kickd_pic32_ubl.dll

Version v0.2.0

2/28/2014 2:19:00 PM

Oliver Heggelbacher / www.kickdrive.de

Table of Contents

kickd_pic32_ubl

Updating firmware on Microchip PIC32 devices from your own Windows application.

kickd_pic32_ubl.dll is a simple Windows 32 bit DLL API that offers the functionality of the Microchip PIC32UBL.exe Windows PC application. No additional dependencies.

How to Use:

- Alternative 1: Use the kickd_pic32_ubl.h DLL include header and the kickd_pic32_ubl.lib file for implicit / static linking of the kickd_pic32_ubl.dll.

- Alternative 2: Use the Pic32UblRtLink C++ class for explicit linking at runtime. I.e. the DLL is only loaded on use, and not required at startup of your application.

See also Pic32UblRtLink - How to use for more information and example code.

See the Microchip Application Note AN1388 and the related source code ("PC Application" folder) for copyright and additional information:

http://ww1.microchip.com/downloads/en/AppNotes/AN1388%20Source%20Code%202014_02_14.zip

For the modifications and DLL wrapper around the Microchip PIC32UBL code, the following applies:

Author: Oliver Heggelbacher.

kickd_pic32_ubl.dll and the related example have been compiled using Visual Studio 2008, but there is no need to recompile kickd_pic32_ubl.dll yourself. kickd_pic32_ubl.dll can be used "as is" - It is linked statically and does not require additional runtime DLLs.

$Id: kickd_pic32_ubl_doc.h 339 2014-02-28 13:00:04Z oliver $

Pic32UblRtLink - How to use

The Pic32UblRtLink adapter class implements explicit linking of the kickd_pic32_ubl.dll. This is to avoid external references to the DLL during compile time and allows starting your application even when the kickd_pic32_ubl.dll is not available.

How to use:

- Make sure the PIC_DLL_PATH define in Pic32UblRtLink.h contains the correct path to the DLL.
- Create a Pic32UblRtLink object, and get your downloader started, e.g using something like this:

```
Pic32UblRtLink *pUpdater;
pUpdater = new Pic32UblRtLink();   // (with pUpdater a Pic32UblRtLink * pointer)
bool success = (pUpdater->getDLLStatus() == 0);
if (!success) {
    // indicate error
}
else {
    success = pUpdater->ConnectAndProgram("myfile.hex", 0x4d8, 0x3c);
    if (!success) {
        // indicate error
    }
}
if (!success) {
    delete pUpdater;
    pUpdater = NULL;
}
```

- Then periodically call the message pump and check the status of the current programming task. E.g something like

```
bool finished = false;
int status = -1;
while (!finished) {
    pUpdater->MessagePump();
    Sleep(10);
    int newStatus = pUpdater->GetStatus();
    if (newStatus != status) {
        status = newStatus;
        switch (status) {
            case UBLAPI_READY:
                cout << "Could not connect to device. Aborting." << endl;
                finished = true;
                success = false;
                break;
            case UBLAPI_CONNECTING:
                cout << "Connecting..." << endl;
                break;
            case UBLAPI_ERASE:
                cout << "Erasing..." << endl;
                break;
            case UBLAPI_PROGRAM:
                cout << "Programming..." << endl;
                break;
            case UBLAPI_VERIFY:
                cout << "Verifying..." << endl;
                break;
            case UBLAPI_APPSTART:
```

```
                        cout << "Programming completed. Application started." << endl;
                        finished = true;
                        break;
                case UBLAPI_ERROR:
                        cout << "Error while programming." << endl;
                        finished = true;
                        success = false;
                        break;
        }
    }
}
pUpdater->Disconnect();
delete pUpdater;
pUpdater = NULL;
```

A fully working example is "pic32_firmware_upload_demo.exe", source code in folder .

It uses a Microchip example .hex file name by default, "Demo_App_PIC32_Starter_Kits.hex".

