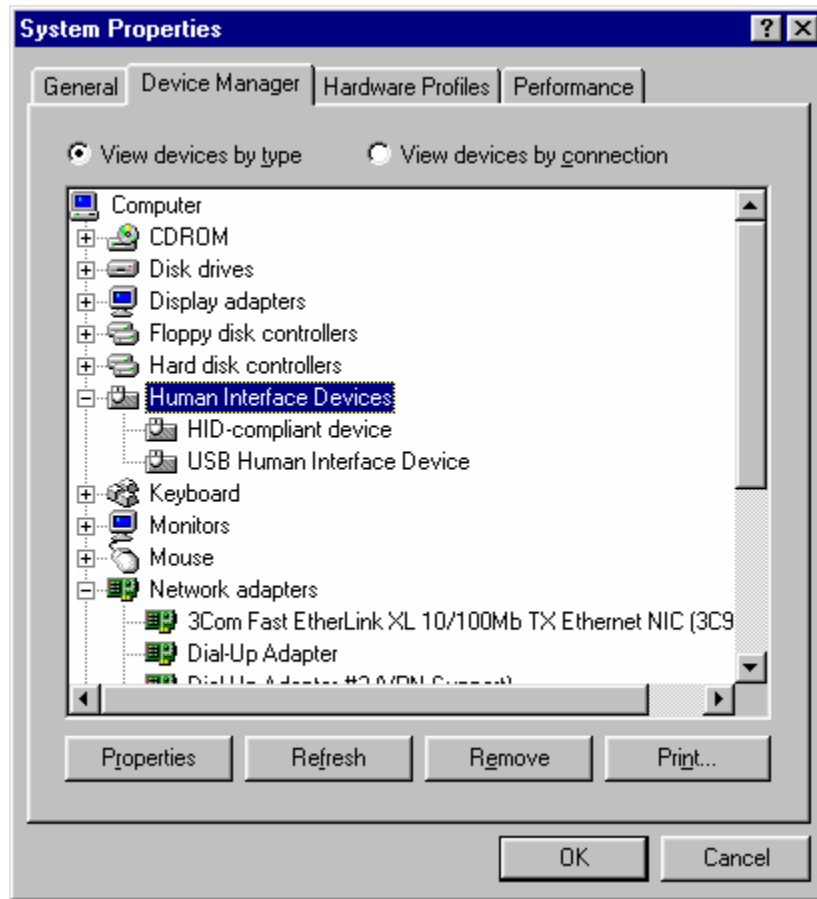


Figure 11 – Window 98 device manager

In order to install the SDK, copy the directory Alon1USBSDK from the CD/internet download to your hard disk.

Windows 2000/XP – Installation is made totally automatic by the operating system. Connect the device to the PC and let Windows do the installation for you. In order to verify proper installation, right click the “My Computer” icon and select properties. A dialog will be opened. Now, select hardware and the device manager. The board will appear as a Human Interface Device

