Optical Finesse LLC

μLC400-30 Four-Channel Liquid-Crystal Controller

User's Manual

Manual revision
October 2013

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 μLC400-30 quad liquid-crystal controller is suitable for driving continuously-variable large-area LC devices such as variable retarders, polarization rotators & switches, shutters, variable attenuators, tunable color filters, and polarization-state generators/analyzers.

The μLC400-30 incorporates four independent LC channels, each with $30V_{RMS}$ of range and fully short-circuit protected. The LC channels are capable of operating either AC (carrier) or DC (carrier-less), and can drive large-area cells up to 650 nF capacitance. The standard LC connectorization is a mini-DIN6 receptacle.

The controller is operated by our *LCDriver* application via a full-speed USB 2.0 compliant interface. *LCDriver* permits dynamic editing of AC or DC programs up to 32 lines in length. A separate dialog box of the *LCDriver* application permits simple amplitude control of the LC channels in AC mode. Three trigger modes (internal, line, program) determine how program lines are executed. Up to nine programs may be pre-stored on the μLC400-30 for standalone execution upon disconnect of the USB cable. A front-panel pushbutton switch selects the executing standalone program.

A device-driver DLL is provided to permit user programming of the μLC400-30. The controller is 100% RoHS-compliant, CE certified, and is powered by a separate 12V international power supply.

Features

- Four independent LC channels with V_{RMS} amplitudes up to 31.6V
- Each channel capable of driving cells up to 650 nF of capacitance
- Pre-store up to nine 32-line AC (carrier) or DC (carrier-less) LC programs for standalone operation
- LC program timebase resolution of 0.05 ms
- Variety of programmable trigger modes
- Full-speed USB 2.0 compliant host-computer interface
- Intuitive *LCDriver* host application for Windows 7, Vista, Windows XP & Windows 2000 permits dynamic LC program editing
- 12VDC international power supply included

Technical Specifications

LC CHANNELS	
Number of LC channels	Four, independently programmable
Maximum voltage stroke	31.60 V_{RMS}
LC voltage output	Either AC (carrier + envelope) or DC (carrier-suppressed) LC drive waveforms, software selectable
AC waveforms	Square-wave carrier, 50.00% duty cycle. Frequencies of 1.0, 2.0, 2.5, 4.0 or 5.0 kHz, software-selectable
DC waveforms	Bipolar, 0.20 ms minimum time interval, 0.05 ms interval resolution
Amplitude resolution	16-bit; 1 mV voltage resolution
Amplitude accuracy	Better than ±0.3% typical, ±0.5% maximum, measured at full range
Residual DC	Less than 2 mV at any output
Drive capability	Each channel can drive cells up to 650 nF capacitance
Short circuit	Full-recovery short-circuit protection on all LC channels. Front-panel “Fault” LED will illuminate when channel is shorted
LC programs	Up to nine pre-stored 32-line LC programs for standalone operation stored in onboard EEPROM (Electrically Erasable Programmable Read-Only Memory). Front-panel pushbutton switch selects executing standalone program

INTERFACE	
Host computer interfaces	Full-speed (12 Mb/s) USB 2.0 compliant. 9600 bps RS-232 upon special arrangement
Software	*LCDriver* application software with host USB drivers for Windows 7, Vista, Windows XP, and Windows 2000 (executable only, Internet download; 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	DLL toolkit for programming in *LabVIEW*, *Visual Basic* or *Visual C++*. Toolkit works with 32-bit user-supplied applications in 32- or 64-bit environments. Support for 64-bit applications upon special arrangement
External Trigger	Opto-isolated Trigger Input; TTL-compatible logic levels and fan-in. Controller triggers on rising logic input edge

PHYSICAL	
Dimensions	6.1 in W *x* 7.1 in L *x* 2.1 in H (15.5 cm W *x* 18.0 cm L *x* 5.3 cm H)
Weight	1.1 lbs (0.5 kg), controller alone; 1.9 lbs (0.9 kg), controller + power supply
Chassis material	Black ABS plastic; machined aluminum front & back panels, black-anodized and laser etched
Front panel	• Power, Fault and Status LED's • Ten-position pushbutton switch for program selection in standalone mode
Back panel	• 50-ohm BNC female bulkhead for Trigger Input • Series B USB receptacle for USB to host computer • LC channel output connectorization: mini-DIN6 is standard; BNC or SMA connectors available upon special arrangement • 2.1 mm power jack receptacle
RoHS compliance	100% compliant (no exemptions used)
CE compliance	CE certified to EN 55022, EN 55024, and EN 61010-1
Operating temperature	5 to 45 ^{o}C
Power supply	Separate, supplied with controller. International +12VDC, 18W, 100-240VAC 50-60Hz input, UL/CE listed. 2.1 *x* 5.5 mm center-positive plug

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.

- 12V⎓1.5A The electrical input rating of this device is 12VDC/1.5A. Use only with supplied AC mains adaptor (international power supply).

- The power disconnect means for the controller is the plug on the international power supply cord. The mains socket outlet for this supply should be installed near the controller and be readily accessible. For normal use, we recommend plugging the controller's AC power cord into a switched AC power strip whose switch is readily accessible by the operator.

- ⚠ The voltages at the exposed pins of the mini-DIN6 "LC Output" receptacle, or the output terminations of a cable plugged into this receptacle pose a potential shock hazard, and must be enclosed so as to prevent contact.

- 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.

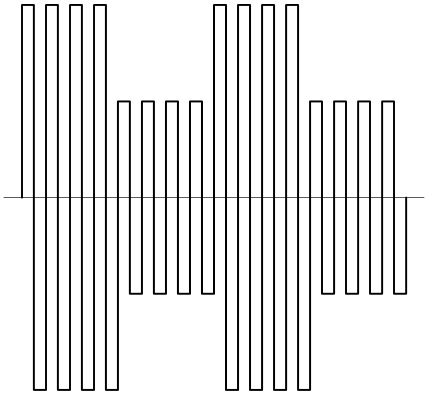
- The mini-DIN6 LC output cable length is recommended to be 5 ft (1.5 m). To avoid conducted RF immunity problems, under no circumstances should the LC cable harness be longer than 3 m.

