Display++

Getting Started

Version R10

June 2018

Cambridge Research Systems Ltd

80 Riverside Estate
Sir Thomas Longley Road
Rochester
Kent
ME2 4BH
United Kingdom

www.crsltd.com

Introduction

Display++ makes it simple to display calibrated visual stimuli with precision timing, and provides robust and reliable synchronisation of the stimulus presentation with external data collection equipment.

Configurable contrast resolution combined with fast panel drive rate, custom lag free electronics, and a configurable strobing or scanning LED backlight are some of the tools that make Display++ ideal for cognitive, psychophysical and neurophysiological investigations of vision and the brain.

System Features

- 32″ 1920x1080 IPS LCD with 120 Hz panel drive.
- 1400:1 contrast ratio (typical; actual value measured and reported).
- 10-bit RGB native, configurable up to 16-bit using temporal dithering algorithms.
- Hardware gamma correction tables, spatial uniformity correction plus CIE XYZ Colour Management System ensure accurate colour and luminance reproduction.
- Real time calibration with integrated colour sensor provides stable peak output.
- SuperBright white edge illuminated LED backlight. Supports DC, scanning and strobing modes to minimise transition artefact.
- Individual Factory Calibration with full spectral and colorimetry report provided.
- Multiple synchronous TTL trigger outputs.
- Multiple synchronous TTL trigger inputs.
- Optional integrated IR touchscreen.
- Optional integrated synchronous analogue I/O.
- Optional integrated FPR layer for stereoscopic display.
- Integrates with CRS accessories including audio stimulator, eye tracking and behavioural response devices.
- Compatible with community software tools like Psychtoolbox and PsychoPy, and commercial tools like Presentation and Psykinematix, or your own software.

Parts List

Display++ Monitor — 32″ with removable Desktop Stand.

Getting Started Guide — Printed version of this document.

Factory Calibration Report — Printed version specific to each Display++ Monitor. Labelled with Serial Number.

SD Card — San Disk 4 GB or 8GB SD/SDHC card or similar. Contains Display++ firmware/logic files, drivers, documentation, and ancillary software. Usually this is already inserted in the rear of the Display++ unit.

USB cable — 3 m Standard A to Standard B.

HDMI to Dual-Link DVI — 3 m.

Power Supply Unit — IEC320-C8 input connector. 165 W, 12 V DC output. 1 m attached cable terminates in XLR connector.

Power Cord — 2 m IEC320-C7 power cord. Terminates in region-specific plug.

Hardware Requirements

Host Computer

- Operating System: Windows XP SP3 or later (ideally Windows 10), Mac OS X 10.11 (El Capitan) or later, or a Linux distribution like Ubuntu 16.04 or later.
- One USB port.

Graphics Card

- Discrete NVIDIA or AMD GPU, or integrated Intel HD 4000 graphics or later, with high-bandwidth, digital video output for 1920x1080 @ 120 Hz displays.
- HDMI 1.4 or later that supports 1920x1080 @ 120 Hz. Check the vendor's specification to confirm compatibility.
- A low-bandwidth HDMI will constrain the output to 1440x1080 @ 100 Hz. This is the maximum that can be achieved with a 165 MHz pixel clock.
- For correct operation with Display++, the graphics card will require an identity CLUT (linear ramp) to be loaded into its palette. Spatial and temporal dithering must be disabled. Please refer to the Bits# Technical Manual for more information about how to configure the graphics card for use with the Display++ system. If using Psychtoolbox, the function BitsPlusIdentityClutTest should be used to confirm correct operation.

Installation

Make the following connections between Display++ and the Host Computer. Start the procedure with both devices switched off.

1. Using the provided USB cable, connect the Display++ USB Type B port to a spare USB Type A host port on the Host Computer. The USB Type B port is labelled “Computer” on the row of Display++ connectors.
2. Assemble the Power Supply Unit (PSU) with the supplied Power Cord for connection to the AC mains outlet. Do not connect the Power Cord to the AC outlet.
3. Connect the PSU XLR plug to the Display++ DC power input socket.
4. Connect the Power Cord to the AC mains outlet; switch on the outlet if appropriate.
5. Notice that Display++ generates its own Status Display with diagnostic information (see Figure 1). This self-generated output is produced in the absence of an appropriate video signal. If this is not seen, check the Status LED on the rear of the unit and refer to the table in Appendix A.

