X-Rite Company Confidential Information
For Distribution under NDA only

Application Note: I1Display3 Calibration Management

DOCUMENT REVISION 7:4-OCTOBER-2019

Contents

Introduction

All calibrated measurements of displays with i1Display3 devices rely upon calibration parameters calculated by the i1d3SDK software. The calculation of each set of calibration parameters is performed at run-time based upon the spectral sensitivities of the particular i1Display3 as stored in the EEPROM in the factory, data describing the Spectral Power Distribution of the source to be measured as contained in Electronic Display Reference (EDR) files, and reference Standard Observer Color Matching Functions (CMFs). The Run Time Library (RTL) of the SDK inventories available calibrations (from which the calibration parameters are calculated); building a list for application use as follows:

- An initial matrix is calculated using CMF data stored internal to the SDK and is selected as the default calibration and added to the list.
- If the SDK has access to Emissive Display Reference (EDR) files for use with generic display types (see companion document Emissive Display Reference (EDR) File Principals of Operation for further information), those calibrations are included in the list. These generic EDR files are normally installed with the SDK Run Time Library (RTL) and are considered part of the SDK.
- Later, and optionally, upon application selection of a custom EDR file, that calibration is added to the list. Such custom EDR are stored externally to the SDK and are referenced by file name.

The appropriate calibration to use generally depends only on the Spectral Power Distributions, P(λ) of the primaries of the display. The P(λ) in turn depends only on the backlight type in the case of LCD displays, lamp type in the case of front screen projectors, and display type in the case of other displays. EDR files are installed in the SDK for the following types of displays:

- LCDs with standard gamut CCFL backlight
- LCDs with wide gamut CCFL backlight
- LCDs with white LED backlight
- LCDs with RGB (wide gamut) LED backlight
- Front screen projectors with Ultra High Pressure (UHP) lamps
- Plasma displays
- LCDs with OLED backlight
- LCDs with RG-Phosphor backlight

Generally the strategy that an applications programmer should use in choosing a matrix is as follows, though the exact order of these steps may depend on the specific application flow and desired user interaction:

1. Determine calibrations available as installed in the SDK
2. Determine the type of display being calibrated by some combination of the following:
    a. Hard-coded if application is meant to support only one model or range of models of display
    b. Determine display model from EDID and then consult a hard- coded or file-based table to determine display type from model string in EDID
    c. Ask the user through a pull down
3. If necessary, Load Custom EDR file from file location external to the SDK
4. Match available calibrations from steps 1 and 3 against required calibration from step 2 and set the resulting calibration as active
5. Measure normally

Additional detail for these general steps is described in the following sections.

Finding and Selecting Calibrations

In order to understand how to select a calibration, it is necessary to understand the data structures provided to the application program. Header files containing full structure definitions are provided along with the SDK and are not fully described in this document.

After opening the i1Display3 (using the **i1d3DeviceOpen** function), the application can get the list of available calibrations, examine the source (type) of calibration, and then select one of them. To do this:

1. Call **i1d3GetCalibrationList()** to get the list of entries and construct a set of names for user selection in, for example, a pull down.
2. Look at the **calSource** field of each entry.
    a. If **calSource** is **CS_GENERIC_CMF**, show the user a "default" or "generic CMF" name for calibration selection. Recommended name: "Generic CMF."
    b. If **calSource** is **CS_GENERIC_DISPLAY** (from SDK's supplied EDR file), use the value of the **edrDisplayType** field to select a string to show the user for selection of display type. The application should use the string as translated, if necessary. English language versions of the display type strings are supplied in the file TechnologyStrings.txt, which is provided along with the EDR files. The list of possible **edrDisplayType** values is shown in Table 1.
3. If a custom EDR is required, it is the application's responsibility to load that file into the SDK for calibration use. The SDK provides functions to do that. After that has been done,

the name of that calibration will also appear in the list returned by **i1d3GetCalibrationList()**.

Selecting Calibration For A Known Monitor Type

If the program that is using the i1display3 is intended for a single display type (i.e., a vendor-specific one) the user may not need to select any calibration. Such a situation might include a monitor OEM supplying a program along with the i1display3 to calibrate a particular White LED backlit display, for example. In that case, the programmer can simply follow the description in the previous section and select the calibration that matches the technology type as described in item 2 above. Failure to find the correct technology type would indicate that the EDR file for that display has not been provided.