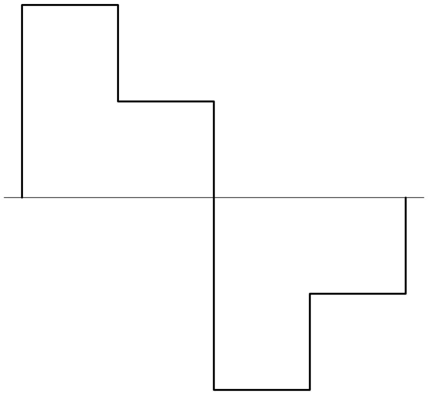
- The interior of the controller is not designed to be user accessible, and there are no user-serviceable parts inside. Contact Optical Finesse directly and return the controller if maintenance or calibration is required.

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

3. Overview

The μLC400-30 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 2 kHz. DC waveforms suppress the carrier and apply just the slowly modulating envelope to the cells. Because of hardware implementation, it is required that all 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.

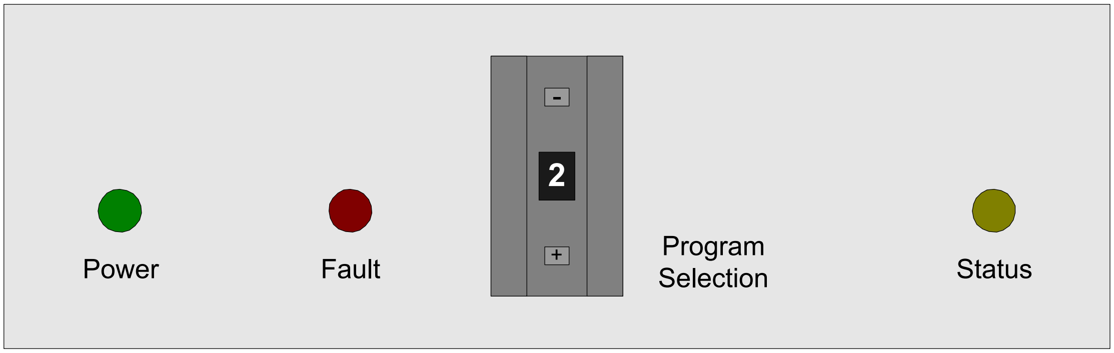
The μLC400-30 is designed to output voltages in the form of LC programs. Each line of a program consists of up to four voltages (one for each channel), and a time duration for which these voltages are held. These voltages specify either the AC-mode amplitude envelope, or the DC-mode bipolar voltage whose 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 a BNC receptacle located on the back panel. The external trigger source can be any TTL-compatible logic-level source. Triggers are generated by the rising edge of this source.

The μLC400-30 can run in either host or standalone mode. In host mode, the controller is connected by USB to a host computer, and the host can run or dynamically edit executing LC programs, either through the *LCDriver* application or an application that uses the device-driver toolkit to access the functionality of the controller. In standalone mode, no host computer and USB connection are present. The controller executes LC programs that have been previously stored in EEPROM onboard the controller; the front-panel ten-position pushbutton switch selects which program the controller is currently running.

4. Front & Back Panels

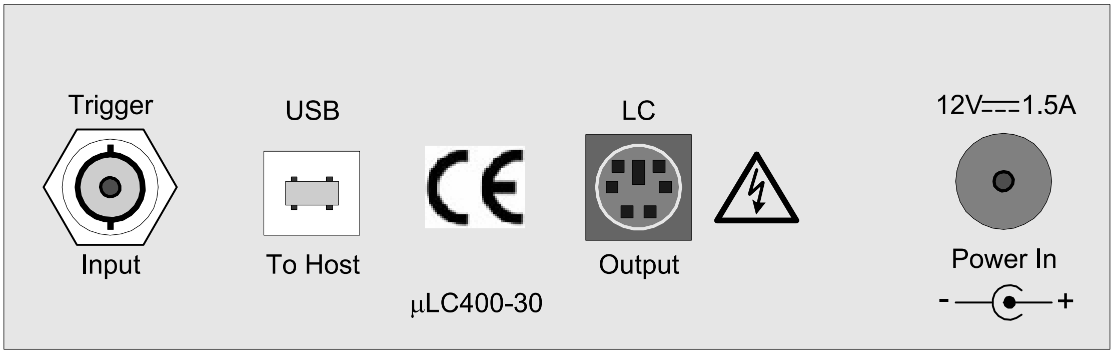
A brief description of the front & back panel indicators, controls & connections follows. Note that the trade-dress appearance of your controller's front & back panels may vary (e.g. different logos), but the functionality described below is the same.

a. Front Panel



- Power LED: lit green when controller is connected to power supply and properly powered (normal operation).
- Fault LED: In normal operation, this LED is off. It will be lit red when a short-circuit or other fault condition exists on an LC output. When the short-circuit or 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.
- Status LED: will be lit yellow and steady-ON when controller is running a pre-set program in standalone mode. If a blank or bad program is selected in standalone mode by the pushbutton switch, the Status LED will blink slowly. When the controller is operating in host mode (USB connection present), the Status LED will be steady-OFF.
- Ten-position pushbutton switch: selects currently-running LC program when the controller is operating in standalone mode (no USB connection). The program number is incremented/decremented by the "+" / "-" buttons. The pushbutton switch setting is ignored when the controller is running in host mode (host USB connection present).

b. Back Panel

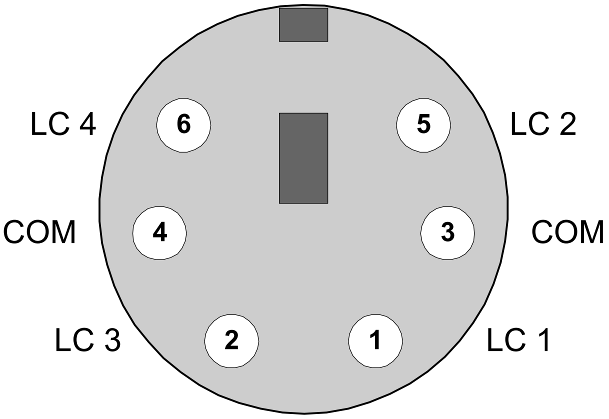


- BNC Trigger Input connector: accepts triggers from external TTL-compatible trigger source. The controller is designed to be rising-edge triggered (duty cycle does not matter). If a function generator is used as a trigger source, do *not* apply its analog output to this connector—use the sync or logic output instead, or set up the output for TTL.

- USB Series B receptacle: for full-speed USB 2.0 compatible connection to host computer. Note that the controller's behavior depends on whether a host USB connection is present or absent.

