Optical Finesse LLC

μLC200-12 Two-Channel USB Liquid-Crystal Controller

User’s Manual

Manual revision
June 2017

Optical Finesse LLC
3133 Indian Road
Boulder CO USA 80301
(303) 442-1549
www.opticalfinesse.com

Made in USA

Table of Contents

1. Description & Specifications

The μLC200-12 is a versatile, low-cost, dual-channel liquid-crystal controller, designed to run continuously-variable LC devices such as variable retarders, polarization switches, shutters, attenuators, and tunable color filters.

The μLC200-12 is a USB bus-powered device with two independent LC channels, each with 12 V_{RMS} of range. The channels are capable of operating either AC (carrier) or DC (carrier-less), and can drive cells up to 600 nF of capacitance. The μLC200-12 is expressly designed to be a "mini" LC-Tec LCC-230.

The controller is operated by the *LCDriver2* application via a full-speed USB 2.0 compliant interface. *LCDriver2* permits dynamic editing of AC or DC programs up to 96 lines in length. A separate dialog box of the *LCDriver2* application permits simple amplitude control of the LC channels in AC mode.

Technical Specifications

LC CHANNELS	
Number of LC channels	Two, independently programmable
Maximum voltage range	12.10 V_{RMS}
LC voltage output	Either AC (carrier + envelope) or DC (carrier-suppressed, sign reversal after each iteration) LC drive waveforms, software selectable. LC common (analog ground) is isolated from input USB power ground
AC waveforms	Square-wave carrier, 50.00% duty cycle. Frequencies of 1 Hz to 2.0 kHz, software-selectable
DC waveforms	Bipolar, 0.10 ms minimum time interval, 0.01 ms interval resolution
Amplitude resolution	14-bit; 1 mV voltage resolution
Amplitude accuracy	Better than ±0.3% typical, ±0.5% maximum, measured at full range
Residual DC	Less than 3 mV at any output
Drive capability	Each channel can drive cells up to 600 nF capacitance
Short circuit	Full-recovery short-circuit protection on LC outputs. Interior “Fault” LED will illuminate when an LC output is shorted
LC programs	Up to nine pre-stored 96-line LC programs stored in onboard EEPROM (Electrically Erasable Programmable Read-Only Memory) for ‘standalone’ operation (no *LCDriver2* running). *LCDriver2* selection determines executing standalone program

INTERFACE	
Host computer interface	Full-speed (12 Mb/s) USB 2.0 compliant, bus-powered device, 200 mA typical bus current draw (current draw will depend on LC load & driving scheme). USB power ground is isolated from LC commons (analog ground)
Software	*LCDriver2* application software with host USB drivers for Windows 7, Vista, Windows XP, and Windows 2000 (executable only; 64-bit Windows XP not supported). Permits dynamic creation and editing of LC programs. Separate dialog box for static LC channel amplitude control in AC mode
Device drivers	Standard .DLL device-driver toolkit included with *LCDriver2* installation (requires *LabVIEW* runtime engine). Advanced toolkit for programming with *Visual Studio* to generate 32- or 64-bit applications available upon special arrangement
External Trigger	Opto-isolated Trigger Input, rising-edge triggered, TTL-compatible logic levels and fan-in. Optional trigger output capability upon special arrangement

PHYSICAL	
Dimensions	4.4 in. L *x* 2.5 in. W *x* 1.1 in. H (11.2 cm L *x* 6.4 cm W *x* 2.8 cm H)
Weight	0.3 lbs (0.1 kg)
Chassis material	Blue translucent ABS plastic, white silk-screened nomenclature
Chassis features	• Internal "USB Power", "Fault" and "Status" LED's • Series B USB receptacle • Mini-DIN6 connector for LC outputs & Trigger Input
RoHS compliance	100% compliant
CE compliance	Certification to FCC Class A emissions level upon special arrangement
Operating temperature	5 to 45 ^{o}C
Warranty	One year

In keeping with our commitment to continuous product improvement, these specifications are subject to change without notice

2. Safety Considerations

The following safety and maintenance considerations should be observed:

- The controller is for indoor use only, and not to be used in wet or moisture-laden environments. The controller should only be operated in relative humidities between 20-80% non-condensing, and at temperatures between 5 to 45 ^{o}C.

- The controller is powered by the nominal 5VDC power supplied by USB. There is no external ON/OFF switch. The power disconnect means for the controller is the USB connection located on one face of the controller.

- The LC output cable should be jacketed, with a 150 V minimum insulation rating. The mini-DIN6 cable supplied with the controller meets this specification.

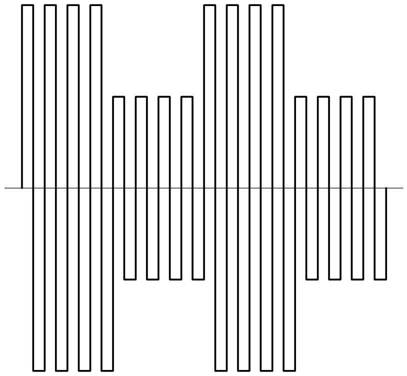
- The mini-DIN6 cable length is recommended to be 5 ft (1.5 m). To avoid conducted RF immunity problems, under no circumstances should the cable harness be longer than 3 m.

- The interior of the controller is not designed to be user accessible, and there are no user-serviceable parts inside. Contact Optical Finesse directly and return the controller if maintenance or calibration is required.

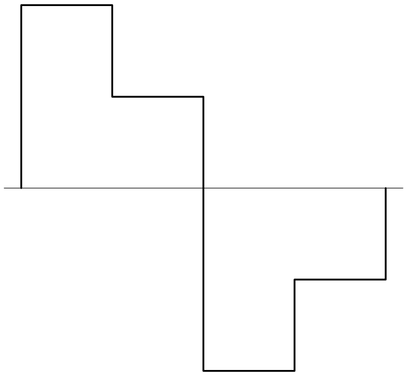
- Clean the exterior of the unit with a soft dry cloth only.

3. Overview

The μLC200-12 is designed to output voltage waveforms tailored for use in liquid crystal (LC) drive applications. There are two basic types of waveforms. AC waveforms amplitude-modulate a faster bipolar square-wave carrier, typically of frequency 1 kHz or slower. DC waveforms suppress the carrier and apply just the slowly modulating envelope to the cells. Because of hardware implementation, it is required that both controller channels operate in the same mode.



AC waveform



DC waveform

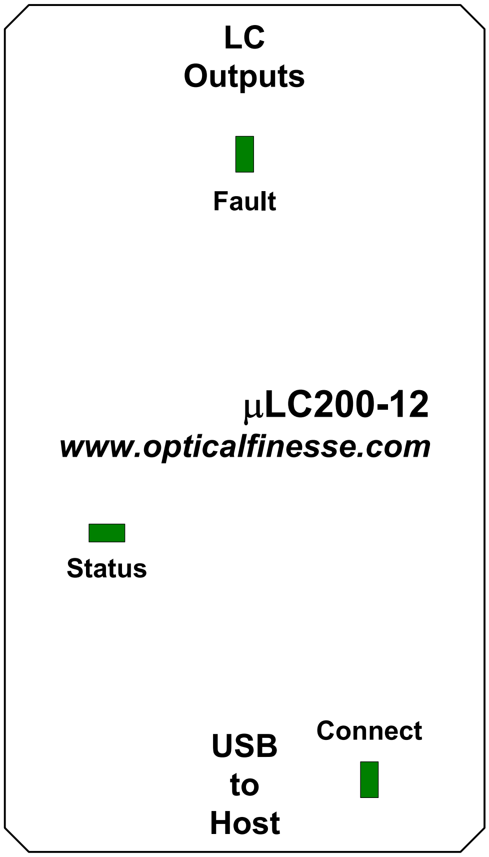
For either AC or DC mode, a fundamental LC driving requirement is DC balance, which requires that the time integral of any given number of carrier periods always yield zero, so that there is no residual DC offset applied to the cells. DC balance means that an LC cell always responds to the root-mean-square (RMS) average of the drive voltage. DC balance prevents slow ion migration to the indium tin oxide (ITO) electrode surfaces, which can damage or destroy the cell. For AC mode, this requirement is met by applying amplitude modulation on a cycle-count basis. For DC mode, DC balance is maintained by the controller automatically changing the sign of the envelope after every cycle.

The μLC200-12 is designed to output voltages in the form of LC programs. Each line of a program consists of one or two voltages, and a time duration for which these voltages are held. These voltages specify either the AC-mode amplitude envelope, or the DC-mode bipolar voltage whose polarity will change after each program iteration. The lines of the program can be advanced (depending on chosen mode) by an external trigger. The controller accepts an external trigger from an outside source through two pins on the mini-DIN6 connector. The external trigger source can be any TTL-compatible logic-level source. Triggers are generated by rising edges (duty cycle does not matter).

The μLC200-12 can run in either host or 'standalone' mode. In host mode, the controller is connected by USB to a host computer running the *LCDriver2* application, and the host can run or dynamically edit executing LC programs through *LCDriver2*. In standalone mode, the controller is still powered by USB, but *LCDriver2* is not running. A program may be pre-selected to run once *LCDriver2* exits. The controller executes LC programs that have been previously stored in EEPROM onboard the controller.

4. Chassis Features

A brief description of the controller chassis indicators & connections follows.