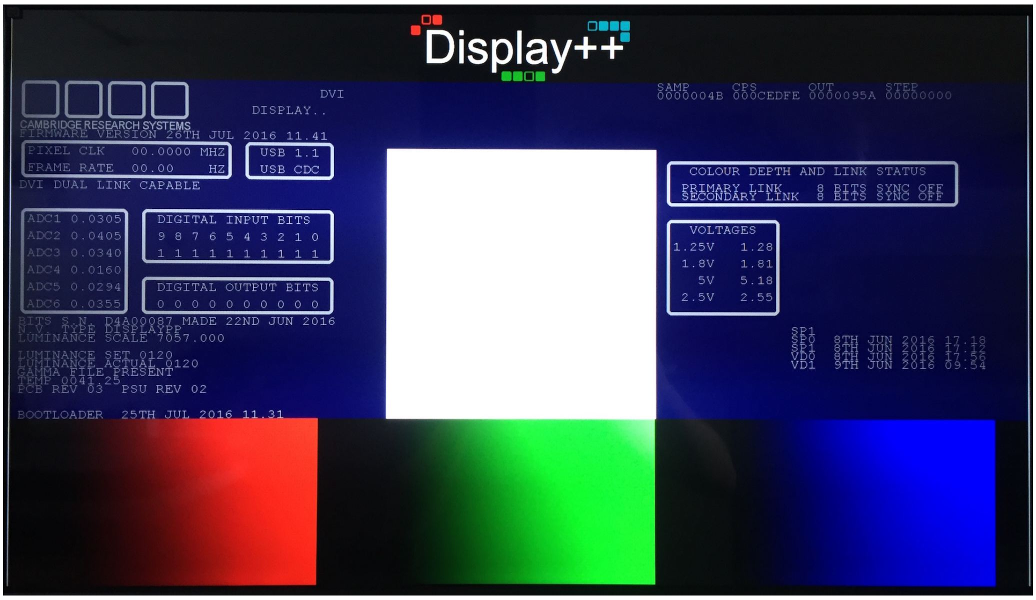

Figure 1. Display++ - self-generated ‘Status Display’.

6. Connect the HDMI to Dual-Link DVI cable between the Host Computer graphics card and the Display++ DVI connector. The icon for the DVI connector, in row of Display++ connectors, is labelled “HDMI” to indicate the video format (the supplied HDMI to DVI cable must be used). The icon also shows the correct orientation of the DVI cable.
7. Switch on the Host Computer and login to your Operating System desktop.
8. Notice that Display++ operates just like other desktop LCD monitors; the Status Display mode will switch to showing the video signal generated by the Host Computer as soon Display++ detects an appropriate HDMI signal.
9. The Operating System should detect the Display++ USB connection as a Removable Disk Drive volume, labelled DisplayPP. This interface is known as USB Mass Storage Device (MSD) mode. Opening the DisplayPP volume will show the contents of the SD Card inserted in the rear of the Display++ unit.

10. The Display++ HDMI connection should be reported as 1920x1080 @ 120 Hz or 100 Hz. If a lo bandwidth HDMI output from the graphics card is used, the 1440x1080 @ 100 Hz output from Display++ will be centred in the middle of the active LCD module.

The next steps will reconfigure the USB interface to facilitate bidirectional communication between Display++ and the Host Computer. The communication mode uses the USB CDC ACM device class; a virtual serial port (VCP) is generated in CDC mode.