- Mini-DIN6 LC Output receptacle: Supplies four LC voltage output connections, along with two common (COM) connections. The COM connections are tied to the ground plane of the controller's PCB. The pin-out for this connector is shown below.

- 2.1mm 12VDC power jack: accepts 2.1mm output power connector of supplied external 12VDC/1.5A international power supply. Use <u>only</u> the power supply shipped with the controller.



Pin-out of female LC Output mini-DIN-6 receptacle (looking at face of connector)

5. Getting Started

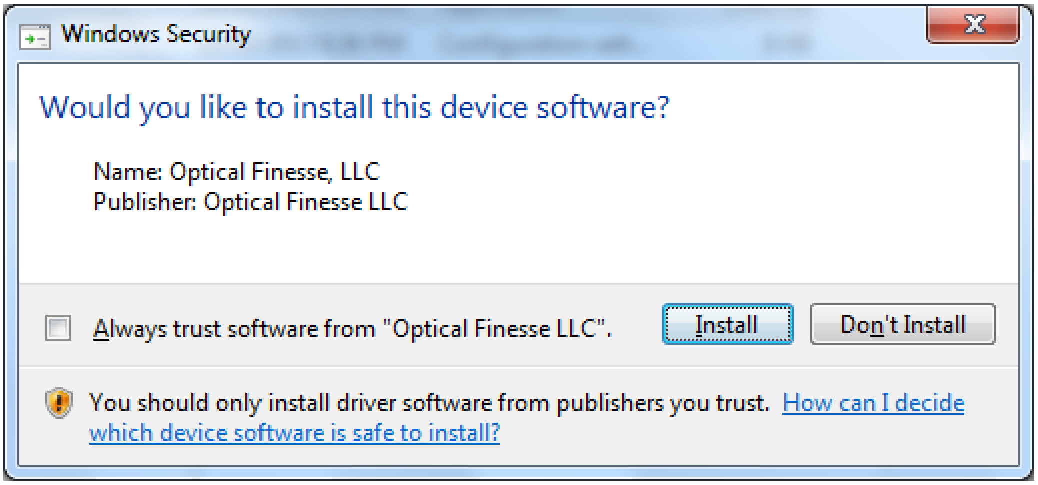
Prior to using the μLC400-30, the *LCDriver* application must be pre-installed on the host computer to be used with the controller. This application, the USB drivers, and the device-driver toolkit are available as a monolithic .ZIP file downloadable from the Internet. There are multiple builds of this file; please consult 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 *LCDriver* version with the most recent version.

Prior to software installation, unpack the μLC400-30, and plug the supplied AC power cord into the 12VDC international power supply. Connect the power supply to AC mains with the AC power cord. For normal use, it is recommended that the controller's AC cord be plugged into a switched AC power strip, and this switch be used to turn the controller on and off. If a power strip is used, the 2.1mm plug of the 12V supply can be plugged into the jack on the controller's back panel. For either method, leave the controller initially <u>off</u> until directed to turn it on in the instructions below. Plug the supplied USB cable into the controller's back panel USB receptacle.

a. New installs on Windows 7 & Vista systems

1. Download and save the appropriate .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)\LCDriver\` for the *LCDriver* program (for 32-bit Windows 7 & Vista, this location will be `C:\Program Files\LCDriver\`).
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

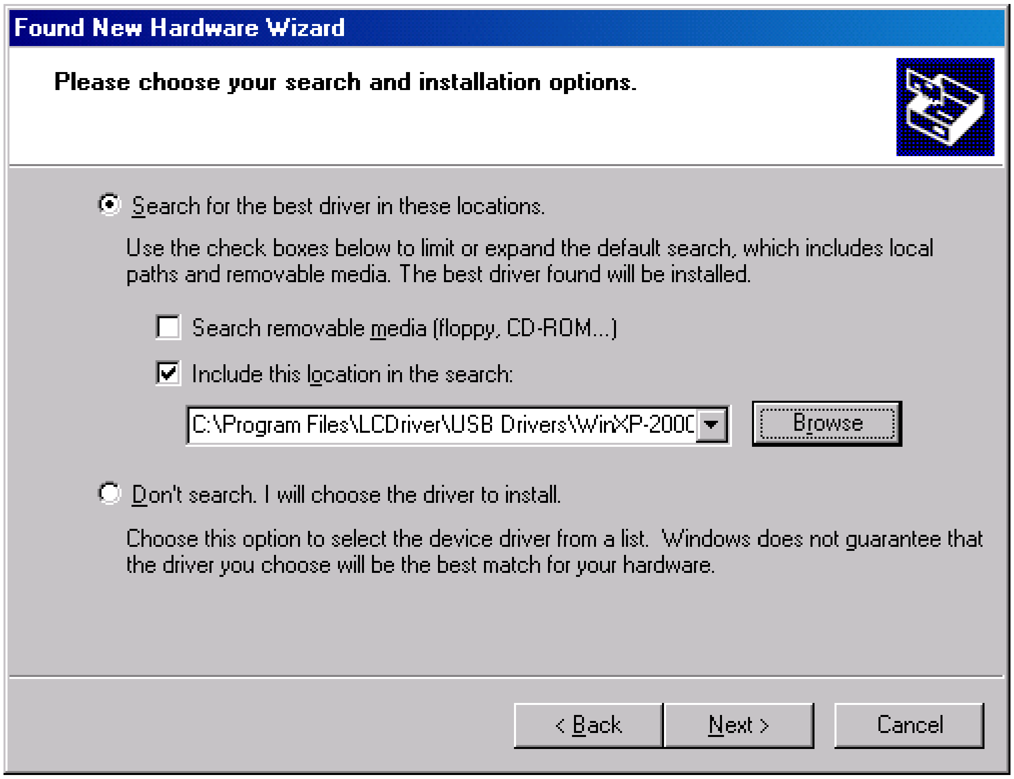
□ Always trust software from "Optical Finesse LLC".

You should only install driver software from publishers you trust.

