



Cleware USB Geräte mit Linux

Version 6.0.0
14.06.2022

Cleware GmbH
Am Lornsenpark 31
24837 Schleswig
Deutschland
www.cleware.de

1. Allgemeines

Alle Cleware-Geräte lassen sich auch unter Linux betreiben. Die im Downloadbereich befindlichen Dateien sind unter Ubuntu 20.04.4 LTS getestet.

Alle Funktionen der Geräte werden über eine Library „USBaccess.a“ und der dazugehörigen Headerdatei „USBaccess.h“ angesprochen. Die Funktionen sind weitgehend mit dem Interface der Windows-DLL identisch, sodass die darauf basierenden Programme nahezu ohne Änderung auf beiden Betriebssystemen übersetzbar sind.

Neben dem Interface sind verschiedene Beispielprogramme im Downloadbereich verfügbar:

Example	Einfaches Beispiel für USB-Temp und USB-Switch
USBadc	Abfrage von ADC, Luminos und PT100 Sensoren
USBswitchCmd	Programm zur Steuerung von USB-Schaltern, Multiplexern, Umschaltern und Ampeln
USBio	Steuerung des USB-IO16
USBcounter	Interface für den USBcounter
USBtemp	Abfrage von USB-Temp und USB-Humidity Sensoren

Um das Linux-System mit der graphischen Oberfläche „ClewareControl“ auf einem Windows-PC zu verbinden, kann man die Messwerte über TCP/IP Sockets übertragen. Hierzu liegt das Programm „send2cc“ bei. Damit werden die Messdaten angeschlossener Temperatur- und Feuchtigkeitssensoren an einen Windows-PC übertragen. Das Programm und seine Installation ist in Kapitel 5 beschrieben.

Zum Übersetzen der Programme wird einfach `make` aufgerufen. Fehlt hierfür die USB-Library, wird sie so installiert:

```
sudo apt-get update
sudo apt-get install libusb-dev
```

Der Aufruf der Programme erfolgt am einfachsten durch

```
sudo ./USBswitchCmd 1
```

wie hier am Beispiel für Switches. Ohne `sudo` müssen die Rechte für `/dev/hiddev..` gesetzt werden.

2. USBaccess.h

Die Datei USBaccess.h beinhaltet die Schnittstelle zu dem USB-Geräten der Cleware GmbH. Nach dem Einbinden der Datei können die Geräte geöffnet und gelesen werden. Die einzelnen Definitionen und Methoden sind identisch mit den Windows-Definitionen. Die detaillierte Beschreibung der Methoden ist in der Windows API-Beschreibung zu finden.

3. API-Beispiel

Das folgende einfache Beispiel zeigt die Anwendung der API zum Auslesen des Temperatursensors und zum Schalten eines Schalters unter C++. Wird das Programm ohne Argument aufgerufen, wird nach Temperatursensoren gesucht und in einer Schleife 10 Meßwerte ausgegeben. Mit einem Argument aufgerufen wird ein Schalter geschaltet (0=aus, 1=ein). Eine erweiterte Version des Bespiels befindet sich auf der CD.

```

...
#include "USBaccess.h"

int
main(int argc, char* argv[]) {
    CUSBaccess CWusb ;
    printf("Start USB Access Beispiel!\n") ;
    int USBcount = CWusb.OpenCleware() ;
    printf("OpenCleware fand %d Geräte\n", USBcount) ;

    for (int devID=0 ; devID < USBcount ; devID++) {
        if (argc == 2) { // nur den Schalter schalten
            if (CWusb.GetUSBType(devID) == CUSBaccess::SWITCH1_DEVICE) {
                if (argv[1][0] == '0')
                    CWusb.SetSwitch(devID, CUSBaccess::SWITCH_0, 0) ;
                else if (argv[1][0] == '1')
                    CWusb.SetSwitch(devID, CUSBaccess::SWITCH_0, 1) ;
                else
                    printf("Falsches Argument für den Schalter\n") ;
                break ;
            }
            else
                continue ; // die anderen Interessieren uns nicht
        }

        if ( CWusb.GetUSBType(devID) != CUSBaccess::TEMPERATURE_DEVICE &&
            CWusb.GetUSBType(devID) != CUSBaccess::TEMPERATURE2_DEVICE)
            continue ; // nur Temperatur lesen!

        CWusb.ResetDevice(devID) ;
        Sleep(500) ; // etwas warten

        // nun 10 Messwerte abfragen
        for (int cnt=0 ; cnt < 10 ; cnt++) {
            double temperatur ;
            int zeit ;
            if (!CWusb.GetTemperature(devID, &temperatur, &zeit)) {
                printf("GetTemperature(%d) fehlgeschlagen\n", devID) ;
                break ;
            }
            printf("Messwert %lf Grad Celsius, Zeit = %d\n",
                temperatur, zeit) ;
            Sleep(1200) ;
        }
    }

    CWusb.CloseCleware() ;
    return 0;
}

