USBrelayDeviceNET

version 1.24.02.22

# Summary

A C# library that provides methods to control HID compliant USB Relay Devices. The library uses native function calls to standard Windows APIs for communicating with the USB Relay devices.

# Motivation

The standard library provided by the vendor of the USB Relay Devices (usb\_relay\_device.dll) is an unmanaged library and for using in C# applications requires a wrapper be written. It has been found that the standard library has limitations (does not provide a method of changing device id/serial number) and does not function as expected (with multiple devices connected to a system, will only find two devices and does not properly report the device types if different).

In order to qualify the USB Relay Devices to be used in new equipment designs, control of the source code is preferred and if to use third-party libraries they should be bug free and provide all desired functionality, the standard library did not meet that basic criterion.

# Technical Details

Language: C#

.NET Framework: 4.5

Supported Platforms: x86, x64 and AnyCPU

External Library: None

Native API: hid.dll, kernel32.dll and setupapi.dll

# Features

# API reference

## Constructors

|  |  |
| --- | --- |
| USBrelayDevice() | Initializes a new instance of USBrelayDevice class |
| USBrelayDevice(bool) | Initializes a new instance of USBrelayDevice class and when parameter is set true, opens the first USB Relay Device found. |
| USBrelayDevice(string) | Initializes a new instance of USBrelayDevice class and opens a USB Relay device matching the passed string (Device ID/serial number). |

## Properties

|  |  |
| --- | --- |
| DeviceInfo | Structure contains USB Relay device information, this structure is returned in a list using the GetDevices() method. Members:   * device\_ID - Device ID/serial number * device\_path - System path to device * index - List index value * product\_name - Device description * relay\_Count - Number of relay channels |
| DeviceOpen | True when a USB Relay device is connected (Get only) |
| DevicesFound | True when USB Relay devices are found and available for connection |
| IsDisposed | Tue when class has been disposed |

## Methods

|  |  |
| --- | --- |
| ALLRelaysOFF() | Turns off all relays, returns true if successful |
| ALLRelaysON() | Turns on all relays, returns true if successful |
| CloseDevice() | Closes connected USB Relay device, returns true if successful |
| Dispose() | Releases all resources used by USBrelayDevice. |
| GetDevices(bool) | Finds all USB Relay Devices attached to the system and returns a list containing device information structures for each of the devices. |
| GetLibraryInfo() | Gets the name and version number of library, returns two element string array |
| GetRelayStatus() | Get the current status of all relays, returns 8 element bool array |
| OpenDevice(Int32) | Opens a USB Relay Device using list index, returns true if successful |
| OpenDevice(string) | Opens a USB Relay Device using device id/serial number, returns true if successful |
| RelayOFF(Int32) | Turns off a single relay, returns true if successful |
| RelayON(Int32) | Turns on a single relay, returns true if successful |
| SetDeviceID(string) | Sets the Device ID/serial number, returns true if successful |

## Events

|  |  |
| --- | --- |
| USBrelayError | USBrelayDevice class Error Event |

## Remarks

Only one USB Relay device can be opened and controlled with each instance of the class, to open multiple devices connected to a system, a new instance of the class must be created for each device.

# Example

This is where you try to compress your project and make the reader understand what it does as simple as possible. This should help the reader understand if your code solves their issue.

# Installation

If your project needs the installation of certain software or configurations to the system. Do mention it in this section as it helps a lot for the reader to use your project. The steps mentioned should be precise and explanatory. If possible, you can add links that can help them better understand how to configure/install the necessary files or software.

# Source Code

Notes

HID Set Feature Commands (9 element byte array passed in the lpReportBuffer parameter)

|  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| Index | 0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 |
| Value | 0x00 | CMD | Data [0] | Data [1] | Data [2] | Data [3] | Data [4] | 0x00 | 0x00 |

* Set Device ID: CMD = 0xFA, Data 0 = ID[0], Data 1 = ID[1], Data 2 = ID[2], Data 3 = ID[3], Data 4 = ID[4]

ID[n] represents the 5 characters from a Device ID/serial number string , the characters are converted to a 5-element byte array.

* All Relays OFF: CMD = 0xFC, Data 0 = 0x00, all remaining bytes = 0x00
* Single Relay Off: CMD = 0xFD, Data 0 = Relay Number, all remaining bytes = 0x00
* All Relays ON: CMD = 0xFE, Data 0 = 0x00, all remaining bytes = 0x00
* Single Relay On: CMD = 0xFF, Data 0 = Relay Number, all remaining bytes = 0x00

HID Get Feature Details (9 element byte array returned in the lpReportBuffer parameter)

|  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| Index | 0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 |
| value | 0x00 | ID[0] | ID[1] | ID[2] | ID[3] | ID[4] | 0x00 | 0x00 | Relay status |

The returned byte array contains the Device ID/serial number and the relay status.

ID[n] is a 5-element byte array that is then converted to a string value.

The relay status is a byte representation of a bit array having the dimensions of 8 bits, i.e. value of 0xFF = 11111111, which for a USB Relay device having 8 relays would indicate that all relays were on.

# Credits

# Written by: David Neumeier, Email: [david.neumeier@adv-chem.com](mailto:david.neumeier@adv-chem.com)

Copyright © 2024, Advanced Chemical Solutions LLC, All rights reserved

# License

Redistribution and use in source and binary forms, with or without modification, are permitted provided that the following conditions are met:

Redistributions of source code must retain the above copyright notice, this list of conditions and the following disclaimer. Redistributions in binary form must reproduce the above copyright notice, this list of conditions and the following disclaimer in the documentation and/or other materials provided with the distribution.

# Legal

THIS CODE AND INFORMATION IS PROVIDED "AS IS" WITHOUT WARRANTY OF ANY KIND, EITHER EXPRESSED OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE IMPLIED WARRANTIES OF MERCHANTABILITY AND/OR FITNESS FOR A PARTICULAR PURPOSE. IT CAN BE DISTRIBUTED FREE OF CHARGE AS LONG AS THIS HEADER REMAINS UNCHANGED.