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. After computer restart, plug the µLC400-30's USB cable into one of the host computer's USB ports. Power up the controller by turning on the AC power strip switch (if used), or plugging in the 2.1 mm plug of the 12V supply into the jack on the controller's back panel. The green Power LED should be ON.
9. The device driver software install will be completed, and a balloon will appear onscreen informing you that the install was successful. Start *LCDriver* from Window's Start menu: Start > All Programs > LCDriver > LCDriver. 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 appropriate .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\LCDriver\` for the *LCDriver* 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. After computer restart, plug the µLC400-30’s USB cable into one of the host computer’s USB ports. Power up the controller by turning on the AC power strip switch (if used), or plugging in the 2.1 mm plug of the 12V supply into the jack on the controller’s back panel. The green Power LED should be ON.
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\LCDriver\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\LCDriver\USB Drivers\WinXP-200C

○ 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.

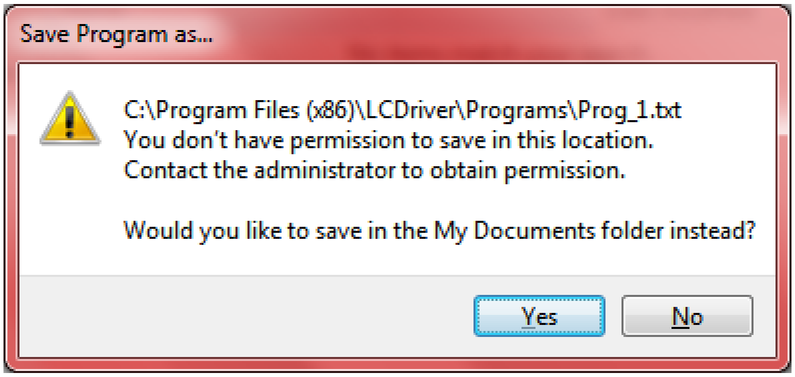
12. The USB device driver software install will be completed, and a balloon will appear onscreen informing you the install was successful.
13. Start *LCDriver* from Window's Start menu: Start > All Programs > LCDriver > LCDriver. The "Dynamic Connection" annunciator should appear in the application window's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

c. Updating *LCDriver* on an existing install

1. If the previous version of *LCDriver* is running, exit it.
2. Go to Windows Control Panel > Add or Remove Programs. Locate *LCDriver* in the list, and click on the Remove button to uninstall. Close Control Panel.
3. Download and save the appropriate .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 *LCDriver* 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.
9. The USB device drivers do *not* need to be reinstalled for an update of *LCDriver*.

d. Errata

As discussed in Sec. 7 below, *LCDriver* permits saving LC programs to the host computer's hard drive. Under 32-bit operating systems (Windows XP & 2000), the default location *LCDriver* will use for these programs is `C:\Program Files\LCDriver\programs`. However, the 64-bit operating systems (Windows 7 & Vista) will not permit *LCDriver* to save files to the corresponding default location of `C:\Program Files (x86)\LCDriver\programs`, even if administrator privileges are enabled for the user's log-in (see example warning prompt below). For Windows 7 & Vista, these files can be saved in `\My Documents` or another subdirectory under the user's log-in.



C:\Program Files (x86)\LCDriver\Programs\Prog_1.txt
You don't have permission to save in this location.
Contact the administrator to obtain permission.

Would you like to save in the My Documents folder instead?

Likewise, under Windows 7 & Vista, the log files discussed in Sec. 7.d below cannot be created or saved to their default location of `C:\Program Files (x86)\LCDriver\logs.` If log files are used, these files can be saved in `\My Documents` or another subdirectory under the user's log-in.

The µLC device-driver toolkit is also installed during the *LCDriver* installation process described above. Programmers who plan to use the toolkit to develop their own applications that incorporate the µLC400-30 should first install the *LCDriver* package, as the toolkit's .DLL requires the *LabVIEW* runtime engine, which is installed with *LCDriver*. After successful installation, programmers should copy `LCDriverUsb.dll` from `\LCDriver` and `ulc_dd.dll` from `\LCDriver\Toolkit` to the local subdirectory of their application.

Do <u>NOT</u> use the version of `LCDriverUsb.dll` found in `\LCDriver\Toolkit`! The version of `LCDriverUsb.dll` installed in `\LCDriver` will be the correct one for the host operating system; the `.dll` in `\LCDriver\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 to support programs using the µLC toolkit 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 all LC channels in the program. LC programs can be written for one to four active channels, but these channels must be sequential and start at LC_1. A one-channel program must use LC_1, a two-channel program must use LC_1 and LC_2, and so forth. Non-active LC channels for a program are 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 32 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. 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 0.20 ms, and is settable in 0.05 ms increments from this minimum value (0.20, 0.25, 0.30, etc.). *LCDriver* 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. Only AC mode programs can be Line-triggered.

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 External trigger BNC 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 external trigger edge restarts the program at the first line. AC programs can be re-triggered during any program line. 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. Program trigger is typically used to sync the μLC400-30 to an external periodic source (e.g. projector). 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 line.

Line Trigger mode is only meaningful for AC programs. In 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 BNC 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.

