Laboratory Workbench version 4.1.1
Release Notes for RedHawk Linux

October 14, 2004

DAC002691-4.1.1

READ ME BEFORE
INSTALLING
THIS PRODUCT

concurrent

1. Introduction:

This document assists the user in installing the Concurrent Computer Corporation's Laboratory Work Bench™ (LWB) on a RedHawk™ Linux® OS for use with the LCAIO (PCI-16AIO) board.

LWB Version 4.1.1 is an updated version of Concurrent Computer Corporation Laboratory Workbench. It is a graphical user interface integrated with data acquisition, graphics, and data processing modules. LWB supports the Concurrent family of data acquisition hardware interfaces, graphics display systems, and high-performance Xeon processors in an environment that requires no programming for a wide variety of applications.

The LWB V4.1.1 distribution includes: LWB V4.1.1 release notes (this document) and a system-dependent software kit consisting of LWB V4.1.1 rpm for RedHawk Linux.
The Laboratory Workbench User's Guide man pages are included as html files and are included with the LWB rpm. In order for LWB V4.1.1 to be installed correctly, the hardware and software requirements outlined in the following sections must be observed.

2. Requirements:

The following hardware is required for LWB V4.1.1: Any iHawk Series Concurrent system:

- IHawk Series Concurrent System
- Minimum of 512MB of memory. (1Gb recommended)
- Hard disk capacity commensurate to expected data recording levels.

The following hardware may be used with LWB V4.1.1:

  - LCAI - 16bit 16ch(diff.) / 32ch (single) 300KHz Multiplexer type Analog to Digital Converter (PCI-16AIO)
  - LCAO - 16bits 4ch 300KHz Digital to Analog Converter (PCI-16AIO)
  - /dev/audio – Generic audio device (used for demo purposes only)

The following software is required for LWB V4.1.1:

- RedHawk Revision 2.1 or 2.2
- PCI board & Driver installed as required. (Optional)

3. Installation:

This section provides detailed, step-by-step instructions for the installation of the LWB software.

The installation is performed by downloading a Linux rpm file from the Concurrent Computer Corporation ftp site, running the rpm utility and configuration your LWB user environment.

Following are the step-by-step instructions that should be read before installation begins.

3.1 Installation

If you have installed the previous version of a LWB internal release, it is safer to remove it before installing the new version. Uninstalling it will not delete your instruments if you had created any.

Installing LWB:

```
# cd /mnt/cdrom  (or location where rpm has been placed)
```

This is the directory under which your CD-ROM is mounted. If you have a different setup, just change directory "cd" to the directory that corresponds to your mounted cd or location you have downloaded the rpm and perform the following:

```
# ./ccur_install
```

If you have a previous version of LWB installed or if you have lwb_ivp, the script will prompt you to remove it before proceeding with the installation. At this point you can either continue with the installation or abort the installation.

For more information, see the online help documentation while running the LWB product or you may go to the link: file:///usr/lib/LWB/help/C/Index.htm once the product has been installed.

Remove Previous Versions of LWB: *(If required)*

Open an xterm or shell on your Linux workstation and sign on as root user

```
# su
# Password:   <enter root password>
# rpm -e lwb
```

3.2 Running LWB

You may now start LWB from an xterm window. Ensure that your $DISPLAY variable is set properly before running from a remote X-server terminal.

open a new Xterm window and type :

```
# lwb
```

This should start the Laboratory WorkBench application.

If you are running the KDE or Gnome window manager, the install procedure will create two shortcut icons on your desktop:

LWB
LWB release notes.

The two Desktop icons will only be created for the user root and for all subsequent login ids created after the installation of LWB.

If Gnome is your window manager you can manually add those shortcuts to your desktop by copying the *LWB*.desktop* files from:

/root/.gnome-desktop to /your_home_directory/.gnome-desktop

Similarly, if KDE is your window manager, copy the files:

/root/Desktop/LWB to /your_home_directory/Desktop*

3.3 Example Instruments

Some examples of LWB instruments are provided as part of the *lwb* package.
You should have a private copy of those examples in:

/your_home_directory/LWB/instruments

This is the default location for LWB instruments.

For any examples that require actual data acquisition hardware, the appropriate hardware needs to be installed. The driver must also be installed and loaded prior to loading these examples. Failure to do so will result in the example being displayed, however all links to the components will be broken and the example will fail to run.

3.4 Removing LWB

To remove the *lwb* rpm from your system.