11. Windows 10, Mac OS X 10.11 and recent Linux distributions will automatically detect and enumerate the CDC VCP, then load the class driver included with the operating system. Earlier versions of Windows require a suitable INF driver information file to configure the VCP: copy the folder “USB CDC INF for Windows” from the DisplayPP MSD volume to the Windows desktop; open the folder, then right-click on the file “crsltd_usb_cdc_acm.inf” and choose **Install** to load the class driver.
12. Copy the folder “Documentation” from the DisplayPP MSD volume to the local drive. The folder contains a PDF version of this manual plus the Bits# Technical Manual, which explains how to use the integrated Bits# (Bits Sharp) features of the Display++ system.
13. Open the “Firmware” folder on the DisplayPP MSD volume and load the file “config.xml” into a text editor. We recommend using Notepad on Windows or the XCode IDE on Mac OS X (do not use TextEdit on OS X – this can corrupt the file). On GNU/Linux, use gedit or similar.
14. Change `<Entry USB_CDC=”OFF” />` to `<Entry USB_CDC=”ON” />`. Note that the OFF and ON states are wrapped between ASCII “ characters and not UTF-8 “” characters. If you are using Mac OS X 10.10 or earlier, also change `<Entry USB_MSD=”ON” />` to `<Entry USB_MSD=”OFF” />`. Save and close the file.
15. Right-click on the DisplayPP volume icon and choose Eject. This will restart Display++ with the new settings applied. If Display++ does not reset at the Eject event, switch off the AC mains outlet (or remove the Power Cord from the outlet), wait a few seconds, then switch on (or reconnect the Power Cord). During the restart procedure, Display++ will briefly display the Status Screen before switching back to the output from the graphics card.

Controlling Display++ from MATLAB

Display++ contains integrated Bits# functionality. Please refer to the Bits# Technical Manual to learn how to control Display++ using MATLAB and the freely available Psychtoolbox-3 extensions. Any other software system which contains support for the Bits# device can also be used.

Summary of USB CDC ACM mode Commands

For testing purposes a terminal emulation program (e.g. “Screen” on Mac OS X and GNU/Linux or “PuTTY” on Windows) can be used to type the commands. However, the commands would be usually be scripted as part of the experiment. This is documented in the Bits# Technical Manual. All commands are preceded with a dollar sign $. Returned data is always delimited with semicolons:

Generically commands follow the syntax of either:

>>$<command>

Or if a parameter is updated:

>>$<command>=[<command data>]

The commands must be followed by a carriage return (use ASCII code ‘13’ in MATLAB). Returned results follow this format:

>>$<data type>;<value>;<value>;

Table 1.USB CDC Commands.

Command	Arguments	Description
$Help	(none)	Returns a list of all commands.
$Start	(none)	Resets the sample number and starts acquiring analogue and digital input data. Returns: #sample;<sample#>;<timestamp>;<trigger-in bit value>;<digital input bits 0 .. 9>;<IR response bits 1 .. 6>;<ADC 1 .. 6>;
$Stop	(none)	Stops data acquisition. No return.
$ProductType	(none)	Returns the name and type of the product. Return example: #ProductType;Display++;
$SerialNumber	(none)	Returns the 8 character serial number. Returns: #SerialNumber;12345678;
$FirmwareDate	(none)	Returns the firmware compilation date. Returns: #FirmwareDate;DD/MM/YYYY hr:min;
$USB_massStorage	(none)	Enables Display++ as a Mass Storage Device and disables the CDC interface. To revert back to the CDC mode, make sure Display++ is set to boot in CDC mode before restarting the system. No return.
$setMonitorType	=[<filename. Edid>]	Sets the EDID file that Display++ sends to the host computer. The EDID file must be located in the EDID folder on the SD Card. Returns the updated status, or if no argument, the current status. Default is ‘AUTO’. Return example: $setMonitorType=AUTO
$autoPlusPlus	(none)	Sets T-Lock control of the video mode. No return.
$monoPlusPlus	(none)	Switch to Mono++ video mode. No return.
$colourPlusPlus	(none)	Switch to Color++ video mode. No return.
$colorPlusPlus	(none)	Switch to Color++ video mode. No return.
$BitsPlusPlus	(none)	Switch to Bits++ video mode. No return.
$statusScreen	(none)	Force display of Status Screen when video is present. Revert with a video command e.g. $monoPlusPlus. No return.
$TemporalDithering	=[<ON | OFF>]	Enable or disable temporal dithering. Returns the updated status, or if no argument, the current status. Return Example: $TemporalDithering = ON
$EnableTouchScreen	=[<ON | OFF>]	Enable or disable integrated Touch Screen. This command can only be used with Display++ Touchscreen models. Returns the updated status, or if no argument, the current status. Return Example: $TouchOutput = ON When the touch screen is on, it will return data of the following format (a string for each new position detected): $touch;aaaa.aaaa;bbbb;cccc;d; aaaa.aaaa: Timestamp showing the time in seconds since the monitor was switched on (or since a Tlock command reset it).

