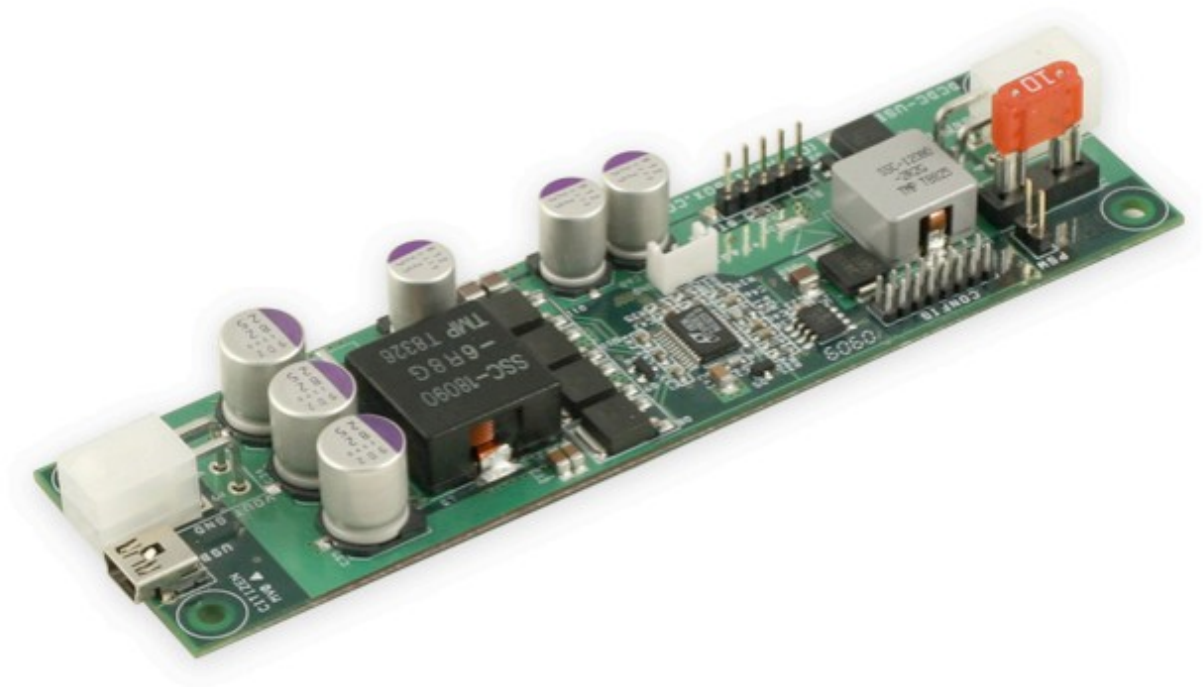


Mini-Box.com



DCDC-USB

API Manual
v1.0

1. Introduction

This manual is provided together with the DCDCUsb API and the C++, C# , Visual Basic example codes. Read first before start using the API.

The API is compiled in a dll (*DCDCUsbLib.dll*) using Visual Studio 2005. The manifest is included in the dll since C# and Visual Basic cannot handle dynamic dll loading otherwise.

To run the dll on any PC it does need four redistributable files from Visual Studio 2005 redistribution pack. Therefore it is recommended to have those four files in the same directory with the dll: Microsoft.VC80.CRT.manifest, msvcm80.dll, msvcp80.dll and msucr80.dll. They are found in the package together with the API dll. Another solution is to install the Visual Studio 2005 redistribution pack on the PC. This package can be downloaded from here: <http://www.microsoft.com/en-us/download/details.aspx?id=3387>

All the example projects were compiled with Visual Studio 2005 but updating it to a newer versions of Visual Studio should not be a problem. Just be sure You select x86 processor type (even if the project is compiled on a 64 bit OS) since the API is x86 compiled.

Basic C++, C# and Visual Basic knowledge is required to use and understand the API and the provided example code. Additional to Visual Studio any other compiler should be fine to use which can load a Windows dll and call the functions inside the dll.

2. The API

The API exports a set of cdecl C functions.

See the difference between cdecl and stdcall, fastcall declared functions here: [http://msdn.microsoft.com/en-us/library/system.runtime.interopservices.callingconvention\(v=VS.71\).aspx](http://msdn.microsoft.com/en-us/library/system.runtime.interopservices.callingconvention(v=VS.71).aspx) or simply search after "cdecl vs stdcall" on any search engine.

DCDCUsbLib.h provided with the API has a complete description of the functions exported in the API and how they should be used. All functions which are working with strings are using multibyte version of strings (not unicode!) therefore use string formatting routines to read and write those strings in multibyte.

There are more set of functions:

1. Open and close the connection (*dcdcOpenDevice* and *dcdcCloseDevice*)

IMPORTANT: after unsuccessful *dcdcOpenDevice* if there is no *dcdcCloseDevice* call the API will wait for the first DCDCUsb plugged in to the USB ports and will connect to it! Connection status can be requested with *dcdcGetConnected()*.

2. Get state variables (all of them are also displayed in the Status tab of the Windows software). Those functions are usually starting with *dcdcGet* and after that is the variable name (*dcdcGetMode* etc.)
3. Set state variables (*dcdcSetEnabledAuxVOut*, *dcdcSetEnabledPowerSwitch*, *dcdcSetEnabledOutput*, *dcdcIncDecVOutVolatile*, *dcdcSetVOutVolatile*)

4. Flash functions to read and write the DCDCUsb preset variable values (*dcdcLoadFlashValues*, *dcdcGetLoadState*, *dcdcGetMaxVariableCnt*, *dcdcGetVariableData*, *dcdcSetVariableData*, *dcdcSaveFlashValues*). With this functions anyone can read and write the variables set from the Settings tab of the Windows software. Always load at least once the whole flash before trying to modify anything – the flash variables are loaded and saved in one piece with *dcdcLoadFlashValues* and *dcdcSaveFlashValues*.

IMPORTANT: be sure the flash write functions (*dcdcSaveFlashValues*) are used in rare cases. Do not build functionality which writes voltages or any other values frequently since the flash in the DCDCUsb can be written only 10000 times according to the chip manufacturer so it shouldn't be overused.

3. Example projects

All projects were built with Visual Studio 2005 but converting them to newer versions of Visual Studio should not be a problem.

3.1. C++ project (DCDCUsbLibTest)

Uses the common way of importing a C dll using his library, h file and the dll itself. All examples necessary are provided in the project and C++ test code.

3.2. Visual Basic (VB) project (VBLibTest)

Uses the common way of importing of a C dll from Visual Basic. All function prototypes are provided in the example code together with VB test code.

3.3. C# Project (CSLibTest)

Uses the common way of importing of a C dll from C#. All function prototypes are provided in the example code together with C# test code.