Sign on as root user

```
# su
Password:  <enter root password>
```

Remove the *lwb* rpm

```
# rpm -e lwb
```

3.5 Miscellaneous

- In order to use the LWB plot3d and multi-scope modules, you will need an X server that supports the GLX extension. Some graphics boards (NVIDIA, ATI, Intel) support GLX (OpenGL) in hardware, which dramatically improve the performance of LWB when running with plot3d modules.

- You do not need any data acquisition hardware to run LWB. You can simulate data acquisition signals by using the signal generator module. You can then display the resulting signal on a scope and /or record it to a disk file.

4. Configuration:

4.1 LCAIO (PCI-16AIO) – 32/16Ch. 16Bit Analog Input, 4Ch Analog output

The ***lcaio*** driver is designed to support IRQ sharing. If this device's IRQ is being shared by another device then this driver's performance could be compromised. Hence, as far as possible, move this board to a PCI slot whose IRQ is not being shared with other devices. A ***'lspci -v'*** command can be used to determine the IRQs of various devices in the system.

4.1.1

The *lcaio* driver release notes describes how to extract the driver and install onto you iHawk system. Following are configuration considerations as related to LWB.

Building the lcaio driver

The ***lcaio*** driver files will be extracted into the current ***/usr/local/CCUR/drivers*** directory from the CDROM drive.

Next, you will need to build the driver:

```
> === as root ===
  > cd /usr/local/CCUR/drivers/lcaio
  > make       (build the driver and tests)
```

(P.S. if you only wish to build the driver, you can enter the 'make driver' command instead).

NOTE!! If the ***make*** fails with some module version related error, then you will need to follow the directions *(see below)* ***"Building driver on a currently running RedHawk kernel".*** Once done, you will then need to re-make the driver as described above.

NOTE!! Ignore the following warning if it appears:

```
*** Warning: Overriding SUBDIRS on the command line can
 cause
***          inconsistencies
```

After the driver is built, you will need to install the driver:

```
> === as root ===
> cd /usr/local/CCUR/drivers/lcaio
> make install  (install the driver software and man page)
```

Once the driver is installed, you will need to ***load*** it before any access to the PMC-16AIO board can be made. At this time, make sure that the hardware is physically installed in the machine, otherwise the load will fail.

```
> === log in as either user or root ===
> lspci -v  (display all the PCI devices in the system)
```

Now, look for the "Bridge: PLX Technology, Inc. 9080 (rev 03)" bridge with a device id of 2402 (i.e. the PCI-16AIO device).

If the above device is not displayed by the ***lspci*** command, the device has not been properly installed in the system. Make sure that the device has been correctly installed and running prior to proceeding with the next step.

Once the driver and the hardware have been successfully installed, the driver can be loaded into the running kernel:

```
> === as root ===
> cd /usr/local/CCUR/drivers/lcaio
> make load        (load the lcaio driver)
```

### Ignore any warnings about licensing ###

NOTE!! If the ***make*** fails with 'Invalid module format' or similar error, then you will need to
follow the directions *(see below)* ***"Building driver on a currently running RedHawk kernel"*****.** Once done, you will then need to re-make the driver as described above.

(A "... successfully loaded" message should appear if the driver was loaded successfully).

```
> lsmod            (this command will display the lcaio driver)
> dmesg            (should display LCAIO driver successfully inserted)
                   (Ignore any license related warnings)
```

4.1.2 Edit the lwb_clk.devs file.

The */usr/lib/LWB/lwb_clk.devs* file contains a list of clocks that will be scanned during system initialization. The first successfully opened clock will be selected as the internal Master clock used by LWB.

Example File:

```
************************************
***        Lcaio Clocks          ***
************************************
/dev/vdaq/lcck0
/dev/vdaq/lcck1
/dev/null
/dev/null
/dev/null
************************************
***        Audio Clocks          ***
************************************
/dev/vdaq/audiock0
/dev/null
/dev/null
/dev/null
/dev/null
```

In the above `/usr/lib/LWB/lwb_clk.devs` file, lwb will attempt to open the /dev/vdaq/lcck0 device first, if not successful, the /dev/vdaq/audiock0 will be tried next etc.

To test the installation, login and invoke LWB according to the procedure detailed in Section 3.

4.2 User Environment Variables

The user choose to modify LWB related shell environment variables as follows:

LWB_DEVS_PATH	- Specifies the first directory search path used to search for LWB related files.
LD_LIBRARY_PATH	- Specify additional directories to search for dynamic libraries
DISPLAY	- Changes the default X-Server display that LWB will be run from.