- USB to Host: Series B receptacle for full-speed USB 2.0 compatible connection to host computer. The controller is powered through the 5VDC present on USB.

There are three green LED's interior to the controller body, visible through the blue plastic translucent top & sides of the chassis. These LED's are:

- Connect LED: lit when controller is connected to USB and properly powered. In normal operation, this will be the only LED illuminated.

- Status LED: in normal operation, this LED is off, for either host or 'standalone' mode. Upon initial connection to the host by USB, this LED will blink slowly until the μLC200-12 is fully configured by the host system software. If the host is unable to configure the controller, the Status LED will continue to blink; this most likely means that the host is unable to accommodate the USB power requirements indicated in the configuration descriptor for its analog voltage supplies. In OLD_STANDALONE mode (see Sec. 6.b below), this LED will be steady-ON.

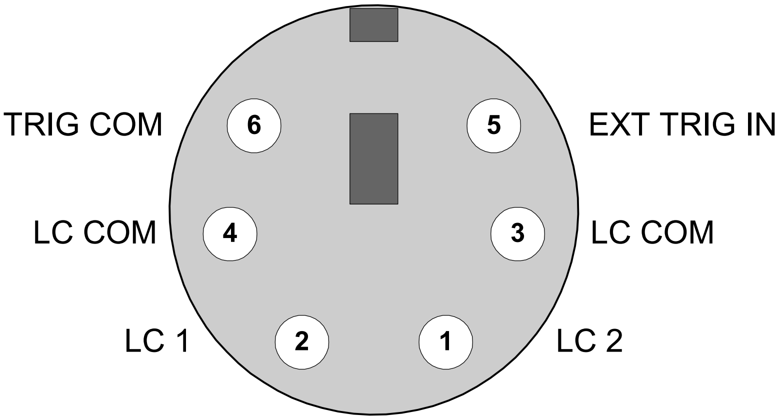
- Fault LED: In normal operation, this LED is off. It will be lit when a short-circuit or other fault condition exists on an LC output. When the fault is removed, this LED will automatically turn off. The Fault LED will also be illuminated if a fault condition exists in the controller's internal analog voltage supplies.

- LC Outputs: mini-DIN6 receptacle that supplies two LC voltage output connections (LC1 and LC 2), along with two common (LC COM) connections. The LC COM connections are tied to the analog ground plane of the controller's PCB, and are isolated from the input USB power ground. The pin-out for this connector is shown below. The connector case is also tied to the analog ground plane of the controller PCB.

  Note that pins 1-4 of this connector are identical to the pin-outs of the LC-Tec LCC-230.

  This receptacle also has two pins (EXT TRIG IN and TRIG COM) for triggering from an external TTL-compatible trigger source. These trigger pins are opto-isolated from the LC outputs and LC commons. Do *not* tie the TRIG COM pin to either of the LC COM pins.

  The controller is designed to be rising-edge triggered. If a function generator is used as a trigger source, do *not* apply its analog output to this connector—use the sync or logic output instead, or set up the output for TTL.



Pin-out of female LC Output mini-DIN6 receptacle (looking at face of connector)

5. Getting Started

Prior to using the μLC200-12, the *LCDriver2* application must be pre-installed on the host computer to be used with the controller. This application and the associated USB drivers are available as a monolithic .ZIP file downloadable from the Internet. Please consult with Optical Finesse for the appropriate URL for your specific controller.

The installation process will extract information from the .ZIP file and create an installation subdirectory. After successful installation, both the downloaded .ZIP file and the installation subdirectory can be deleted. The installation process depends on the host computer operating system, as detailed below. Directions are also given for updating a pre-installed *LCDriver2* version with the most recent version.

Prior to software installation, unpack the μLC200-12 and supplied USB cable. As the controller is powered by USB bus, do not plug it into the host until instructed to do so.

The μLC200-12 is shipped with pre-installed programs on its EEPROM useful for driving various LC devices. These programs may be read and modified by *LCDriver2*.

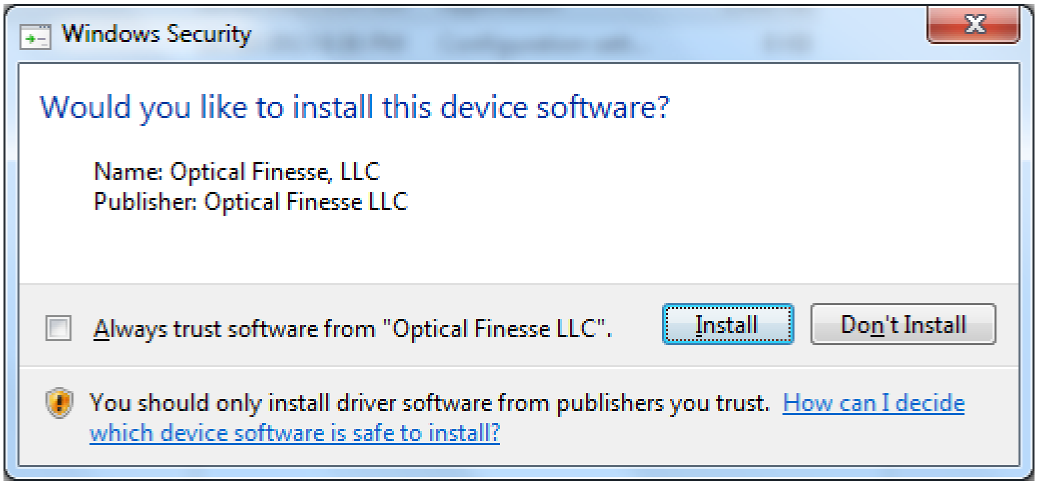
Note: if *LCDriver2* and the associated USB device drivers have already been installed successfully for a LC-Tec LCC-230, the installation .ZIP file need not be unzipped, and `setup.exe` does not need to be re-run.

However, upon first connection of the μLC200-12 to USB, a device-driver system prompt will still appear, even though the LCC-230 and μLC200-12 use the same device drivers. This is because the two controllers have different USB Device Instance ID's. For Windows 7 installs, follow from step 8) in Sec. 5.a below; for Windows XP installs, follow from step 7) in Sec. 5.b below.

LCC-230's and μLC200-12's may coexist on USB and can be accessed concurrently from *LCDriver2*. See Sec. 7.a for details.

a. New installs on Windows 7 & Vista systems

1. Download and save the installation .ZIP file to your computer, using the URL provided by Optical Finesse.
2. In Windows Explorer, right-click on the downloaded .ZIP file and select Extract All… to unzip the file into an installation subdirectory.
3. Locate `setup.exe` in the root of the installation subdirectory. Double-click on `setup` to start the install process.
4. Accept the default location of `C:\Program Files (x86)\LCDriver2\` for the *LCDriver2* program (for 32-bit Windows 7 & Vista, this location will be `C:\Program Files\LCDriver2\`).
5. Accept the National Instruments license agreement.
6. The USB device drivers will be pre-installed ("Installing Device Driver…"), and a Windows Security prompt will appear, as shown below. Select "Install" to complete the device software pre-install.



Would you like to install this device software?

Name: Optical Finesse, LLC
Publisher: Optical Finesse LLC

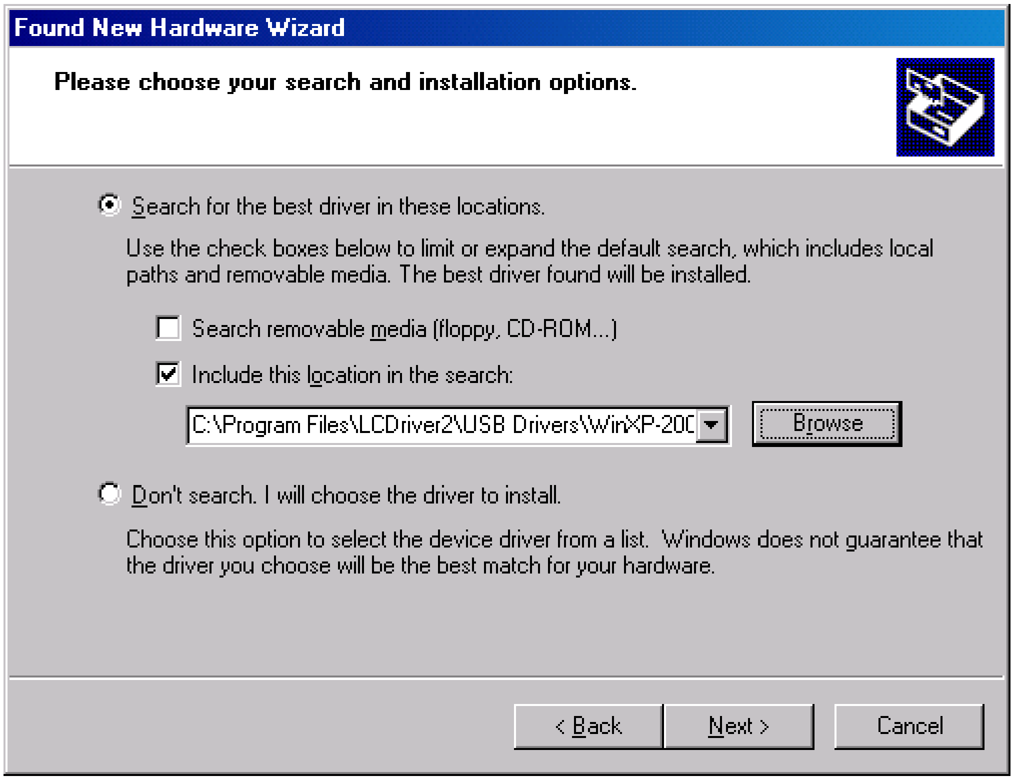
□ Always trust software from "Optical Finesse LLC".