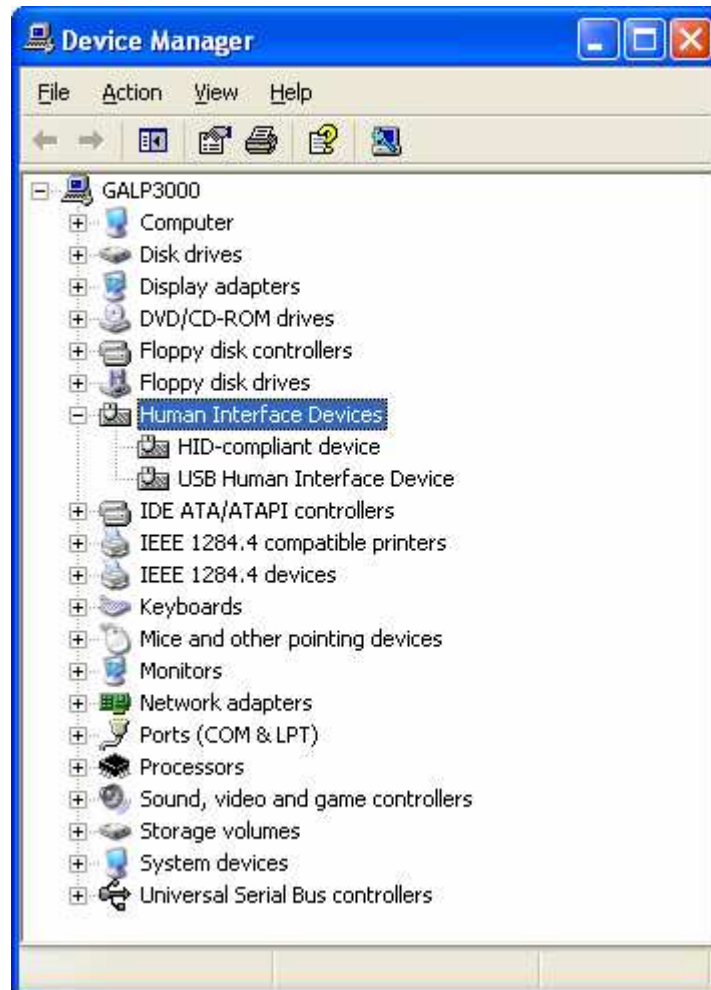


Figure 12 – Window XP device manager

In order to install the SDK, copy the directory AlonUSBSDK from the CD/internet download to your hard disk.

Software programming - SDK

As already mentioned, the ALON1 SDK (Software Developer Kit) enables the user to implement automation software by means of standard Windows programming. The SDK uses a single Dynamic Link Library (DLL) that interfaces with all boards. The software may be written in any high level Windows programming language. The SDK includes examples in Visual C and Visual Basic.

We are adding new boards and improving our software on a regular basis. Please visit our site, www.galium.co.il, for the latest software version.

The SDK includes interface files as well as examples in Visual C and Visual Basic. The interface files are:

1. Alon1USB.dll – Software library file that interfaces with the ALON1 boards. The DLL should be copied to your application directory. It is also possible to place the DLL at the Windows\System directory.
2. Alon1USB.lib – Static linked library file that is linked to C programs and allows access to the SDK functions. Visual C programmers should add this file to the linker list of files.
3. Alon1USB.h – C header file that contains all declarations to use the .lib functions. Include this file into any file that uses the ALON1 functions.

The documentation is written for C programmers. Visual Basic programmer will use the same functions, using VB ability to call DLL functions. Check with software examples.

Initializing ALON1 board system

The ALON1 board system is initialized by calling the **InitALON1** function:

```
void ALON1_Init(ALON1CALLBACKFUNC ALON1CallbackProc);
```

The function should be called prior to any call to the ALON1 DLL.

The parameter **ALON1CallbackProc** is a pointer to a function in user's code that is called by the DLL during events.

```
typedef int (CALLBACK* ALON1CALLBACKFUNC)  
(CONST ALON1BOARDSTRUCT *, CONST VOID *);
```

More detailed on this function will be discussed later.

Before quitting the program, the **CloseALON1** function should be called.

```
void ALON1_Close(void);
```

Acquiring a list of ALON1 boards connected to the computer

USB devices are hot pluggable, thus may be connected and disconnected at any time. The ALON1 DLL should therefore keep an inventory of current ALON1 devices that are attached to the PC. This inventory may be changed due to insertion and removal of devices.

Software should be notified by events of new devices that are being attached and by devices that were removed.

Each USB device is identified by a unique combination of 16 bit Vendor ID (Purchased from the USB implementation forum), 16 bit Product ID and a 32 bit unique serial number, that is also written on the board with a sticker.

All ALON1 boards share the *Dr. Gal Ben-David's* vendor ID = 0x16E9.

Products IDs are defined in Alon1USB.h. For example

```
#define PRODUCT_ID_RL4OP4 11 // Relay cards product ID
```

A list of ALON1 board is maintained by the DLL. The list is a simple array of up to 127 elements. Each element is a structure keeping the board Product ID, Serial Number and a handle that is used internally. Structure should not be changed by user's software. Important: Structures should not be used after a device is notified to be detached.

```
typedef struct  
{  
    unsigned short ProductID;  
    // Common to all cards of the same type  
    unsigned long SerialNumber;  
    // Unique number programmed during manufacturing  
    HANDLE DeviceHandle;  
    // For DLL internal use only  
} ALON1BOARDSTRUCT;
```

At any time the software may obtain the number of active ALON1 boards by calling:

```
int ALON1_GetNumberOfActiveBoards(void);
```

Program may go over all boards by calling

```
int ALON1_GetBoardStructure(  
int BoardIndex, // Range: 1 to Number of active boards  
ALON1BOARDSTRUCT *); // Output structure  
// returns zero if OK, nonzero on error
```

In order to search for a specific board, using its Product ID and Serial number

```
int ALON1_GetBoardStructure(  
unsigned short ProductID; // input Product ID  
unsigned long SerialNumber; // Input S/N  
ALON1BOARDSTRUCT *); // Output structure  
// returns zero if OK, nonzero on error
```

Input/Output

Communicating with boards is device dependant. Please check with each board specifications.