b. Host & Standalone

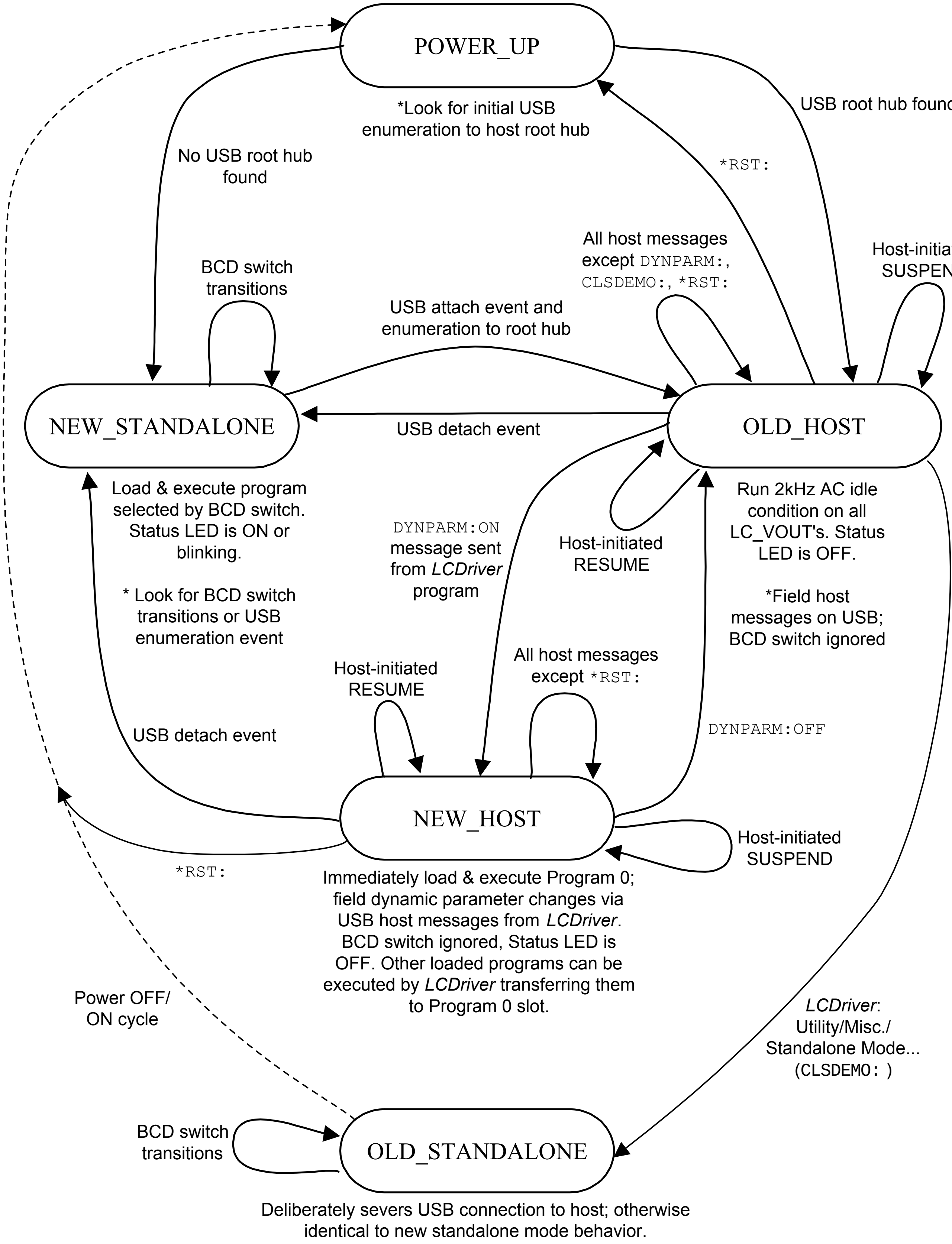
If, upon controller power-up, the μLC400-30 is recognized and enumerated by a USB root hub, the controller will go into host mode and accept messages from the host computer. In host mode, the front-panel pushbutton switch setting is ignored, and the front-panel Status LED is OFF. The host mode application may be either *LCDriver* or a user-supplied application that accesses the controller functionality through the .DLL device-driver toolkit.

If there is no USB root hub present upon power-up, or if the controller is dynamically detached from USB, the controller will go into standalone mode. In standalone mode, the controller executes whichever pre-loaded LC program is selected by the ten-position pushbutton switch. The front-panel Status LED is steady-ON in standalone mode. If a blank (un-programmed) LC program is selected by the pushbutton switch, the Status LED will blink slowly, and the LC outputs will run an "idle" condition of 2 kHz AC carrier on all channels. The AC idle voltage amplitudes can be set by host software.

When a controller in standalone mode is reattached to USB, the controller will transition back to host mode. Prior to *LCDriver* being run, the LC channels will run AC idle condition at their set amplitudes.

A more detailed state diagram is shown on the following page. Normally, the controller makes transitions between the POWER_UP, OLD_HOST, NEW_HOST, and NEW_STANDALONE states. The controller is in the NEW_HOST state (dynamic host mode) while *LCDriver* is being run; the controller is normally in OLD_HOST when the .DLL toolkit is being used with a user-supplied application. Note that the controller will normally transition to the NEW_STANDALONE state upon a USB detach.

The OLD_STANDALONE state is a special state that is only accessible by a command issued by *LCDriver*. In OLD_STANDALONE, the controller deliberately severs its USB connection to the host. This state is useful for setups in which the controller must always execute a pre-set stored program, even if the controller is re-attached to USB. In the normal NEW_STANDALONE standalone state, a USB attach event will cause the controller to stop executing its pre-loaded program, run AC idle condition on its channels, and await instructions from the host computer. This behavior differs from OLD_STANDALONE, as USB attach and detach events do not transition the controller out of this state.