		Bbbb: X position of the last touch cccc: Y position of the last touch d: Flag indicating whether touchscreen is being pressed (1) or released (0)
$enableGammaCorrection	=[<filename.txt>]	Sets the gamma correction look-up-table (LUT) to load into the hardware memory. The specified file must be located in the Gamma folder on the SD Card. Factory distribution provides a LUT containing a linear ramp (i.e. no transformation) named 13bitLinearLUT.txt; a gamma corrected (linearized output) named invGammaLUT.txt and a LUT for adjusting the white point to D65 named D65invGammaLUT.txt are also provided. Refer to Factory Calibration Report for unit specifics. Return Example: $enableGammaCorrection;13bitLinearLUT.txt;
$Beep	=[<freq.>,<duration>]	The Beep command activates a piezo sounder inside the Display++. This can be used to provide feedback on e.g. button pushes. Valid arguments are in the range 10 20,000 Hz for frequency (freq.) and 0.0001-6.5 seconds for duration. NB. An extreme frequency or very short duration can be inaudible! Example use: $Beep=[1000,2] . No return.
$GetVideoLine	=[<line#>,<n>]	Returns n pixels starting from the line specified. This command is extremely useful for monitoring the actual RGB values received from the graphics card. Use this command to learn why the T-Lock encoded packets are failing to be recognised by Display++. Note that the Status Screen mode will be enabled when using this function. Example use: $GetVideoLine=[1,265] returns a string beginning with the command name followed by the RGB values of each pixel: #GetVideoLine;<R1>;<G1>;<B1>;<R2>;<G2>;- <B2>; … <Rn>;<Gn>;<Bn>;
$btn<ID>	=[<event source>]	Response Box: Set the event source for one of the four buttons. <ID> can be 1, 2, 3 or 4. <event source> can be one of the 17 input event sources (see list below). No return.
$light	=[<event source>]	Response Box: Set the event source for the light. <event source> can be one of the 17 input event sources (see list below). No return.
$pulse	=[<event source>]	Response Box: Set the event source for the pulse. <event source> can be one of the 17 input event sources (see list below). No return.
$trigger	=[<event source>]	Response Box: Set the event source for the trigger. <event source> can be one of the 17 input event sources (see list below). No return.
$VideoPixelClock	(none)	Returns the video pixel clock in MHz. Return example: $VideoPixelClock;108.0020;
$VideoFrameRate	(none)	Returns the video frame rate in Hz. Return example: $VideoFrameRate;120.02;
$LuminanceVariables	(none)	Returns the current state of the luminance variables from the integrated light measuring feedback system. Return format: $LuminanceVariables;aaaa;bbbb; cccc.ccc; dddd;-eeee;-ffff.ff; aaaa: Actual luminance (counts per second / scale factor). Bbbb: Set point aim value (from config.xml file).

		cccc.ccc: Scale factor (calibration factor) held in the non-volatile area of the processor but can be overwritten in the config.xml file. Dddd: Current PWM value. 0 is 0% and 4095 is 100%. The PWM has lower and upper limits defined in the config.xml file (EntryLumCtrlLowLimit and EntryLumCtrHighLimit). Eeee: The last step that was applied to dddd above (this is for CRS engineer debug). Ffff.ff: Temperature of the LED backlight in 0.25°C steps.
$LuminanceSetPoint	=[<luminance>]	Sets the luminance in cd/m^2. Returns the updated status, or if no argument, the current status.
$enableColourCorrection	=[<ON | OFF>]	Enables or disables the factory hardware colour correction.
$spatialCoeffs	=[<spatial correction coefficients>]	Returns the current coefficients for luminance spatial uniformity correction. If an input is provided with the 22 coefficients for spatial uniformity correction these values will overwrite the coefficients defined in the config.xml file and be used until the next reset. The first coefficient should always be zero. The next 21 coefficients define the entries a0 through a21 in the config.xml file. To disable the uniformity correction use zero (0) for all coefficients except the third coefficient, i.e. $spatialCoeffs =[0 0 1 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0]. This can also be permanently set in the config.xml file.