A sample run, using a custom firmware file for our own device "usb2drive" looked like this:

```
Kickdrive / Microchip PIC32 UBL Demo application. Use at own risk!
PIC32 UBL bootloader must be active.
Firmware hex file used: usb2drive_V1.1.1.353.hex
Press any key to continue . . .
Starting...
Connecting...
Erasing...
Programming...
Verifying...
Programming completed. Application started.
Microchip PIC32 UBL demo ended.
Press any key to continue . . .
```

Hierarchical Index

Class Hierarchy

This inheritance list is sorted roughly, but not completely, alphabetically:

Class Index

Class List

Here are the classes, structs, unions and interfaces with brief descriptions:

Class Documentation

_HIDD_ATTRIBUTES Struct Reference

Public Attributes

- ULONG **Size**
- USHORT **VendorID**
- USHORT **ProductID**
- USHORT **VersionNumber**

CBootLoader Class Reference

Public Member Functions

- void **ShutdownThread** ()
- void **CreateRxTxThread** (HWND)
- void **TransmitTask** (void)
- void **ReceiveTask** (void)
- bool **SendCommand** (char cmd, unsigned short Retries, unsigned short RetryDelayInMs)
- void **BuildRxFrame** (unsigned char *, unsigned short)
- void **HandleResponse** (void)
- void **StopTxRetries** (void)
- void **NotifyEvent** (unsigned int lEvent)
- void **GetRxData** (char *buff)
- void **GetProgress** (int *Lower, int *Upper)
- void **HandleNoResponse** (void)
- unsigned short **CalculateFlashCRC** (void)
- bool **LoadHexFile** (const char *hexFilePath)
- void **OpenPort** (UINT portType, UINT comport, UINT baud, UINT vid, UINT pid, USHORT skt, ULONG ip, HWND hwnd)
- BOOL **GetPortOpenStatus** (UINT PortType)
- void **ClosePort** (UINT PortType)
- BOOL **NotifyDeviceChange** (UINT portType, char *devPath)

Public Attributes

- bool **ExitThread**
- bool **ThreadKilled**
- CComPort **ComPort**
- CUsbHid **UsbHid**
- CEthernet **Ethernet**

CComPort Class Reference

Public Member Functions

- void **OpenComPort** (unsigned int, unsigned int)
- void **CloseComPort** (void)
- void **SendComPort** (char *, int)
- bool **GetComPortOpenStatus** (void)
- unsigned short **ReadComPort** (char *, int)

CEthernet Class Reference

Public Member Functions

- BOOL **OpenUdpPort** (USHORT, ULONG)
- void **CloseUdpPort** (void)
- void **SendUdpPort** (BYTE *, INT)
- USHORT **ReadUdpPort** (BYTE *, INT)
- BOOL **GetSocketOpenStatus** (void)

CHexManager Class Reference

Public Member Functions

- bool **ResetHexFilePointer** (void)
- bool **LoadHexFile** (const char *myHexFilePath)
- unsigned short **GetNextHexRecord** (char *HexRec, unsigned int BuffLen)
- unsigned short **ConvertAsciiToHex** (void *VdAscii, void *VdHexRec)
- void **VerifyFlash** (unsigned int *StartAdress, unsigned int *ProgLen, unsigned short *crc)

Public Attributes

- unsigned int **HexTotalLines**
- unsigned int **HexCurrLineNo**

CKickdPic32Ubl Class Reference

Main header file for the kickd_pic32_ubl DLL.
#include <kickd_pic32_DLLmain.h>
Inherits CWinApp.

Public Member Functions

- virtual BOOL InitInstance ()
- virtual int **ExitInstance** ()

Detailed Description

Main header file for the kickd_pic32_ubl DLL.

DLL entry point

Member Function Documentation

BOOL CKickdPic32Ubl::InitInstance ()[virtual]

multithreading and multiple instances not supported, go through global object always (improvement should protect this with a critical section)

CPIC32UBLDlg Class Reference

Modified CPIC32UBLDlg code.
#include <PIC32UBLDlg.h>
Inherits CWnd.