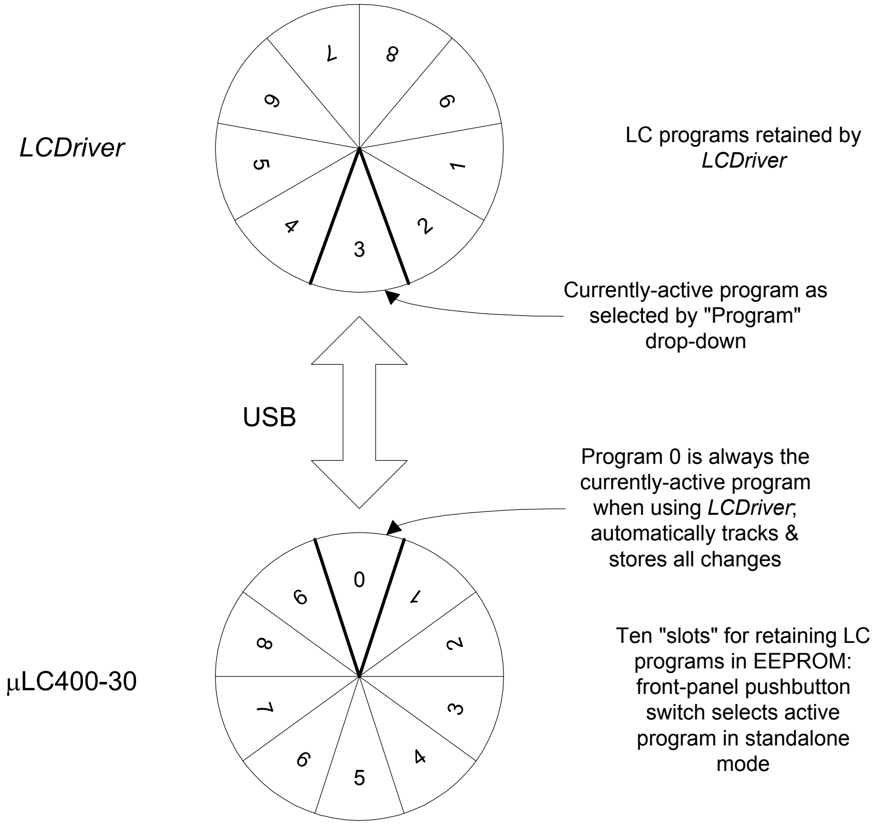

POWER_UP
*Look for initial USB enumeration to host root hub
No USB root hub found
USB root hub found
*RST:
All host messages except DYNPARM:, CLSDEMO:, *RST:
Host-initiated SUSPEND
BCD switch transitions
USB attach event and enumeration to root hub
NEW_STANDALONE
USB detach event
OLD_HOST
Load & execute program selected by BCD switch. Status LED is ON or blinking.
* Look for BCD switch transitions or USB enumeration event
DYNPARM:ON message sent from LCDriver program
Host-initiated RESUME
Run 2kHz AC idle condition on all LC_VOUT's. Status LED is OFF.
*Field host messages on USB; BCD switch ignored
Host-initiated RESUME
All host messages except *RST:
USB detach event
DYNPARM:OFF
NEW_HOST
Host-initiated SUSPEND
*RST:
Immediately load & execute Program 0; field dynamic parameter changes via USB host messages from LCDriver. BCD switch ignored, Status LED is OFF. Other loaded programs can be executed by LCDriver transferring them to Program 0 slot.
Power OFF/ ON cycle
LCDriver: Utility/Misc./ Standalone Mode... (CLSDEMO:)
BCD switch transitions
OLD_STANDALONE
Deliberately severs USB connection to host; otherwise identical to new standalone mode behavior.

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 *LCDriver* running, *LCDriver* will automatically read all pre-stored programs on the controller and retains a copy of each program in its own memory for later editing. When using *LCDriver*, it is important to understand and maintain the distinction between the LC programs retained by *LCDriver* (stored in the host computer's memory) and the LC programs retained by the μLC400-30 (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 *LCDriver*, 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 (*LCDriver* 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 *LCDriver*, 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 *LCDriver*. Likewise, to edit a previously-saved program in another slot, it must first be transferred to the Program 0 slot by *LCDriver* and then dynamically edited.

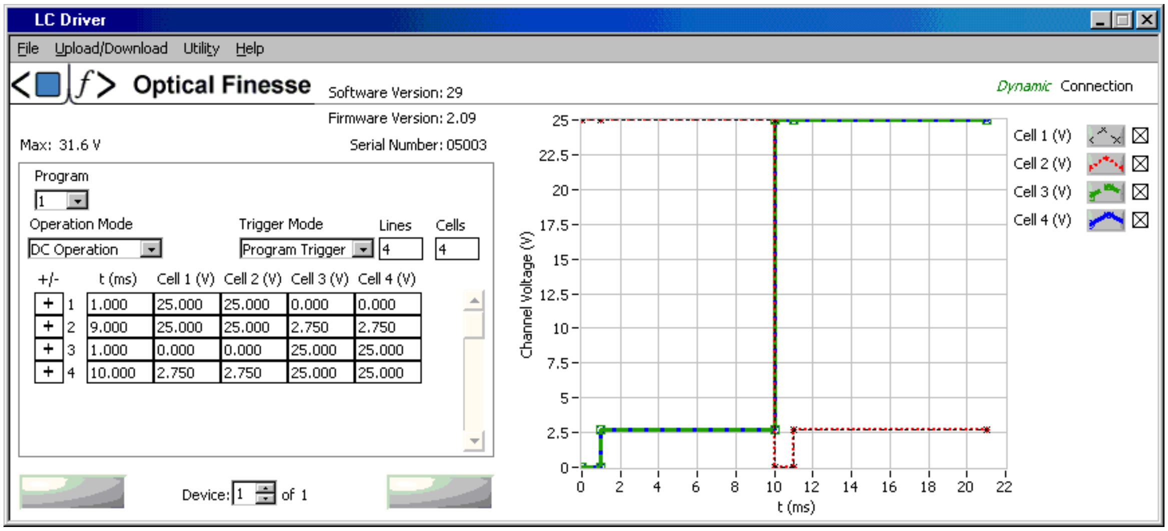

LCDriver
7
8
6
9
5
1
4
3
2
LC programs retained by *LCDriver*
Currently-active program as selected by "Program" drop-down
USB
Program 0 is always the currently-active program when using *LCDriver*; automatically tracks & stores all changes
0
9
1
8
2
μLC400-30
7
3
6
5
4
Ten "slots" for retaining LC programs in EEPROM: front-panel pushbutton switch selects active program in standalone mode

7. LCDriver Application Details

This section explains the functionality of the *LCDriver* application supplied with the μLC400-30. *LCDriver* can be used to read, edit, and store LC programs on controllers.

a. Overview

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



Software Version: 29

Firmware Version: 2.09

Serial Number: 05003

Max: 31.6 V

Dynamic Connection

Program: 1

Operation Mode: DC Operation

Trigger Mode: Program Trigger

Lines: 4

Cells: 4

+/-		t (ms)	Cell 1 (V)	Cell 2 (V)	Cell 3 (V)	Cell 4 (V)
+	1	1.000	25.000	25.000	0.000	0.000
+	2	9.000	25.000	25.000	2.750	2.750
+	3	1.000	0.000	0.000	25.000	25.000
+	4	10.000	2.750	2.750	25.000	25.000

Device: 1 of 1

Cell 1 (V) ☒

Cell 2 (V) ☒

Cell 3 (V) ☒

Cell 4 (V) ☒

The software version of *LCDriver* is displayed at the application window center; below is the embedded software (firmware) version of the active μLC400-30 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 *LCDriver* 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". *LCDriver* supports multiple instances of μLC400-30's on USB. The currently-active controller is selected by the "Device" control at window bottom. This control also shows how many μLC400-30's are presently on USB.

When *LCDriver* 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 *LCDriver* 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, *LCDriver* will display “Blank Program” adjacent to the “Program” drop-down. Blank programs are always one-line, four-cell 2 kHz AC internal trigger programs, with the cell voltages set to the currently-set AC idle amplitudes.