Note that for each gamma correction table defined in the config.xml file there is an associated luminance scaling factor. The gamma tables for native white point and D65 correction have two luminance scaling factors, according to whether the spatial uniformity correction is enabled or not. By default, Display++ is set to use the linearised native white point (i.e. the invGammaLUT.txt correction table), with the colour correction and spatial uniformity correction both enabled. Please refer to the Factory Calibration Report provided with your unit. Use the USB CDC commands if you want to change the default settings, or edit the config.xml file to make permanent changes.

List of Response Box event sources

Din0
Din1
Din2
Din3
Din4
Din5
Din6
Din7
Din8
Din9
TrigIn
IrbuttonA
IrbuttonB
IrbuttonC
IrbuttonD
IrbuttonE
IrbuttonF

Technical Data

System Environmental Conditions

Indoor use only.
Operating Temperature: 0°C to +30°C *
Cooling: Convection-cooled.
Operating (or Storage Humidity up to 40°C): 90% RH, non-condensing.
Storage Temperature: -20°C to +60°C.

Monitor Characteristics

Screen size: 32" diagonal (active area 710 mm x 395 mm).
Screen resolution: 1920 x 1080 pixels (Note: a small area about 16 x 16 pixels in the top left corner is masked for the built in luminance sensor).
Grey-to-Grey Response time: 5 ms.
Video Input: 1920 x 1080 pixels, HDMI, 120 Hz, 24 bit.
Peak white output: 350 cd/m^2 *
Contrast ratio: 1400:1 typical (refer to Factory Calibration Report for unit specifications).

** Extended operation allows the peak white output to reach 500 cd/m^2 (consult CRS for details).*
If the peak white output is set above 350 cd/m^2, then the ambient operating temperature should be restricted to 0°C to +25°C. Note that the case temperature will be elevated in this configuration.

Monitor Dimensions

Dimensions: 800 x 490 x 70 mm.
Weight: 18.65 kg.

Stand Dimensions

Dimensions: 400 x 480 x 230 mm.
Weight: 3.55 kg.

Power Supply

Mains Input: 100-240 V AC, ~1.8 A, 50/60 Hz.
Output: 12 V DC, 12.5 A, 165 W.

Touchscreen Characteristics

Spatial accuracy: 3-4 mm.
Timing: 22 ms resolution.
Optical clarity: near 100 % optical transmission.
Touch methods: bare or gloved finger or any solid object, no pressure required.

Appendix A

A Status LED is located on the back panel. It indicates the status of the Display++ system:

LED colour/behaviour	**Interpretation**
Blue	Display++ is working normally and is displaying DVI video from the connected Host Computer.
Flashing blue.	Self test in progress.
Green	Display++ is working, but no valid video DVI signal is detected. Check DVI cable and signal source.
Flashing Green	Standby Mode. LED backlight off.
Flashing Red	Display++ is updating firmware or persistent configuration information. Do not interrupt the power while the LED is flashing red.
Solid Red	A problem has been detected and Display++ cannot start. This usually occurs if the SD Card is missing or not inserted correctly in the slot in the rear of the monitor.
Orange	BOLDscreen 32 main firmware is absent. This happens because a bootloader update has been performed. Perform firmware update to restore operation.
Flashing Orange	Firmware update of second processor in progress.
Flashing Purple	Error in config.xml file (check for inappropriate UTF-8 “” characters).
Solid Purple	Internal LED backlight communication problem; consult Cambridge Research Systems.

Revision History

Document version	Firmware version	Document change
R01	2088	First release
R02	2090	Added CDC command $LuminanceSetPoint
R03	2096	Edited the Installation procedure
R04	2096	Add blank page 2
R05	2150	Mention different luminance scaling factors
R06	2150	Mention different luminance scaling factors
R07	2154	Correct typo for colour correction CDC command
R08	2156D	Information about Status LED added
R09	2169	Amend system environmental conditions to include 500 cd/m2. Documented flashing blue self test
R10	2188A	Changed video input to HDMI and added FPR layer to list of features (optional)

Contact

Cambridge Research Systems Ltd

Telephone: +44 1634 720707

Website: www.crsltd.com

80 Riverside, Sir Thomas Longley Road

Rochester

Kent ME2 4BH

United Kingdom