You should only install driver software from publishers you trust. How can I decide which device software is safe to install?

7. At installation end, a prompt may appear indicating that you must restart your computer to complete the install. If so, choose "Restart now".
8. Plug the supplied USB cable into the controller's USB receptacle, and plug the other end into one of the host computer's USB ports. After a brief pause, the interior green Connect LED should go ON. The interior Status LED will be slowly blinking.
9. The device driver software install will be completed, and a balloon will appear onscreen informing you that the install was successful. The Status LED will go OFF. Start *LCDriver2* from Window's Start menu: Start > All Programs > LCDriver2 > LCDriver2. The "Dynamic Connection" annunciator should appear in the application window's upper right-hand corner.

b. New installs on Windows XP & Windows 2000 systems

1. Download and save the installation .ZIP file to your computer, using the URL provided by Optical Finesse.
2. In Windows Explorer, right-click on the downloaded .ZIP file and select Extract All… to unzip the file into an installation subdirectory.
3. Locate `setup.exe` in the root of the installation subdirectory. Double-click on `setup` to start the install process.
4. Accept the default location of `C:\Program Files\LCDriver2\` for the *LCDriver2* program.
5. Accept the National Instruments license agreement.
6. At installation end, a prompt may appear indicating that you must restart your computer to complete the install. If so, choose "Restart now".
7. Plug the supplied USB cable into the controller's USB receptacle, and plug the other end into one of the host computer's USB ports. After a brief pause, the interior green Connect LED should go ON. The interior Status LED will be slowly blinking.
8. The Found New Hardware Wizard will appear onscreen. If a Wizard dialog appears asking to connect to Windows Update, choose "No, not this time".
9. Select "Install from a list or specific location (Advanced)" in the Wizard to install the USB device drivers.
10. Select "Search for the best driver in these locations/Include this location in the search" in the Wizard.
11. Using the Browse button, navigate to the subdirectory: `C:\Program Files\LCDriver2\USB Drivers\WinXP-2000\`.



Found New Hardware Wizard

Please choose your search and installation options.

● Search for the best driver in these locations.

Use the check boxes below to limit or expand the default search, which includes local paths and removable media. The best driver found will be installed.

□ Search removable media (floppy, CD-ROM...)

☑ Include this location in the search:

C:\Program Files\LCDriver2\USB Drivers\WinXP-20C

○ Don't search. I will choose the driver to install.

Choose this option to select the device driver from a list. Windows does not guarantee that the driver you choose will be the best match for your hardware.

12. The USB device driver software install will be completed, and a balloon will appear onscreen informing you the install was successful. The interior Status LED will go OFF.
13. Start *LCDriver2* from Window's Start menu: Start > All Programs > LCDriver2 > LCDriver2. The "Dynamic Connection" annunciator should appear in the application window's upper right-hand corner.

c. Updating *LCDriver2* on an existing install

1. If the previous version of *LCDriver2* is running, exit it.
2. Go to Windows Control Panel > Add or Remove Programs. Locate *LCDriver2* in the list, and click on the Remove button to uninstall. Close Control Panel.
3. Download and save the installation .ZIP file to your computer, using the URL provided by Optical Finesse.
4. In Windows Explorer, right-click on the downloaded .ZIP file and select Extract All… to unzip the file into an installation subdirectory.
5. Locate `setup.exe` in the root of the installation subdirectory. Double-click on `setup` to start the install process.
6. Accept the default location for the *LCDriver2* program.
7. Accept the National Instruments license agreement.
8. At installation end, a prompt may appear indicating that you must restart your computer to complete the install. If so, choose "Restart now".
9. The USB device drivers do not need to be reinstalled for an update of *LCDriver2*.

d. Errata

End-to-end communication between *LCDriver2* and the µLC200-12 can always be confirmed by the Upload/Download > Blink Controller LED menu option (see Sec. 7.c below). This will momentarily blink the green Status LED interior to the controller body. We recommend performing this verification after an *LCDriver2* install.

As discussed in Sec. 7 below, *LCDriver2* permits saving LC programs to the host computer's hard drive. For Windows XP & 2000 operating systems, the default location *LCDriver2* will use for these programs is:
`\My Documents\LCDriver2_Data\programs`.

For Windows 7 & Vista operating systems, the default location *LCDriver2* will use for these programs is:
`Libraries > Documents > My Documents >LCDriver2_Data > programs.`

A standard *LabVIEW*-based .DLL device-driver toolkit is available for interfacing the µLC200-12 to a user-supplied application. Please contact Optical Finesse for further technical and support information on the application programming interface for this toolkit.

The device-driver toolkit is installed during the *LCDriver2* installation process described above. Programmers who plan to use the toolkit to develop their own applications that incorporate the µLC200-12 should first install the *LCDriver2* package, as the toolkit's .DLL requires the *LabVIEW* runtime engine, which is installed with *LCDriver2*. After successful installation, programmers should copy `LCDriverUsb.dll` from `\LCDriver2` and `ulc_dd.dll` from `\LCDriver2\Toolkit` to the local subdirectory of their application.

!Do NOT use the version of `LCDriverUsb.dll` found in `\LCDriver2\Toolkit`! The version of `LCDriverUsb.dll` installed in `\LCDriver2` will be the correct one for the host operating system; the `.dll` in `\LCDriver2\Toolkit` is an older version that must remain in this subdirectory due to a *LabVIEW* installer artifact.

The files `LCDriverUsb.dll` and `ulc_dd.dll` are designed to support user programs that will be built as 32-bit applications and run in a 32-bit or 64-bit environment. This is the usual case for most applications. Please contact Optical Finesse for special arrangements for an advanced *Visual Studio*-based toolkit to support programs that must be built as 64-bit applications and run in a 64-bit environment.

6. LC Program Details

Every LC program is either AC-mode or DC-mode, and this choice must hold for both LC channels in the program. LC programs can be written for one or two channels. A one-channel program must use LC_1, a two-channel program uses LC_1 and LC_2. For a one-channel program, LC_2 is set to zero volts by the controller.

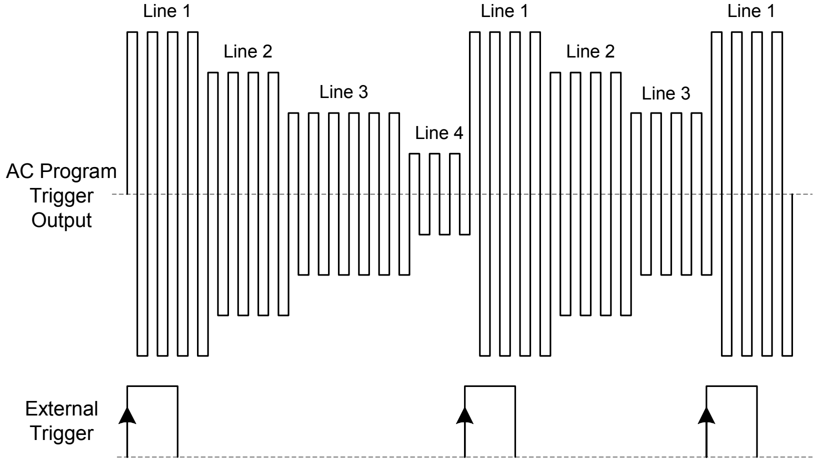
Every LC program consists of one or more lines of voltage settings for the active channels. Programs may be up to 96 lines in length. Each line is held for a time duration that is treated differently depending on whether the program is in AC or DC operating mode. For AC programs, the time duration must be an integral number of full periods of the chosen carrier frequency. For example, for an AC program running at 1 kHz, the time duration for a line must be an integral multiple of 1.00 ms. For DC programs, the time duration must be a minimum of 0.10 ms, and is settable in 0.01 ms increments from this minimum value (0.10, 0.11, 0.12, 0.13, etc.). *LCDriver2* will validate and appropriately round user entries in the "t (ms)" column to enforce these constraints.

a. Trigger Modes

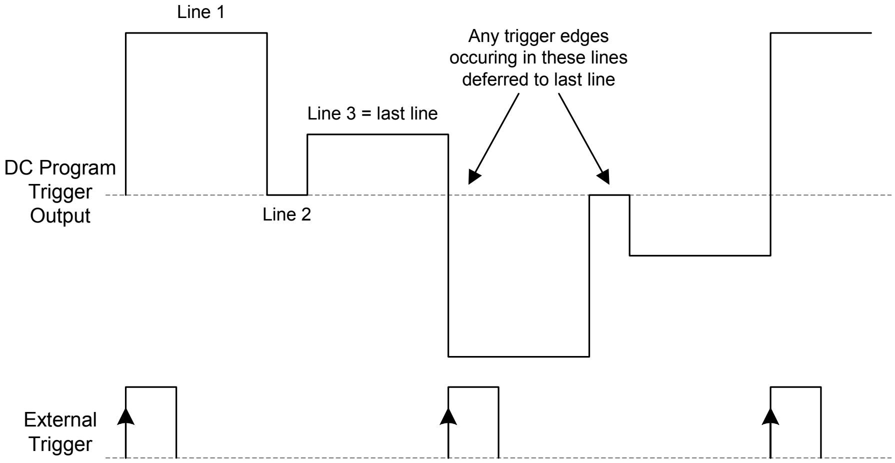
The programs have three input trigger modes, called Internal Trigger, Program Trigger, and Line Trigger. DC Line Trigger mode requires *LCDriver2* V. 12 or greater.