```

4. Programm „send2cc“

Das Programm „send2cc“ ist ein Programm zur Übertragung von Informationen der Cleware USB-Geräte eines Linux-PCs an die graphische Windowsoberfläche ClewareControl. Es werden alle Cleware Temperatursensoren und Feuchtigkeitssensoren unterstützt. Die Sensoren werden in einem festen Zeitintervall von 2 Sekunden abgefragt

Das Programm „send2cc“ hat verschiedene Optionen:

-s servername	spezifiziert den ClewareControl Server
-p	spezifiziert den Port (default 54741)
-r	im Fehlerfall automatisch wiederholen
-d	gebe debuginfos aus
-v	gebe Versionsnummer aus
-h	zeige Optionen

Die Angabe des Servernames ist als einzige Option zwingend erforderlich. Die Angabe kann als Name des Servers oder als IP-Adresse erfolgen.

Tritt bei der Verbindung ein Fehler auf, wird das Programm normalerweise beendet. Eine endlose Wiederholung der Verbindungsaufnahme kann mit der Option `-r` erreicht werden. Dies ist sinnvoll, wenn die Applikation als Service im Hintergrund läuft. Zwischen den einzelnen Versuchen wird jeweils 10 Sekunden gewartet.

Soll das Programm als Service im Hintergrund laufen, ist unter SuSE-Linux das folgende Vorgehen möglich (als User root):

- Kopieren von send2cc nach /usr/sbin
- `chmod 755 /usr/sbin/send2cc`
- Kopieren des Skriptes rcSend2cc nach /etc/rc.d
- `chmod 755 /etc/rc.d/rcSend2cc`
- `cd /etc/rc.d/rc5.d` (ggfs. auch rc3.d ...)
- `ln -s ../rcSend2cc S20send2cc`
- `ln -s ../rcSend2cc K20send2cc`

In dem Skript „rcSend2cc“ muß noch der Name des Servers eingetragen werden, der die Meßwerte erhalten soll. Die entsprechende Zeile lautet „CCSERVERNAME=XXXX“. Der Name XXXX ist auszutauschen. Zum Testen kann das Programm dann in /etc/rc.d mit „rcSend2cc start“ gestartet werden. Die Applikation ClewareControl sollte nun die angeschlossenen Sensoren mit den Meßwerten anzeigen. Wie ClewareControl als Server konfiguriert wird, ist im Handbuch zu ClewareControl ausführlich beschrieben.

5. Programm „USBtemp“

Während in dem Beispiel „Example“ der erste gefundene Temperatursensor ausgelesen wird, ist das Programm „USBtemp“ in der Lage, alle angeschlossenen Temperatur- und Feuchtigkeitssensoren auszuwerten. Hier die Optionen

```
//      USBtemp [-o filename] [-s serialnumber] [-d] [-b] [-h] [-i timeinterval]
//
//          -o filename: write to this file
//          -s serialnumber: get data from this device
//          -i interval between two samples in seconds
//          -b used in a batch, get a single value and exit
//          -d print out debug statements
//          -h help
```