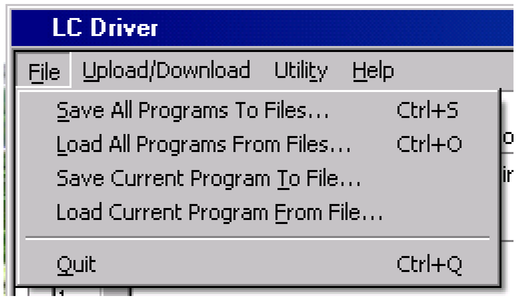
The edit controls will change depending on the selection of “Operation Mode”. For AC operation, a “F (Hz)” drop-down control will be displayed, with a corresponding “Period (ms)” above it. For DC operation, no frequency or period controls will appear, but an additional “+/-” column will be displayed.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

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 32, and the number of cells from 1 to 4. 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 additional cells are 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 µLC400-30’s the maximum settable voltage is 31.600V. 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 0.20 ms, and is settable in 0.05 ms increments from this minimum value (0.20, 0.25, 0.30…). *LCDriver* will validate and appropriately round user entries to enforce these constraints.

b. File Menu

LCDriver 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 *LCDriver*’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*.


LC Driver
File
Upload/Download
Utility
Help
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

Save All Programs to Files… will save all programs currently retained in *LCDriver*'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.

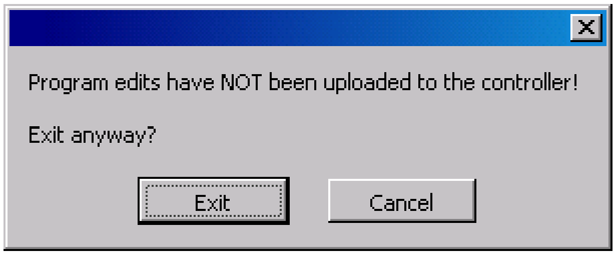
Load All Programs From Files… will load *LCDriver*'s program memory with ten programs previously stored on the host with the Save All Program…option. It will prompt to choose one file from sequence, and then will read all files into memory. Messages will be sent to the controller to adjust the currently-active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

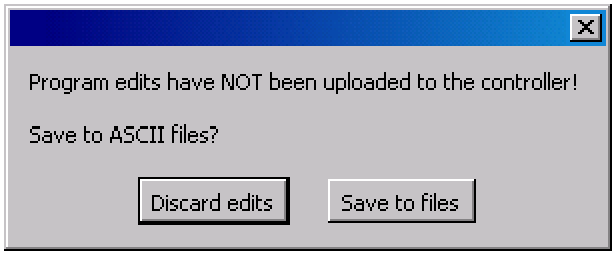
Save Current Program To File… will save the currently-active program in the editor to a text file with an arbitrary name. This menu option does not send any messages to the attached controller.

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


Program edits have NOT been uploaded to the controller!

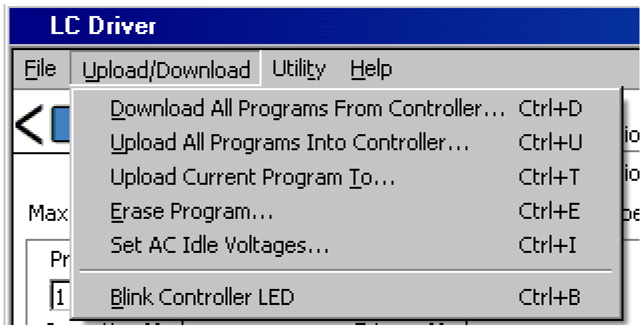
Exit anyway?


Program edits have NOT been uploaded to the controller!

Save to ASCII files?

c. Upload/Download Menu

The options available from the Upload/Download menu permit transfer of LC programs to/from *LCDriver*'s program memory and the µLC400-30's program memory. The µLC400-30 controller has an onboard EEPROM that retains LC program contents for operation in controller standalone mode.


LC Driver
File
Upload/Download
Utility
Help
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 Voltages... Ctrl+I
Blink Controller LED Ctrl+B
Max
Pr
1

Download All Programs From Controller… will refresh all slots of *LCDriver*'s program memory with the contents of the currently-active controller's EEPROM. If program changes have been made in *LCDriver* 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" *LCDriver*'s program memory with the µLC400-30's EEPROM.

Upload All Program Into Controller… will transfer all programs slots in *LCDriver* program memory into the corresponding slots of the µLC400-30's EEPROM. This menu option will make the controller's onboard EEPROM programs identical to what is currently being held in *LCDriver*.

Upload Current Program To… will transfer the currently-active program in *LCDriver* into a designated controller EEPROM slot. A prompt will appear allowing the user to select a µLC400-30 program slot; note this need not be the same slot as selected by the *LCDriver* "Program" drop-down control. Once the update is complete, *LCDriver* will adjust its program memory to agree with the contents of controller EEPROM.

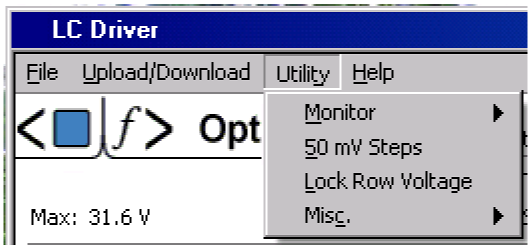
Erase Program… will prompt for a controller EEPROM slot to make blank, and then blank out this program slot both on the controller and in *LCDriver*. By definition, a blank program is a one-line, four-cell 2 kHz AC internal trigger program, with cell voltages set to the currently-set AC idle amplitudes

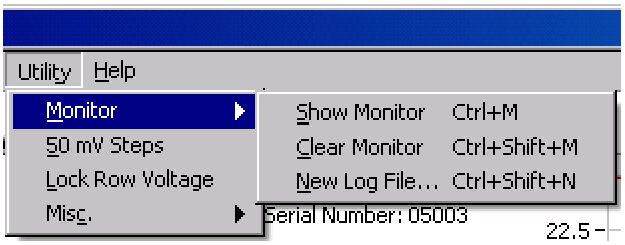
Set AC Idle Voltages… lets the amplitudes of the AC "idle" carrier to be set for each LC channel. The controller will output AC idle amplitudes when a blank program is selected in standalone mode, or when the controller is in OLD_HOST mode.

Blink Controller LED will briefly strobe the Status LED of the currently-active controller. It is useful when there are multiple controller on USB to help identify the currently-active controller selected by the "Device" control. 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.