For Internal Trigger, execution starts at the first line, and the lines are executed sequentially, using the set duration in the time field. Input trigger edges are ignored on the mini-DIN6 connector. At the last line of the program, execution loops back to the first line and continues indefinitely. Internal trigger behavior for AC and DC programs is the same except that for DC programs, the polarity of the line voltages is toggled on every program iteration.

Program Trigger mode, in the absence of any external trigger edges, behaves just like Internal trigger. However, for Program trigger, each new rising external trigger edge restarts the program at the first line. The duty cycle of the trigger waveform is not relevant. AC programs can be re-triggered during any program line, as shown below.

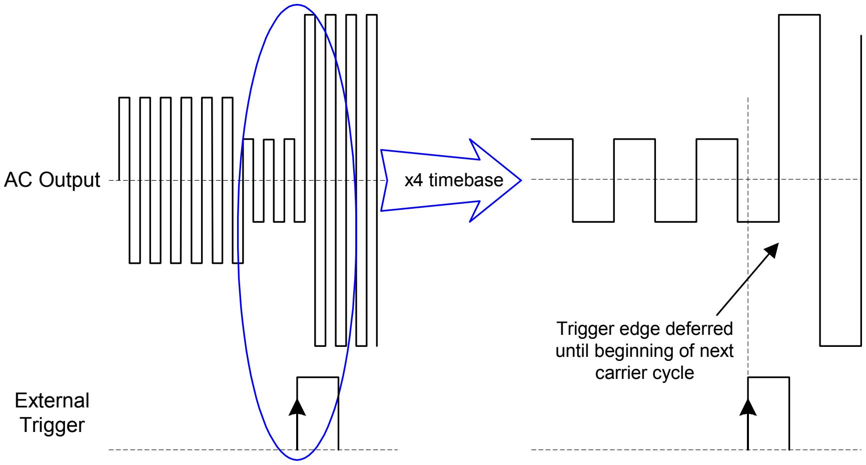


For DC programs, triggers occurring before the last line of the program are deferred until the last line is reached, and each external trigger toggles the polarity of the line voltages, as shown below. DC Program trigger is typically used to 'sync' (phase lock) the μLC200-12 to an external periodic source. For these applications, the last line's duration should be "padded" by a couple ms to insure that the trigger edge will always fall within the last program line.


Line 1
Any trigger edges occuring in these lines deferred to last line
Line 3 = last line
DC Program Trigger Output
Line 2
External Trigger

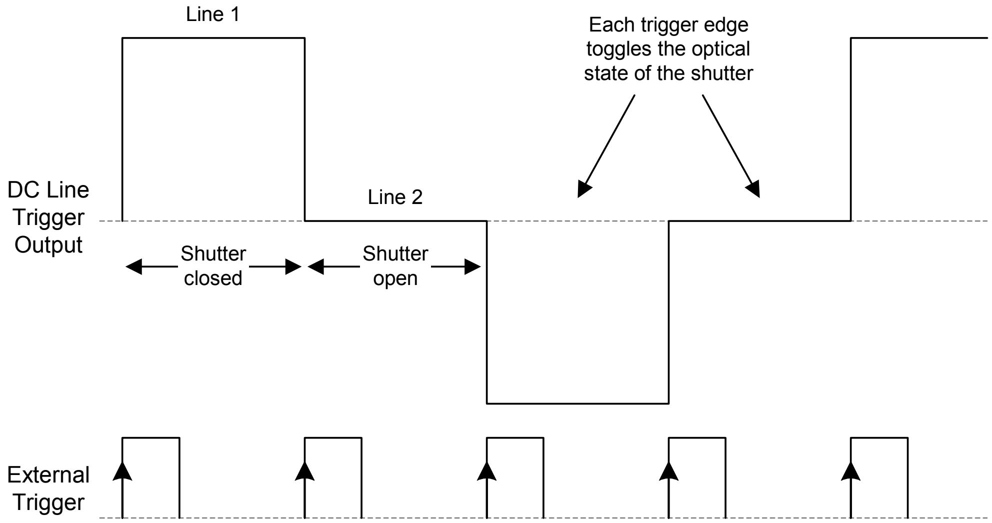
AC Line Trigger mode is supported in all versions of *LCDriver2*. In AC Line Trigger, the time durations for each line are ignored, and the voltages of the lines of the program are held indefinitely until a trigger edge appears on the mini-DIN6 connector. A rising trigger edge advances the program to the next line. At the last line of the program, execution wraps back to the first line.

Note that for AC Program and Line Trigger modes, trigger action is always deferred until the next carrier cycle to maintain DC balance, as illustrated below.



LCDriver2 V. 12 and up supports DC Line Trigger mode; this trigger mode is inaccessible for older host-side software versions. In DC Line Trigger mode, time durations for all but the first line of the program are set to a maximum value. Each rising trigger edge advances the program to the next line. At the last line of the program, execution wraps back to the first line with polarity of all line voltages toggled.

DC Line Trigger is especially useful for 'synching' to an arbitrary frequency to drive a LC shutter with a perfectly symmetric 50% open/closed duty cycle. Shown below is a two-line DC Line Trigger program for a normally-white shutter device.



b. Host & Standalone

The μLC200-12 is entirely powered by the USB root hub, so does not have true 'standalone' modes. Upon being initially enumerated by the host, the controller will go into OLD_HOST mode and accept messages from a host computer application. The host application may be either *LCDriver2* or a user-supplied application that accesses the controller functionality through the .DLL device-driver toolkit. Prior to *LCDriver2* being run, the LC channels will run an "idle" condition of AC carrier on both channels. The AC idle carrier frequency and voltage amplitudes can be set by host software.

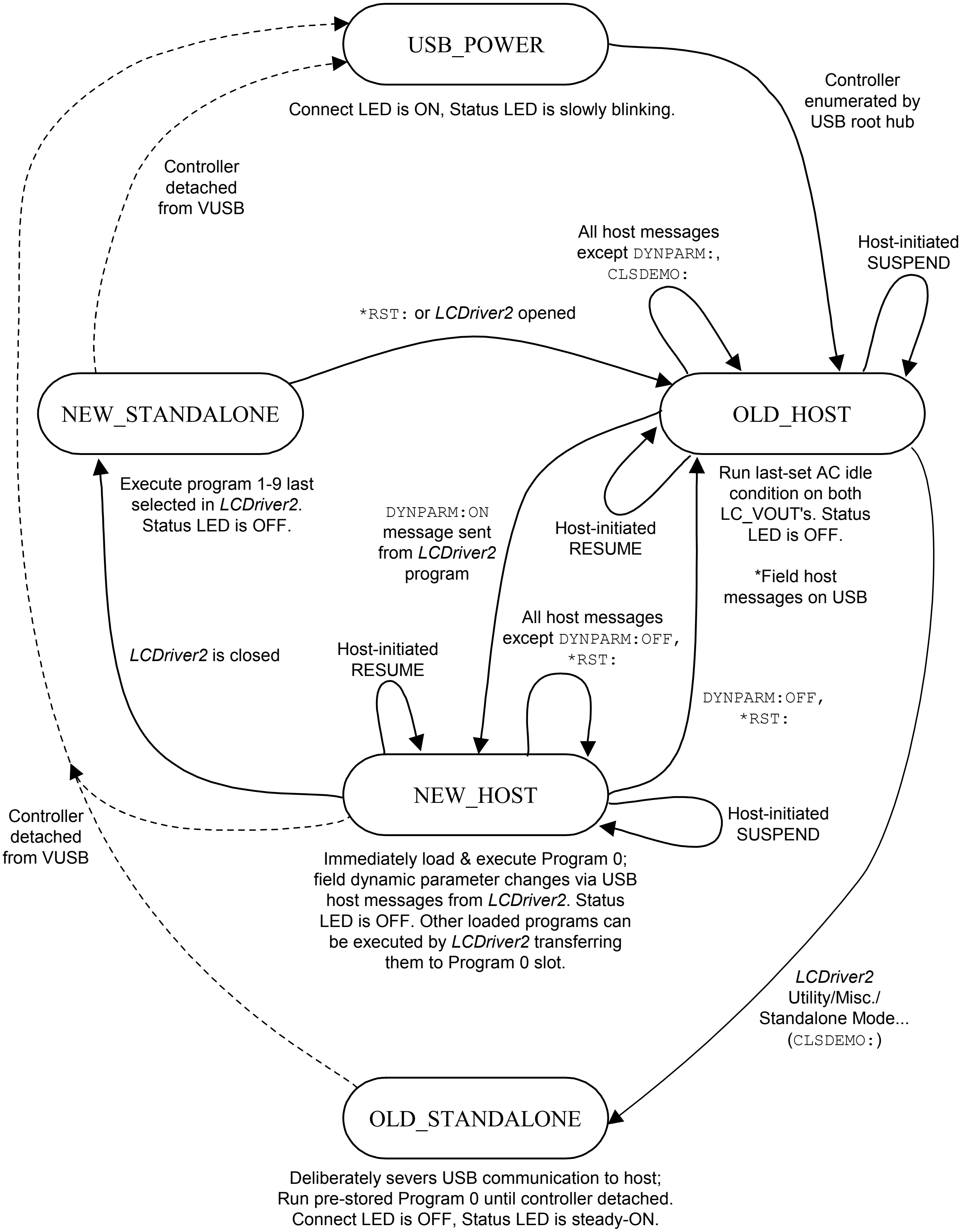
The *LCDriver2* application will operate the controller in the NEW_HOST state (dynamic host mode), in which a LC program is always executing. The controller is normally in the OLD_HOST state when the .DLL toolkit is being used with a user-supplied application.

