

Read Me

Contents of This File

- Introduction
- What You Need to Know
- Configuration Instructions
- Operating Instructions
- Copyright and Licensing Information
- Credits and Acknowledgements

Introduction

The SDK provides an API and software components that enable third party software developers to control their meters and access data from meters or downloaded files. The SDK allows access to the Model 831A and Sound Advisor Model 831C via USB (Microsoft ® Windows® or Linux) or through an 831-INT-ET (Any OS) or over TCP/IP. The SDK also allows limited access to the LxT SoundTrack® over USB only (both Windows and Linux). Licensing options are also available for Linux USB connectivity for the BeagleBone Black and Raspberry Pi. The SDK may be used to communicate with an HVM200 over USB or TCP/IP (over WiFi).

OS Requirements

- For Windows SDK:
 - Windows 10 Pro
- For Linux SDK:
 - 32bit Raspbian Linux, Jessie Image on Raspberry Pi or
 - 32bit Debian Linux Jessie Image on BeagleBone Black

Related Hardware Requirements

For the Linux SDK option, you will need either a BeagleBone Black or Raspberry Pi3 configured using the latest updates. BeagleBone Black must have the Jessie version from their web site.

Your Hardware

An NRE fee may be required if your hardware or OS specifications do not meet the requirements above.

What You Need to Know

The SDK is built upon HTTP requests and responses. You will be able to use most programming languages to communicate with the meter, as long as the language supports HTTP communication. Most of the samples we provide are in C# or JavaScript, with a couple in C++. The responses from the meters are mostly in JSON, which necessitates an understanding of how to read JSON. There are open source libraries available to read and parse JSON to simplify programming.

Prerequisite Skills

You will need the following skills to be able to use the SDK.

- JSON – JavaScript Object Notation
- One of the following:
 - JavaScript/HTML5

- C#
 - C++
 - Java
 - Other HTTP friendly language
- Understanding of HTTP Requests

Configuration Instructions and Software Requirements

In order to compile some of the example code in Windows, you will need Visual Studio 2015 or higher. The C++ code examples may be setup to compile in Linux.

In Windows, to use the SDK or to execute the C++ examples, the VS2015 C++ redistributables are required. You may install G4 to ensure that the proper prerequisites have been installed. This includes but not limited to the libusb driver and Newtonsoft.Json.

Likewise for the C# examples, Visual Studio is required to compile and run. We are using .Net 4.5.2 though for the examples you may be able to use 4.0 or greater.

Operating Instructions

The operation or usage of the SDK and associated information is found in the Documentation folder.

Examples

The SDK contains several examples to help get you started developing and communicating with the meter. All samples, code and documents are copyrighted. See the Copyright and Licensing section below. The following are included in the SDK.

C++

The following C++ examples are included in the SDK

SdkTestApp – Performs several operations (run, stop, store, set properties, file list, download) after connecting to a meter.

StreamingDataDemo – a command line method with output of streaming data from the meter.

C#

The following C# examples are included in the SDK:

DataStreamAndDownload – Demonstrates how to download and polling stream data concurrently.

LiveStreamDemo – User Interface to Streaming data from the SLM.

HVMStreamingData – Demonstrates how to stream data from HVM.

SdkCommon – Helper classes for SLM, HVM and command line args.

SdkDownload – File list and Downloads files from SLM.

SoundRecordingDemo – Demonstrates how to handle the sound recordings from an LDBin file, displays sound records in a table and allows playback of the sound records.

TimeHistoryDemo – Demonstrates how to access Time History data from LDBin file.

VorbisPlayer – Allows us to test Audio Streaming and play OGG files.

HTML5 and JavaScript

GreenStream.html – shows how to get a single streaming data point directly from the meter.

StreamingData – demonstrates more of the streaming functionality.

WebSockets – Helps you to create the calls for a Web Socket call. It also shows the data from the meter and allows you to set the refresh rate and from which TAGs to get data.

Copyright and Licensing Information

The example/samples code, related files and documentation are part of the SDK and are covered under the license agreement found in the Documentation folder. The use of the SDK means acceptance of the end user license agreement.

Certain licenses and copyright information for added libraries can be found in your Other Material in the installed Documentation folder.

Credits and Acknowledgements

Software and Firmware development teams of Larson Davis a PCB Piezotronics division:
Scott Lockwood, Jim Martin, Justin Johnson, Alex Blodgett, Kevin Lawrence, Brad Diamond, John Filoso, Emma Hoggan, Nick Rasmussen, Dan Nelson, Ross W, Kiaya N, Allen P, Jake G.
Brent Iverson

Assistance by:
Ken Cox, Ken Isle.