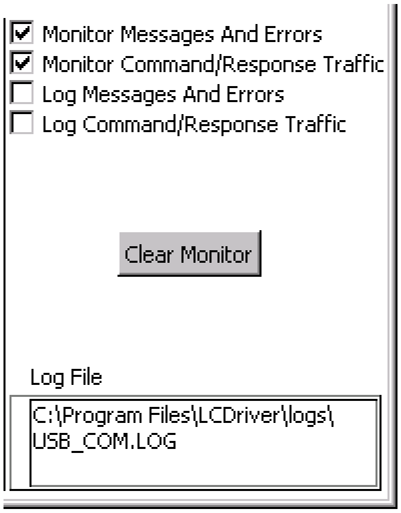




Monitor is a submenu that controls the behavior of a sub-window that displays the actual *LCDriver*/controller protocol message stream. The Monitor is primarily used for diagnostics and troubleshooting. Selecting **Show Monitor** will open the Monitor sub-window just below the main application window. The *LCDriver* commands and queries sent to the µLC400-30 are shown one per line, with the *LCDriver* 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 *LCDriver* 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.

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 *LCDriver* when the application is closed.



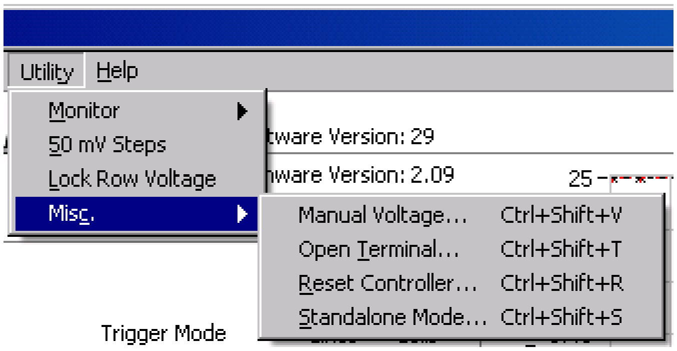
☑ Monitor Messages And Errors
☑ Monitor Command/Response Traffic
☐ Log Messages And Errors
☐ Log Command/Response Traffic

Log File
C:\Program Files\LCDriver\logs\USB_COM.LOG

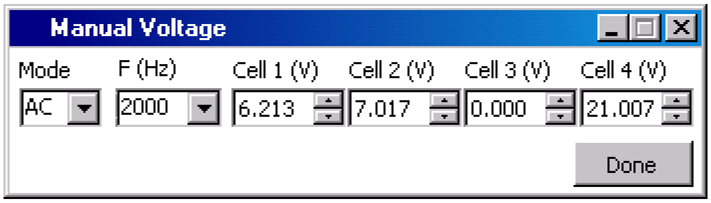
Checking the "Log Command/Response Traffic" box permits the *LCDriver*/controller protocol message stream to be captured to a log file for later inspection. If either of these boxes are checked, *LCDriver* will create a log file. The default capture log file will be `C:\Program Files\LCDriver\logs\USB_COM.LOG`. The log file will be overwritten every time *LCDriver* 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
Help
Monitor
50 mV Steps
Lock Row Voltage
Misc.
Manual Voltage… Ctrl+Shift+V
Open Terminal… Ctrl+Shift+T
Reset Controller… Ctrl+Shift+R
Standalone Mode… Ctrl+Shift+S
tware Version: 29
nware Version: 2.09
25
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)	Cell 3 (V)	Cell 4 (V)
AC	2000	6.213	7.017	0.000	21.007

In this mode, there is no concept of dynamic host state or LC programs. The µLC400-30 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.

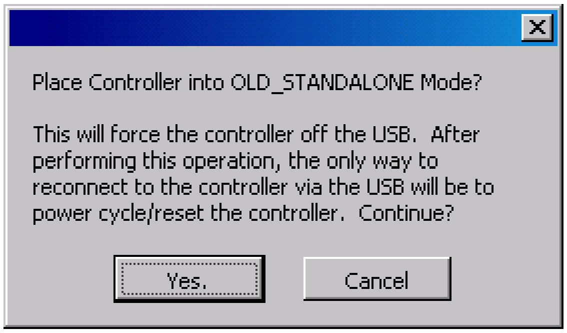
For nematic LC cell applications, AC mode should be used. However, DC mode can be set as well; when selected, warning message boxes will appear cautioning about possible device damage. If “Continue” is chosen and DC mode selected, the controller will output DC voltages of the selected polarity on each 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 *LCDriver*/controller protocol messages to be sent directly to the µLC400-30 and is intended only for field support in consultation with Optical Finesse.

Reset Controller… can be used to reset the µLC400-30'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 µLC400-30, and then the controller placed back in dynamic mode (NEW_HOST).

Standalone Mode… is the only way to access the special 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 a power off/on sequence, by deliberate design. If "Yes" is chosen, the µLC400-30 is placed in the OLD_STANDALONE state, and *LCDriver*'s USB annunciator will change to read "No Connection".



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?

This menu item is intended to provide support for "legacy" µLC400-30's. As explained in Sec. 6.b, OLD_STANDALONE should only be entered in circumstances where the controller must continue to operate a standalone program, even if inadvertently reattached back to USB.

e. Advanced topics

If only one standalone LC program is of interest or needed in a situation, a short-cut can be taken by leaving the controller's front-panel pushbutton switch set to "0". As explained above, all *LCDriver* program edits are implicitly performed on Program 0. If the front-panel switch is left in the "0" position, the controller will automatically execute the currently-active program that has been automatically stored in the Program 0 slot when placed in standalone mode.

LCDriver handles USB attach and detach of controllers while the application is running. *LCDriver* also handles multiple instances of μLC400-30's on USB. Although the "Device" control at window bottom is continuously updated with the number of controllers currently on USB, it is recommended that all controllers be attached to USB prior to *LCDriver* being run to avoid confusion. The "Device" control selector will determine which controller *LCDriver* 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, *LCDriver* will issue a `*RST:` and read all pre-loaded programs, just as if the controller was attached to USB while *LCDriver* is running. When a controller is deselected, it will remain in the NEW_HOST state executing the 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 *LCDriver*, 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 *LCDriver* open.
3. Power up and connect one of the "copy" controllers to USB. *LCDriver* will then read all programs stored on this controller's EEPROM into its program memory.
4. Overwrite these *LCDriver* 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 *LCDriver* into the "copy" controller's onboard EEPROM.
6. Disconnect the "copy" controller with *LCDriver* 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 *LCDriver* if the application is closed.