If *LCDriver2* is closed, the controller will transition to the NEW_STANDALONE state and continue to execute whichever program was last pre-selected in *LCDriver2*. Note this is not a true standalone state as the controller must remain attached to USB in order to be powered.

In the OLD_HOST, NEW_HOST, and NEW_STANDALONE states, the Status LED will be OFF.

The OLD_STANDALONE state is a special state that is only accessible by a command issued by *LCDriver2*. In OLD_STANDALONE, the controller deliberately severs USB communication with the host, while still drawing power from the bus. This state is useful for setups in which the controller must always execute pre-set program 0, but otherwise not been "seen" by the host. In this state, the Connect LED will go off, and the Status LED will be steady-ON.

A more detailed state diagram is shown on the following page.


USB_POWER
Controller enumerated by USB root hub
Controller detached from VUSB
All host messages except DYNPARM:, CLSDEMO:
Host-initiated SUSPEND
*RST: or LCDriver2 opened
NEW_STANDALONE
OLD_HOST
DYNPARM:ON message sent from LCDriver2 program
Host-initiated RESUME
*Field host messages on USB
All host messages except DYNPARM:OFF, *RST:
LCDriver2 is closed
Host-initiated RESUME
DYNPARM:OFF, *RST:
NEW_HOST
Host-initiated SUSPEND
Controller detached from VUSB
LCDriver2 Utility/Misc./ Standalone Mode... (CLSDEMO:)
OLD_STANDALONE

Connect LED is ON, Status LED is slowly blinking.

Execute program 1-9 last selected in *LCDriver2*. Status LED is OFF.

Run last-set AC idle condition on both LC_VOUT's. Status LED is OFF.

Immediately load & execute Program 0; field dynamic parameter changes via USB host messages from *LCDriver2*. Status LED is OFF. Other loaded programs can be executed by *LCDriver2* transferring them to Program 0 slot.

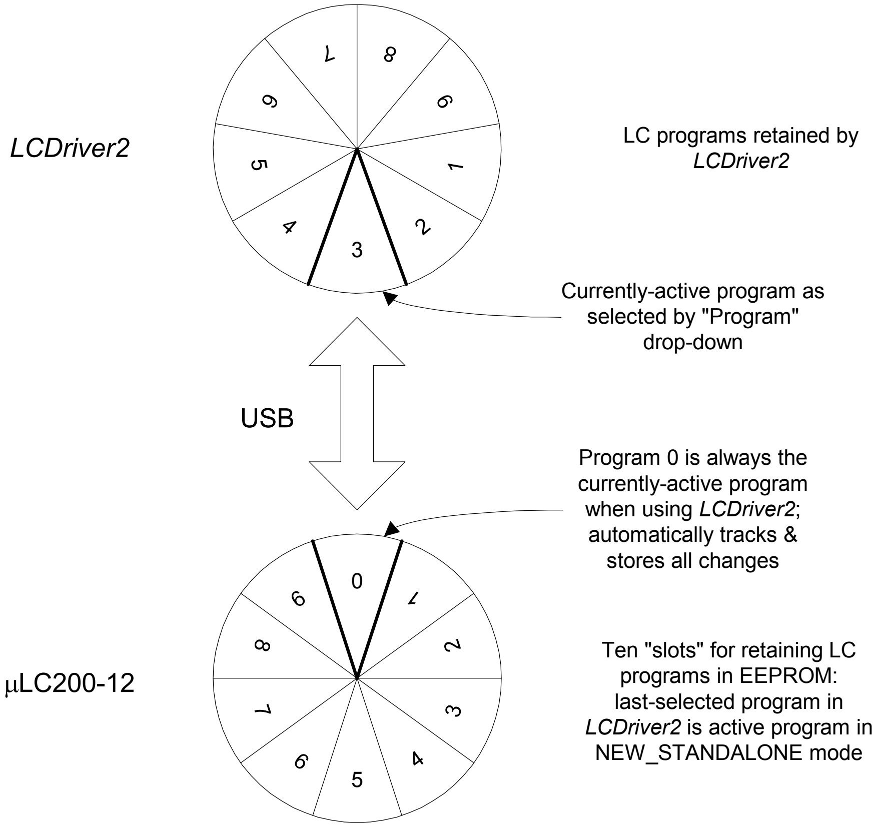
Deliberately severs USB communication to host; Run pre-stored Program 0 until controller detached. Connect LED is OFF, Status LED is steady-ON.

c. Program Upload/Download

Up to ten LC programs can be pre-stored on the controller's onboard EEPROM. Programs 1 through 9 are user-accessible "slots" to store or retrieve programs. When a controller is first attached to USB with *LCDriver2* running, *LCDriver2* will automatically read all pre-stored programs on the controller and retain a copy of each program in its own memory for later editing. When using *LCDriver2*, it is important to understand and maintain the distinction between the LC programs retained by *LCDriver2* (stored in the host computer's memory) and the LC programs retained by the µLC200-12 (stored in the controller's onboard EEPROM).

Program 0 has a special role. Program 0 on the controller tracks and stores in onboard EEPROM whatever program changes are made in *LCDriver2*, even if these changes are not explicitly uploaded to the controller. In this sense, Program 0 is a special "scratchpad" program to which all edits are made and for which the last state is always saved automatically.

In dynamic host mode (*LCDriver2* running, NEW_HOST state), the controller is always executing Program 0, and any parameter changes made to Program 0 are immediately observable in the executing LC program. In *LCDriver2*, the currently-active program is always implicitly Program 0. All LC programs are composited in the Program 0 slot, and then transferred to other program slots by *LCDriver2*. Likewise, to edit a previously-saved program in another slot, it must first be transferred to the Program 0 slot by *LCDriver2* and then dynamically edited.



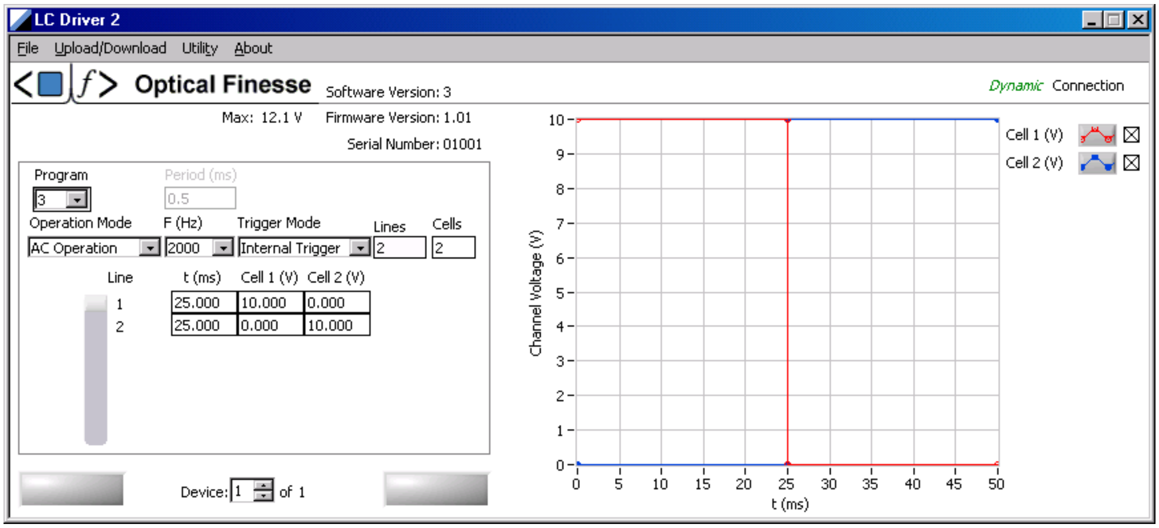
LCDriver2
7
8
9
1
2
3
4
5
6
LC programs retained by *LCDriver2*
Currently-active program as selected by "Program" drop-down
USB
Program 0 is always the currently-active program when using *LCDriver2*; automatically tracks & stores all changes
µLC200-12
0
1
2
3
4
5
6
7
8
9
Ten "slots" for retaining LC programs in EEPROM: last-selected program in *LCDriver2* is active program in NEW_STANDALONE mode

7. LCDriver2 Application Details

This section explains the functionality of the *LCDriver2* application supplied with the μLC200-12. *LCDriver2* can be used to read, edit, and store LC programs on controllers.

a. Overview

The main application window of *LCDriver2* is shown below, as it appears when connected to a controller. The trade-dress of the application may be different for your version of *LCDriver2* (e.g. different logos), but the functionality is the same.