User Selection of Calibration

If the technology type of the display is not known when the program is executed, the user may be presented with the choice of available calibrations (typically in a selection listbox built with the strings referenced in the calibration list's entries). When the correct calibration has been selected the program calls the **i1d3SetCalibration()** function with a pointer to the calibration entry matching the type selected.

Technology Type Values (edrDisplayType field values)

The SDK has been designed to support EDR files for any technology type including some that may not be known (or even yet exist) when the SDK is released. Hence, the enumeration values that may be used are not provided as part of the SDK code. Instead, the values are provided in the file **TechnologyStrings.txt** which is supplied along with the EDR files as a separate release package. As of initial release, the contents of that file are as shown in Table 1.

Table 1. edrDisplayType field values

1,Custom
2,CRT
3,CCFL
4,CCFL
5,CCFL
6,Wide Gamut CCFL
7,Wide Gamut CCFL
8,Wide Gamut CCFL
9,White LED
10,White LED
11,White LED
12,RGB LED
13,RGB LED
14,RGB LED
15,OLED
16,AMOLED
17,Plasma
18,RG Phosphor
19,Projector (RGB Filter Wheel)
20,Projector (RGBW Filter Wheel)
21,Projector (RGBCMY Filter Wheel)
22,Projector

Table 1 contains redundant values. X-Rite Design Verification testing indicates that calibration depends only on the display system backlight technology, and on no other factors, for which the additional entries were originally reserved. Only one of each type are used. As of this writing, X-Rite supplies seven EDR files; for technologies 3 (CCFL), 6 (Wide Gamut CCFL), 9 (White LED), 12 (RGB LED), 15 (OLED), 17 (Plasma) and 22 (Projector). These are listed in the i1d3Mapping.txt file along with the names of the EDR files as shown in Table 2.

Table 2. Contents of i1d3Mapping.txt, EDR files supplied as part of SDK installation

3,CCFLFamily_07Feb11.edr
6,WGCCFLFamily_07Feb11.edr
9,WLEDFamily_07Feb11.edr
12,RGBLEDFamily_07Feb11.edr
15,OLEDFamily_20Jul12.edr
17,PlasmaFamily_20Jul12.edr
18,RG_Phosphor_Family_25July12.edr
22,ProjectorFamily_07Feb11.edr

To summarize:

- If hard-wiring end user application to a specific monitor type, call **i1d3GetCalibrationList()**, iterate over the list of returned calibrations looking for a match of the technology type to match the display type and select that one.

- If prompting user to select type, call **i1d3GetCalibrationList()** to get all available technologies, prompt user to select one (through a listbox or similar mechanism) and select the calibration user picks.

Using CMF values

In some cases it might be necessary to overwrite the SDK's internal CMF values. The SDK provides the function **i1d3LoadCMFDataFile()** to replace the RTLs internal CMF tables represented by the **CS_GENERIC_CMF** calSource, which is a representation of the 1931 CIE Standard (2°) Observer. The function allows to overwrite the spectral CMF tables, the ambient CMF tables, or both. If the function returns successfully, the new CMF is available under the **CS_GENERIC_CMF** calSource.

Note: *After loading the CMF data file using the* **i1d3LoadCMFDataFile()** *function, a previously selected EDR calSource **must** be set again as described in the previous chapters. The EDR calSource then will be recalculated with the custom CMF matrix. To reset the CMF to its default values reinitialize the RTL using* **i1d3Destroy()** *and* **i1d3Initialize()**.

CMF file format

A CMF file is a simple plain text file. The content of the file consists of 5 lines. Windows, Unix and Macintosh line terminators are supported. The X, Y and Z CMF values surrounded by start and end tags. The format is as follows:

1) Start tag: <CMFDATA>
2) CMF samples for the X CMF.
3) CMF samples for the Y CMF.
4) CMF samples for the Z CMF.
5) End tag: <CMFDATA>

The CMFs are stored as **space** or **tab** separated floating point values for the range **380-730nm** in **1nm** steps. The values must be provided in units of lumens/watt. The only accepted floating point character is the decimal point character "**.**" (without quotes).

Schematic CMF example (shortened)

<CMFDATA>

X_{380nm} X_{381nm} … X_{730nm}

Y_{380nm} Y_{381nm} … Y_{730nm}

Z_{380nm} Z_{381nm} … Z_{730nm}

<CMFDATA>

CMF example with actual values (shortened)

<CMFDATA>

0.011859646728	0.015932387654	…	0.278261844820
0.109247535904	0.146900070160	…	0.713573169520
0.481363417552	0.647642986460	…	0.000000000000

<CMFDATA>