



Cleware USB devices with Linux

Version 6.0.0
14.06.2022

Cleware GmbH
Am Lornsenpark 31
D-24837 Schleswig
Germany
www.cleware.de

Linux and Cleware USB devices

1. General

All Cleware devices will easily run in Linux systems. The programs in the download area were tested using Ubuntu 0.04.4 LTS and Raspberry PI.

The devices will be accessed via the library “USBaccess.a” and the corresponding headerfile “USBaccess.h”. The functions are identical to the functions supplied in the Windows DLL. Programs will run on both systems after recompilation. Only small changes in the include section are necessary.

Some example programs in the Cleware linux download :

Example	Simple program to access USB-Temp and USB-Switch
USBadc	Get data from ADC, Luminos and PT100 Sensors
USBswitchCmd	Control USB-Switches, Multiplexers and Traffic lights
USBio	Control USB-IO16
USBcounter	Interface to USBcounter
USBtemp	Get date from USB-Temp and USB-Humidity Sensors

To compile the programs just call make. If the USB-Library is missing, do:

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

To start the program just call

```
sudo ./USBswitchCmd 1
```

like this. Avoid sudo needs more access rights of /dev/hiddev...

2. USBaccess.h

The file USBaccess.h contains the interface to the Cleware USB devices. After including it, the devices could be opened and accessed. The description of the definitions and methods are identical to the windows API functions. Consequently the description of the methods could be found in the Cleware Windows API document.

3. API samples

The following simple sample demonstrates the usage of the API to access the temperature of an USB-Temp and to turn the state of an USB-Switch. The programs are written in C++. If this program will be called without an argument, an USB-Temp will be searched. If one is found, the temperature will be caught and printed 10 times. When an argument is supplied while starting the sample, this argument will be used as the state the USB-Switch should take (0=off, 1=on). An extended version of this sample is part of the CD.

```
...
#include "USBaccess.h"

int main(int argc, char* argv[]) {
    CUSBaccess CWusb ;
    printf("Start USB Access sample!\n") ;

    int USBcount = CWusb.OpenCleware() ;
    printf("OpenCleware found %d devices\n", USBcount) ;

    for (int devID=0 ; devID < USBcount ; devID++) {
        if (argc == 2) { // only USB-Switch
            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("Argument is wrong\n") ;
                break ; // only the first switch found
            }
            else
                continue ; // don't care about the others
        }

        if ( CWusb.GetUSBType(devID) != CUSBaccess::TEMPERATURE_DEVICE &&
            CWusb.GetUSBType(devID) != CUSBaccess::TEMPERATURE2_DEVICE)
            continue ; // only USB-Temp!

        CWusb.ResetDevice(devID) ; // reset the device (only once)
        Sleep(500) ; // wait a bit

        // get 10 values
        for (int cnt=0 ; cnt < 10 ; cnt++) {
            double temperatur ;
            int zeit ;
            if (!CWusb.GetTemperature(devID, &temperatur, &zeit)) {
                printf("GetTemperature(%d) failed\n", devID) ;
                break ;
            }
            printf("Temperature is %lf degree Celsius, time = %d\n",
                temperatur, zeit) ;

            Sleep(1200) ;
        }
    }
    CWusb.CloseCleware() ;

    return 0;
}
```

4. Application „send2cc“

The program “send2cc“ is a Linux program that sends informations about Cleware USB devices to the graphical user interface ClewareControl running under Windows. All Cleware temperature and humidity sensors are supported. The sensors are scanned in a 2 seconds interval to get the measured values.

Several options are available:

-s servername	specify the ClewareControl server
-p	specify the port (default 54741)
-r	repeat action in case of an error
-d	print debug info
-v	print version number
-h	show options

The server name option must be supplied anyway, the other options are optional. The server name could be supplied as the name of the server or as the IP number.

In case of an error while connecting or talking to the server the application normally exits. This could be avoided by using the option `-r`. In this case, the action will be retried in an endless loop after waiting 10 seconds.

If the program should be running as a background process with SuSE Linux, the following steps could be used to implement a service:

- login as root
- Copy send2cc to /usr/sbin
- `chmod 755 /usr/sbin/send2cc`
- Copy script rcSend2cc to /etc/rc.d
- `chmod 755 /etc/rc.d/rcSend2cc`
- `cd /etc/rc.d/rc5.d` (and rc3.d ... if applicable)
- `ln -s ../rcSend2cc S20send2cc`
- `ln -s ../rcSend2cc K20send2cc`

The name of the server must be entered in the script “rcSend2cc“. The corresponding line is “CCSERVERNAME=XXXX”. The text XXXX must be replaced by the correct name or IP address. To test the program call “rcSend2cc start” in /etc/rc.d. The application ClewareControl should show the remote sensors. For setting up ClewareControl to act as a server please refer to the ClewareControl manual.

5. Programm „USBtemp“

This program get the temperature of the USB sensor with the following options:

```
//      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
```