LC Driver 2

Optical Finesse Software Version: 3 *Dynamic* Connection

Max: 12.1 V Firmware Version: 1.01

Serial Number: 01001

Program: 3 Period (ms): 0.5

Operation Mode: AC Operation F (Hz): 2000 Trigger Mode: Internal Trigger Lines: 2 Cells: 2

Line	t (ms)	Cell 1 (V)	Cell 2 (V)
1	25.000	10.000	0.000
2	25.000	0.000	10.000

Device: 1 of 1

The software version of *LCDriver2* is displayed at the application window center; below is the embedded software (firmware) version of the active μLC200-12 controller, and its serial number.

The box in the left portion of the window displays the edit controls for the currently-active LC program. To the right is a graphical representation of one cycle of the program. There is a trace for each LC cell in the program; display of the traces is controlled by the checkboxes adjacent to the trace.

To the right is the "Dynamic Connection" annunciator, which is always displayed when *LCDriver2* has a good USB connection to a controller. If the controller is detached from USB, or if there is a USB communications problem, this annunciator will change to "No Connection". *LCDriver2* supports multiple instances of μLC200-12's on USB. The currently-active controller is selected by the "Device" control at window bottom. This control also shows how many μLC200-12's are presently on USB.

When *LCDriver2* is first connected to a controller, it automatically reads all pre-stored programs on the controller, and will then make Program 1 the currently-active program. The "Working…" and "USB Com" buttons at window bottom will be briefly illuminated during this process. The "USB Com" button will be illuminated orange whenever there are messages sent between *LCDriver2* and the controller.

All edit controls affect the currently-active LC program. The currently-active program is selected by the "Program" drop-down control. As explained in Sec. 6.c above, Program 0 is a special LC program and thus is grayed-out in this drop-down (Program 0 is in fact always the currently-active program). If a selected controller program happens to be blank, *LCDriver2* will display "Blank Program" in red adjacent to the "Program" drop-down control. Blank programs are always one-line, two-cell AC Internal Trigger programs, with the carrier frequency and cell voltages set to the currently-set AC idle frequency and amplitudes (see Upload/Download/Set AC Idle… below).

The edit controls will change depending on the selection of "Operation Mode". For AC operation, a "F (Hz)" drop-down control will be displayed, with a corresponding "Period (ms)" above it. For DC operation, no frequency or period controls will appear, but an additional "+/-" column will be displayed. This column may be used to control the polarity (sign) of DC program lines. For AC operation in "Line Trigger" mode, the time intervals in the matrix are ignored; a reminder about AC Line Trigger time intervals is displayed in red in *LCDriver2* for this mode.

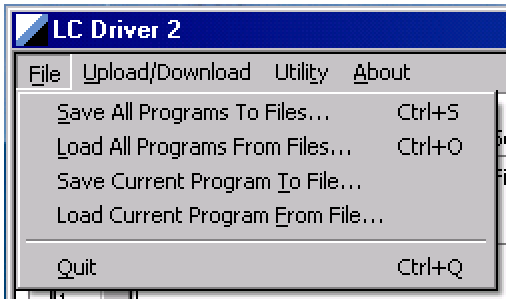
DC Line Trigger operation requires *LCDriver2* V. 12 and up. *LCDriver2* will display reminders in red about the special program constraints associated with DC Line Trigger when this mode is chosen from the drop-down.

The size of the currently-active program matrix is set by the "Lines" and "Cells" edit boxes. The number of lines may be from 1 to 96, and the number of cells either 1 or 2. If additional lines are inserted into a program by incrementing the "Lines" control, the time value will be initially set to 2.00 ms. The voltage values of the new lines will initially be duplicates of the last line of the program. If an additional cell is inserted into a program by incrementing the "Cells" control, the new cell voltages will initially be set to 0.000V.

The Tab key may used to cycle through and edit the program matrix. Cell voltages can be set to mV precision, up to the "Max" voltage displayed above the edit control box. For μLC200-12 controllers, the maximum settable voltage is 12.100V. Permitted values in the "t (ms)" column depend on operating mode. For AC programs, the time duration must be an integral number of carrier periods as shown in the "Period (ms)" display. For DC programs, the time duration must be a minimum of 0.10 ms, and is settable in 0.01 ms increments from this minimum value (0.10, 0.11, 0.12, 0.13…). *LCDriver2* will validate and appropriately round user entries to enforce these constraints.

There are additional special constraints for DC Line Trigger. Programs in this mode must have two or more lines, and the time durations for lines 2 and up are set to a signature maximum value of 9999.990 ms. Durations for line 1 may be set as per regular DC programs. *LCDriver2* will enforce these constraints when this mode is chosen. DC Line Trigger is discussed more in Sec. 7.e below.

b. File Menu


LC Driver 2
File
Upload/Download
Utility
About
Save All Programs To Files... Ctrl+S
Load All Programs From Files... Ctrl+O
Save Current Program To File...
Load Current Program From File...
Quit Ctrl+Q

LCDriver2 retains up to nine programs in its memory for the currently-active controller on USB. The options available from the File menu permit transferring programs to/from *LCDriver2*'s program memory and files stored on the host computer. The LC programs are stored in an ASCII text file format; there are pre-inserted comments in the file that explain the various sections and how to edit them in an ASCII text editor such as *Notepad*.

Save All Programs to Files… will save all programs currently retained in *LCDriver2*'s memory to ASCII text files. It will prompt for a file prefix (e.g. "SetA") for the program set, and then store ten programs to the designated location using the file prefix to compose names for all LC programs (e.g. "SetA_Prog_1", "SetA_Prog_2"). Program 0 will also be saved as a text file. Note that this menu option does not send any messages to the attached controller.

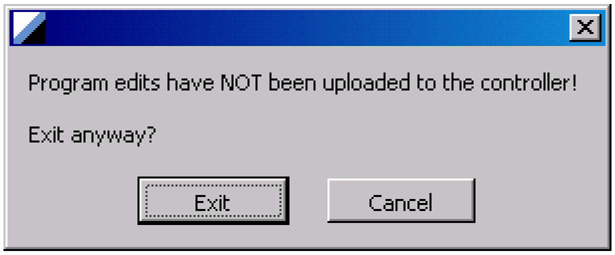
Load All Programs From Files… will load *LCDriver2*'s program memory with ten programs previously stored on the host with the Save All Program…option. It will prompt to choose one file from a sequence, and then will read all files into memory. Messages will be sent to the controller to adjust the currently-active program to agree with the file read in for this slot.

The Save/Load All Program… menu options can be used to replicate one controller's programs to other controllers, as explained in Sec. 7.e below.

Save Current Program To File… will save the currently-active program in the editor to a text file with an arbitrary name. This menu option does not send any messages to the attached controller.

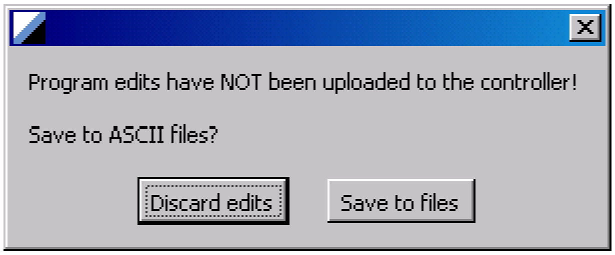
Load Current Program From File… will load a pre-stored LC program in a text file into the currently-active program. Messages will be sent to the controller to adjust the currently-active program to agree with the file read in.

Quit will exit *LCDriver2*, but leave the controller(s) on USB operating in the NEW_STANDALONE state, running the last-set currently active program. If there are program changes made in *LCDriver2* that have not been uploaded to the controller, warning prompts will appear as shown below.



Program edits have NOT been uploaded to the controller!

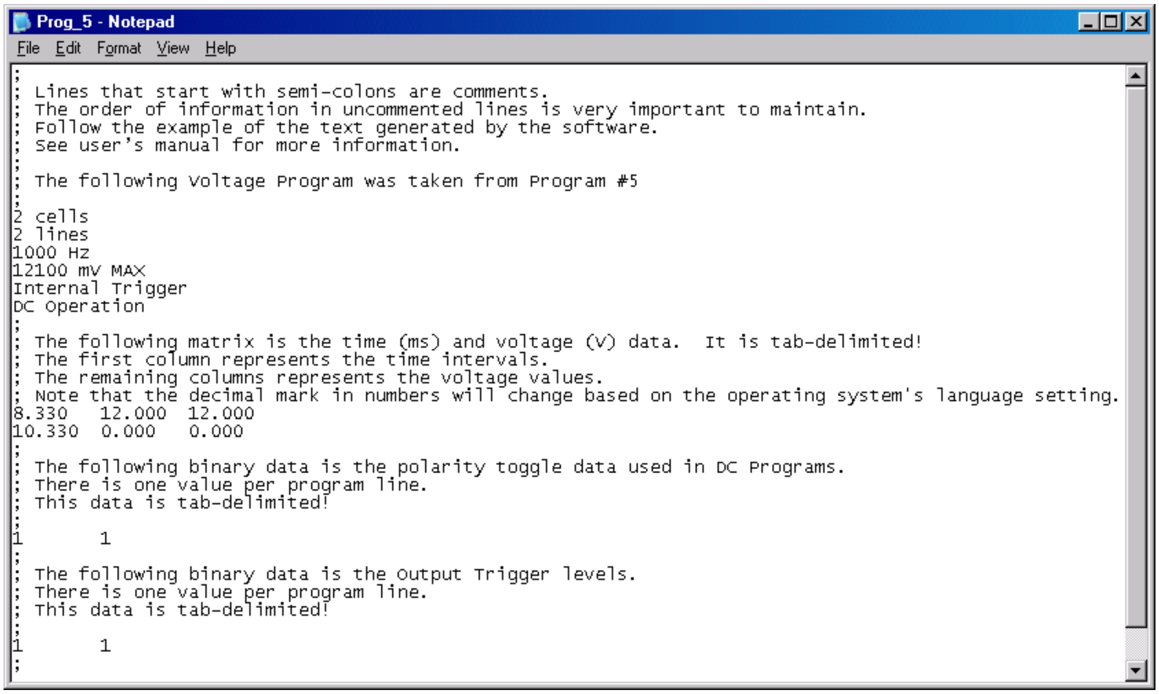
Exit anyway?



