Using DAQFlex with C++ in Linux

This example is meant as an introduction to using the DAQFlex Firmware Architecture (e.g. without the Mono/.NET API library) with C++ in Linux. It describes how to use libusb's synchronous device I/O to send commands and receive data from a DAQFlex device. Note that only certain features of the DAQFlex API are implemented in the example.

The design process is outlined below, and follows along with “DAQFlex_libusb.tar.gz”. The archive contains a Code::Blocks 8 project that demonstrates implementation of the steps. The bulk of the code is in mccdevice.cpp and mccdevice.h, where an mccdevice object is created. The file main.cpp creates an MCCDevice object, sends control messages and performs an analog input scan using a thread. The data is then scaled and calibrated, displayed in the console and written to a *.csv file.

Steps for Communicating with an DAQFlex device using libusb

I. Initialization and device selection (MCCDevice constructors)

A. Initialize libusb

B. Get the list of USB devices

C. Search this list to find an appropriate device

D. Open, claim, and initialize the device

II. Send and receive basic messages using USB control transfers

A. Send a message (sendControlTransfer())

B. Receive a response (getControlTransfer())

C. Combine both into a single function, as every message has a response in DAQFlex (sendMessage())

III. Receive scan data synchronously using a USB bulk transfer (readScanData())

IV. (Optional) Receive scan data asynchronously by threading the USB bulk transfers (startContinuousTransfer() and stopContinuousTransfer()), write calibration routines (scaleAndCalData()), etc.