Public Member Functions

- void messagePump ()
  dispatches as messages and returns, when message queue empty
- void **OnBnClickedCancel** ()
- void **OnBnClickedButtonConnect** (UINT portType, INT comport, INT baud, UINT vid, UINT pid, ULONG ip, USHORT skt)
- void **Disconnect** ()
- void **ExtraInitialization** (void)
- void **StartSendReceiveTimers** (void)
- void **ProgressBarTask** (void)
- void **ProgramHexFile** (void)
- afx_msg LRESULT **OnReceiveResponse** (WPARAM, LPARAM)
- afx_msg LRESULT **OnTransmitFailure** (WPARAM, LPARAM)
- afx_msg void **OnTimer** (UINT nIDEvent)
- afx_msg void **OnBnClickedButtonErase** ()
- bool **OnBnClickedButtonLoadhex** (const char *hexFilePath)
- afx_msg void **OnBnClickedButtonProgram** ()
- afx_msg void **OnBnClickedButtonDisconnect** ()
- afx_msg void **OnBnClickedButtonVerify** ()
- afx_msg void **OnBnClickedButtonEraseProgVerify** ()
- afx_msg void **OnBnClickedButtonRunapplication** ()
- afx_msg void **OnBnClickedButtonBootVer** ()
- afx_msg void **OnBnClickedRadioEnableCom** ()
- afx_msg void **OnBnClickedCheckComEnable** ()
- afx_msg void **OnBnClickedCheckUsbEnable** ()
- afx_msg void **OnBnClickedCheckEthEnable** ()

Public Attributes

- char **TxCommand**
- char **TxData** [255]
- CString **cstringEditConsole**
- unsigned long **uLongEditBoxVID**
- unsigned long **uLongEditBoxPID**
- bool **boolRadioEnableCom**
- bool **boolRadioEnableUSB**
- BOOL **boolCheckComEnable**
- BOOL **boolCheckUsbEnable**
- CString **stringEditBoxUSBVID**
- CString **stringEditBoxUSBPID**
- BOOL **boolCheckEthEnable**
- DWORD **ip_value**
- UINT **valEditBoxSocket**
- int ublApiStatus

this is public information that is evaluated by UblApi

- int **ublApiProgress**

Protected Member Functions

- void **PrintKonsole** (CString string)
- void **ClearKonsole** (void)
- virtual BOOL **OnInitDialog** ()
- afx_msg void **OnSysCommand** (UINT nID, LPARAM lParam)

Protected Attributes

- CBootLoader **mBootLoader**
- bool **EraseProgVer**
- bool **ConnectionEstablished**
- UINT **PortSelected**

Detailed Description

Modified CPIC32UBLDlg code.

CUsbHid Class Reference

Public Member Functions

- BOOL **OpenUSBDevice** (UINT vid, UINT pid, HWND hwnd)
- BOOL **WriteUSBDevice** (CHAR *buffer, INT bufflen)
- USHORT **ReadUSBDevice** (CHAR *buffer, INT bufflen)
- BOOL **GetPortOpenStatus** (VOID)
- VOID **ClosePort** (VOID)
- BOOL **OnDeviceChange** (char *path)

Pic32UblRtLink Class Reference

kickd_pic32_ubl
Adapter class for accessing the kickd_pic32_ubl.dll without lib/header file.
#include <Pic32UblRtLink.h>

Public Member Functions

- int getDLLStatus () const
  returns the current status of the FTDI DLL support
- const char * getDLLFilename () const
  the actual DLL name / path used
- bool ConnectAndProgram (const char *hexFilePath, const int vid, const int pid)
  wrapper for the MicrochipUblApi.h / kickd_pic32_ubl.dll function
- void Disconnect ()
  wrapper for the MicrochipUblApi.h / kickd_pic32_ubl.dll function
- void MessagePump ()
  wrapper for the MicrochipUblApi.h / kickd_pic32_ubl.dll function
- int GetStatus ()
  wrapper for the MicrochipUblApi.h / kickd_pic32_ubl.dll function

---

Detailed Description

Adapter class for accessing the kickd_pic32_ubl.dll without lib/header file.

See Pic32UblRtLink - How to use for more information and example code.

Member Function Documentation

int Pic32UblRtLink::getDLLStatus () const

returns the current status of the FTDI DLL support

Returns:

0 - DLL ok
1 - one or several DLL functions are not available
2 - DLL not found

T_HEX_RECORD Struct Reference

Public Attributes

- unsigned char **RecDataLen**
- unsigned int **Address**
- unsigned int **MaxAddress**
- unsigned int **MinAddress**
- unsigned char **RecType**
- unsigned char * **Data**
- unsigned char **CheckSum**
- unsigned int **ExtSegAddress**
- unsigned int **ExtLinAddress**

Index