Program edits have NOT been uploaded to the controller!

Save to ASCII files?

The files saved in *LCDriver2* are ordinary ASCII text files that can be edited with *Notepad*. There are comments above each data section, as illustrated below.



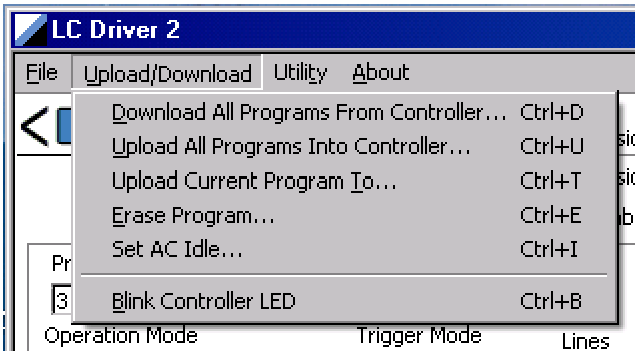
```
;
; Lines that start with semi-colons are comments.
; The order of information in uncommented lines is very important to maintain.
; Follow the example of the text generated by the software.
; See user’s manual for more information.
;
; The following Voltage Program was taken from Program #5
;
2 cells
2 lines
1000 Hz
12100 mV MAX
Internal Trigger
DC Operation
;
; The following matrix is the time (ms) and voltage (V) data.  It is tab-delimited!
; The first column represents the time intervals.
; The remaining columns represents the voltage values.
; Note that the decimal mark in numbers will change based on the operating system's language setting.
8.330	12.000	12.000
10.330	0.000	0.000
;
; The following binary data is the polarity toggle data used in DC Programs.
; There is one value per program line.
; This data is tab-delimited!
;
1	1
;
; The following binary data is the Output Trigger levels.
; There is one value per program line.
; This data is tab-delimited!
;
1	1
;
```

Note in particular that the decimal-point delimiter displayed in the *LCDriver2* onscreen program matrix and written to the program text files is determined by the host computer’s chosen language option. In Windows 7, this is accessed from Control Panel > Clock, Language, & Region > Region & Language; in Windows XP, this is accessed from Control Panel > Date, Time, Language & Regional Options > Regional & Language Options.

μLC200-12 users in mixed-language environments should be aware that program text files written in one language option and read in a different language option will not be converted correctly by *LCDriver2*. For mixed-language environments in which program text files will be exchanged, it is recommended that the text files be hand-edited to insert the proper decimal-point delimiter for the reading computer.

c. Upload/Download Menu

The options available from the Upload/Download menu permit transfer of LC programs to/from *LCDriver2*'s program memory and the µLC200-12's program memory. The controller has an onboard EEPROM that retains LC program contents for operation in controller 'standalone' mode.

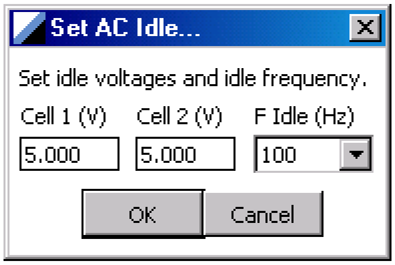

LC Driver 2
File
Upload/Download
Utility
About
Download All Programs From Controller... Ctrl+D
Upload All Programs Into Controller... Ctrl+U
Upload Current Program To... Ctrl+T
Erase Program... Ctrl+E
Set AC Idle... Ctrl+I
Blink Controller LED Ctrl+B
Operation Mode
Trigger Mode
Lines

Download All Programs From Controller… will refresh all slots of *LCDriver2*'s program memory with the contents of the currently-active controller's EEPROM. If program changes have been made in *LCDriver2* and not been saved to text files, a prompt warning will appear before the controller's EEPROM contents are read. This menu option can be used to "resync" *LCDriver2*'s program memory with the µLC200-12's EEPROM.

Upload All Program Into Controller… will transfer all programs slots in *LCDriver2* program memory into the corresponding slots of the µLC200-12's EEPROM. This menu option will make the controller's onboard EEPROM programs identical to what is currently being held in *LCDriver2*.

Upload Current Program To… will transfer the currently-active program in *LCDriver2* into a designated controller EEPROM slot. Most users will find this to be a frequently-used option of the Upload/Download menu. A prompt will appear allowing the user to select a µLC200-12 program slot; note this need not be the same slot as selected by the *LCDriver2* "Program" drop-down control. Once the update is complete, *LCDriver2* will adjust its program memory to agree with the contents of controller EEPROM.

Erase Program… will prompt for a controller EEPROM slot to make blank, and then blank out this program slot both on the controller and in *LCDriver2*. By definition, a blank program is a one-line, two-cell AC Internal Trigger program, with carrier frequency and cell voltages set to the currently-set AC idle frequency and amplitudes



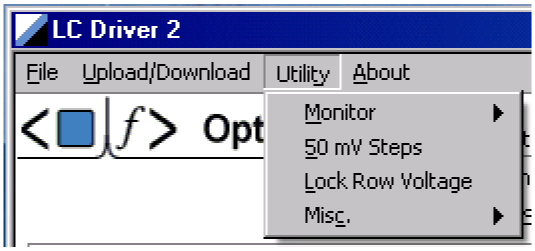
Set idle voltages and idle frequency.

Cell 1 (V)	Cell 2 (V)	F Idle (Hz)
5.000	5.000	100

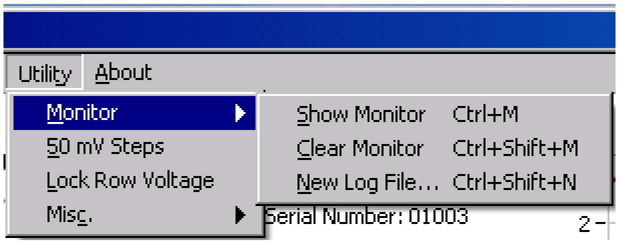
Set AC Idle… sets the AC carrier frequency and cell amplitudes of the AC "idle" condition. The controller will output this AC idle frequency and amplitudes when a blank program is created in *LCDriver2*, or when the controller is in OLD_HOST mode prior to *LCDriver2* being run.

Blink Controller LED will briefly strobe the interior Status LED of the currently-active controller. It is useful when there are multiple controllers on USB to help identify the currently-active controller selected by the “Device” control in *LCDriver2*. It can also be used as a confirmation of host computer/controller USB connectivity.

d. Utility Menu

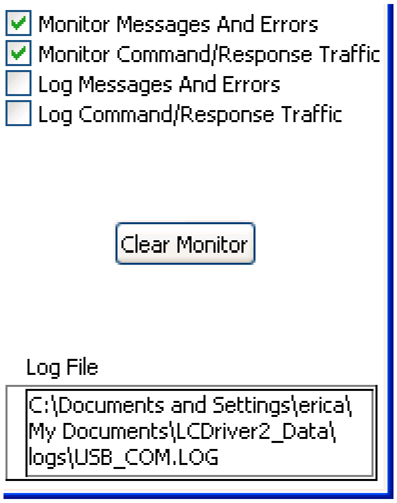


The options available from the Utility menu are not as frequently used. Some of this functionality is more advanced and is discussed in Sec. 7.e below.



Monitor is a submenu that controls the behavior of a sub-window that displays the actual *LCDriver2*/µLC200-12 controller protocol message stream. The Monitor is primarily used for diagnostics and troubleshooting. Selecting **Show Monitor** will open the Monitor sub-window just below the main application window. The *LCDriver2* commands and queries sent to the µLC200-12 are shown one per line, with the *LCDriver2* message on the left, and the corresponding response from the controller on the right. Re-selecting Show Monitor will close the Monitor window. Note that *LCDriver2* will always start up with the Show Monitor option turned off (unchecked).

Clear Monitor will clear the monitor display of all previous messages. This can be useful for capturing a stream of messages of particular interest. This menu items performs the same action as the “Clear Monitor” button in the Monitor sub-window.



