

## Blink 1920 HDMI Software Development Kit (SDK)

Documentation for the SDK can be found in the user manual at:

C:\Program Files\Blink 1920 HDMI\1920 HDMI User Manual.pdf

Files to Link to:

**Blink\_C\_wrapper** (.dll, .h, and .lib) is used as a wrapper class to reformat C++ functions to be compatible with Matlab, LabVIEW, and Python

**HDMIDisplay** (.dll, .h, and .lib) is the C++ code that interfaces to the SLM hardware

**ImageGen** (.dll, .h, and .lib) provides image generation capabilities as documented in the user manual

*Example Programs:*

**BlinkSDKExample** (.cpp, .m, .py, and .vi) is a simple program that loads a series of images to the SLM through C++. This example writes slowly varying gradient patterns to the SLM that are easily visible to the eye when looking at the SLM with room light through a polarizer aligned at 45 degrees to the SLM.

**HMDIDiffractiveTest** (.cpp, .m, and .vi) is an example program illustrating how to generate the measurements required for a custom LUT calibration. The example can generate either a regional or global calibration of applied voltage to output phase for “n” waves of phase delay. The program illustrates how to: initialize the SLM, generate and load the diffractive images required by the calibration routines, and how to format the output file with the measurement data. The end user must fill in code to read from their specific analog input board. More details on the calibration optical setup, measurements, and how to use the post processing software can be found in the user manual.

**DiffractiveLUT.exe** is the program use to post process the measurements generated by PCIDiffractiveTest to generate either a global or regional calibration. More information can be found in the user manual.

This folder also contains miscellaneous supporting C++ files, image files, DLL's for the device driver, and generic calibrations. If you have a specific question about the additional supporting files please contact [slmsupport@meadowlark.com](mailto:slmsupport@meadowlark.com)