☑ Monitor Messages And Errors
☑ Monitor Command/Response Traffic
☐ Log Messages And Errors
☐ Log Command/Response Traffic

Log File

C:\Documents and Settings\erica\My Documents\LCDriver2_Data\logs\USB_COM.LOG

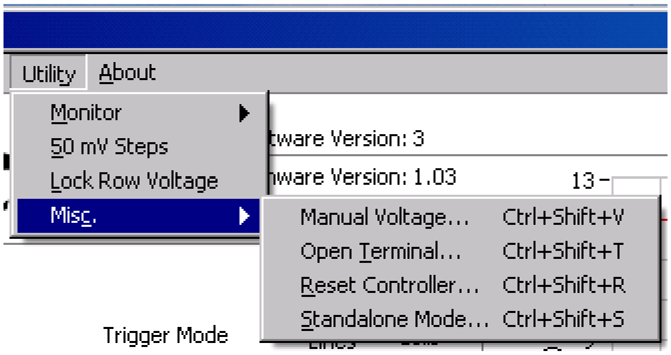
New Log File… permits an alternate log file to be started. The Monitor sub-window contains four checkboxes. The two “Monitor” checkboxes control what is displayed in the Monitor sub-window. The two “Log” checkboxes control what is written to the current log file. These checkbox settings are retained by *LCDriver2* when the application is closed.

Checking the “Log Command/Response Traffic” box permits the *LCDriver2*/µLC200-12 controller protocol message stream to be captured to a log file for later inspection. If either of these boxes are checked, *LCDriver2* will create a log file.
The default capture log file will be stored as

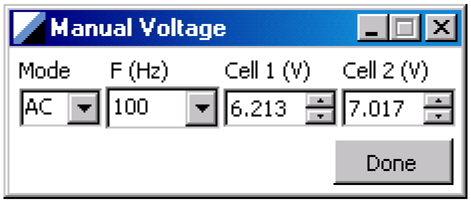
`C:\Documents and Settings\<username>\My Documents\LCDriver2_Data\logs\USB_COM.LOG`. The log file will be overwritten every time *LCDriver2* is run. An alternate log file may be selected by the New Log File… option. This option is intended for test setups where records of each run are desired to be kept in separate files.

The **50 mV Steps** and **Lock Row Voltage** options are discussed below in Sec. 7.e.

The **Misc** submenu allows access to special controller features. The only item in this menu normally accessed by most users is the Manual Voltage… option.


Utility
About
Monitor
50 mV Steps
Lock Row Voltage
Misc.
tware Version: 3
hware Version: 1.03
13 -
Manual Voltage… Ctrl+Shift+V
Open Terminal… Ctrl+Shift+T
Reset Controller… Ctrl+Shift+R
Standalone Mode… Ctrl+Shift+S
Trigger Mode

Manual Voltage… opens up a separate dialog box that permits simple static amplitude control of the LC channels in AC mode.



Mode	F (Hz)	Cell 1 (V)	Cell 2 (V)
AC	100	6.213	7.017

In this mode, there is no concept of dynamic host state or LC programs. The µLC200-12 is forced into AC operating mode in the OLD_HOST state. The edit boxes act as "knobs" that just set a fixed carrier amplitude for each cell.

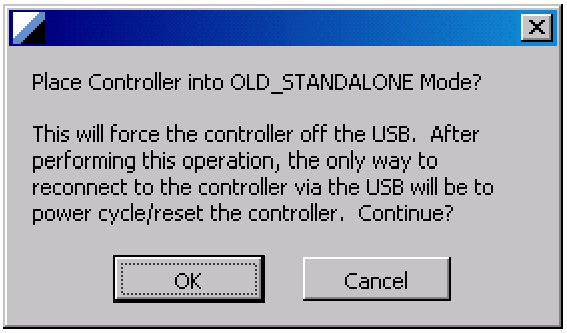
For nematic LC cell applications, AC mode should be used. However, DC mode can be set as well; when selected, warning message boxes will appear cautioning about possible device damage. If "Continue" is chosen and DC mode selected, the controller will output DC voltages of the selected polarity on each channel.

When the Manual Voltage dialog box is closed with the "Done" button, the controller is reset and placed back in dynamic mode (NEW_HOST), and will resume executing the currently-active program. The Manual Voltage settings do not affect the currently-active program in any way.

Open Terminal… is for field diagnostics and troubleshooting. This option allows *LCDriver2*/µLC200-12 controller protocol messages to be sent directly to the µLC200-12 and is intended only for field support in consultation with Optical Finesse.

Reset Controller… can be used to reset the µLC200-12's onboard microcontroller without a power off/on sequence. Selecting it will open an explanatory message box. If "Continue" is selected, a `*RST:` message will be sent to the µLC200-12, and then the controller placed back in dynamic mode (NEW_HOST).

Standalone Mode… is the only way to access the OLD_STANDALONE state discussed in Sec. 6.b above. Selecting it will display the explanatory message box shown below cautioning that once this state is entered, the only way to exit it is to disconnect the controller from USB power. If "Yes" is chosen, the µLC200-12 is placed in the OLD_STANDALONE state, and *LCDriver2*'s USB annunciator will change to read "No Connection". The controller will still be powered by USB but will not be able to be "seen" by *LCDriver2* until detached and then reattached to the bus.



Place Controller into OLD_STANDALONE Mode?

This will force the controller off the USB. After performing this operation, the only way to reconnect to the controller via the USB will be to power cycle/reset the controller. Continue?

e. Advanced topics

LCDriver2 handles USB attach and detach of controllers while the application is running. *LCDriver2* also handles multiple instances of µLC200-12 controllers on USB. Although the "Device" control at window bottom is continuously updated with the number of controllers currently on USB, to avoid confusion it is recommended that all controllers be attached to USB prior to *LCDriver2* being run. The "Device" control selector will determine which controller *LCDriver2* is currently addressing. The **Upload/Download: Blink Controller LED** option may be used to facilitate controller identification. When a controller is first selected on USB, *LCDriver2* will issue a `*RST:` and read all pre-loaded programs, just as if the controller was attached to USB while *LCDriver2* is running. When a controller is deselected, it will remain in the NEW_HOST state executing the last-set currently-active program.

In a multi-controller setup, it is frequently desirable to replicate all of the programs from one "master" controller onto other "copy" controllers. This may be accomplished in the following way:

1. Connect the "master" controller to *LCDriver2*, and save all programs read from its EEPROM with the File/Save All Programs To Files… option.
2. Disconnect the "master" controller from USB, leaving *LCDriver2* open.
3. Connect one of the "copy" controllers to USB. *LCDriver2* will then read all programs stored on this controller's EEPROM into its program memory.
4. Overwrite these *LCDriver2* programs with the "master" programs by selecting the File/Load All Programs From Files… option and selecting one file from the "master" sequence just saved.
5. Select Upload/Download: Upload All Programs Into Controller… to write the "master" LC programs now held by *LCDriver2* into the "copy" controller's onboard EEPROM.
6. Disconnect the "copy" controller with *LCDriver2* still open. Repeat steps 3 through 5 for any additional "copy" controllers in the setup.

The **Utility/Lock Row Voltage** option may be used to set all cell voltages in a row of a program at once by editing a single cell's edit box. The Lock Row Voltage is a toggle setting; when set, a checkmark appears by the option. Re-selecting the option will turn it off. When turned on, changes to any cell voltage value in a row will replicate the new value to all other cells in that row.

Normally, the host computer's up|down arrow edit keys will increment|decrement the digit immediately to the left of the cursor when the cursor is in a time or voltage edit box in the program matrix. The **Utility/50 mV Steps** option changes this default behavior of the up|down arrow edit keys when editing a cell voltage. This option is a toggle setting; when set, a checkmark appears by the option. Re-selecting the option will turn it off. When turned on and the cursor placed in a voltage edit box in the program matrix, the up|down arrow edit keys will increment|decrement the voltage by 0.050 V, regardless of where the cursor is positioned in the edit box. This option is intended to facilitate adjustments in applications where the least-discernable optical difference corresponds to about a 50 mV adjustment. With this option, the cursor arrow can be positioned in a cell and adjustments made by the up|down arrow keys while the user's eyes continuously monitor the optical setup

The Lock Row Voltage option may be used in conjunction with the 50 mV step option to change all voltages is a row in 50 mV increments by up|arrow key adjustments of one cell in the row. These options are commonly used together to optimize settings for LC-panel-based stereoscopic installations. Note that the Utility/Lock Row Voltage and Utility/50 mV step settings are *not* retained by *LCDriver2* if the application is closed.

LCDriver2 V. 12 and up support DC Line Trigger mode. In this mode, red annunciator text appears adjacent to the program matrix regarding the special program constraints. DC Line Trigger programs must always have two or more lines, and the line durations for all lines except line 1 are always set to 9999.990 ms. *LCDriver2* will validate and enforce these constraints. Durations for line 1 may be set from a minimum 0.10 ms value to a maximum 9999.990 ms value. Note in particular that the maximum settable DC line duration for DC Internal and Program Trigger modes is 2000.000 ms; the 9999.990 ms maximum is for DC Line Trigger mode only.

DC Line Trigger is useful for situations in which the optical state of a LC shutter device must be toggled by an external trigger. This mode allows 'syncing' to an arbitrary frequency to set a 50% open/closed shutter duty cycle. Each of the external triggers will advance the program line; the line durations are set long (maximum value) with the understanding that they will be terminated by a trigger edge. In DC Line Trigger, the trigger edges are effectively controlling line duration for lines 2 and up. In the absence of any triggers, the line durations will default to their onscreen values (line 1 as set by user; lines 2 and up 9999.990 ms) and the device drive will still remain DC balanced.