Note: The lwb.sh script no longer exists in LWB. The executable *lwb* in */usr/lib/LWB/bin* is invoked directly. The environment UIDPATH and LD_LIBRARY_PATH are setup by the shell on login. These environment variables (DISPLAY, UIDPATH) may be changed by the user as necessary although UIDPATH needs to be pointing to the binary Motif resource files. Those resources can only be changed by recompiling the *.uil files. Because of the way the modules are defined, a modified version of the uil compiler is needed. The one provided by default with OpenMotif or lesstif will not work.

4.3 Miscellaneous

Documentation for LWB consists of the LWB Release Notes (*this document*) and online help pages.

Right clicking the mouse on a module invokes the setup dialog (if any for the module). Once the dialog is open, every module has help button that can be clicked to open the corresponding html help file in the browser. Additionally, you may select the '?' with the left mouse button on the corner of a module to invoke the html help browser.

Most LWB files reside in */usr/lib/LWB*.

The directory */usr/lib/LWB/ucmd_shells* contains sample shell scripts for use with the UNIX Command module.

The directory */usr/lib/LWB/user_funcs* contains sample C programs for use with the User Function module. Refer to the Laboratory Workbench User's Guide for information on compiling and running these examples.

The LWB file I/O support library liblwb.so is a dynamically loadable driver that resides in /usr/lib/LWB/lib.

5 New features

This section describes new features added in this release.

LWB V4.1.1 is a new release of this product providing many changes to enhance and stabilize the Linux implementation of LWB.

5.1 New LWB 4.1.0 Features

The following features have been added to the **LWB 4.1.0** release:

Usability:

- Elimination of the Control Panel Window – Control panel options and features are now included in LWB's Instrument window allowing for fast access of instrument controls from a single window. All user interactions now take place in the Instrument window. A Control Button Bar in the Instrument Window allows for fast access to file operations, instrument control (Setup, Run, Stop, Abort), recording control and buffer control.
- Block Operations - Now multiple modules can be selected by drawing a selection rectangle. The modules contained within the rectangle can be moved (click & drag), deleted (shift click), duplicated (click, drag & control release) or copied to the copy buffer (Ctrl C). Modules previously copied with Ctrl-C (Edit->copy) can be pasted with Ctrl-V.
  Modules can also be aligned (left, right, top, bottom) within the selection rectangle
- Scrolling - The main instrument window is scrolled when moving single or block of modules outside the current viewing area.
- Tile/Raise - Added feature to tile / raise the top-level plot window to make management of numerous display window easier. Option provided to keep instrument window "on top" of other LWB windows (if supported by your window manager).
- Module Configuration – Simply right clicking the module within the instrument window or the module display such as a scope or plot can now configure the module. Dynamic configuration can also be performed on modules in a running instrument. Any items that cannot be updated while running are grayed out indicating they are not modifiable.
- Prompts – LWB now prompts before leaving an unsaved model, overwriting an existing model and when attempting to exit a running model.
- Extended Help – All modules have a help man page accessible directly from the module in the instrument window.
- File operations – LWB now provides the ability to insert an existing model into an instrument as well as save a model via a "Save As" mechanism.
- Instrument Window – Size of window increased to hold more complex instruments. Window will now scroll when moving a module outside viewing area.
- Environment Variable – A new environment variable LWB_DEVS_PATH has been introduced which can be set by a LWB user to specify the directory where devs reside. Useful for systems with multiple LWB users.
- Delete – Modules & wire connections can now be deleted via a keyboard shortcut (shift + click). If module can be deleted by holding the shift key and select the module to be deleted. If the output connection of a module is selected, all wire connections from the module will be deleted. Similarly, if the input wire connection is selected, the input to that module will be deleted.

Performance Enhancements:

- IPP (Intel Performance Primitives) library software optimizations are now used as the default for processing modules greatly reducing time spent for cpu intensive operations. IPP provided without additional charge.
- LWB no longer “spins busy” while idle.
- Typical model utilizes <10% of CPU.
- Support for > 32 modules of any one type.
- Optimized LCAI/LCAO I/O performance. Frequency setting has been enhanced to provide frequencies with greater accuracy.
- Recording data to disk is now performed via an independent thread that prevents the running model blocking on the disk write.
- LCAI and LCAO modules can now be utilized within the same instrument.

New Modules

Visualization:

- Plot3D (New Module) – Waterfall type plot added that provides easy configuration while the instrument is running. Log and linear scaling is available on the X & Y axis for this plot type.
- Multi Scope (New Module) – A new scope type has been added that displays multi-channel data streams in an individual scope window. Up to 4 plots with independent axis can be displayed in each scope, up to 32 channels can be displayed as individual scopes in a single window. Most visual elements are user configurable: color, axis, display format etc... A zoom feature allows for zooming in on the plot display by a left mouse click to drag a box over the area to be magnified. Note that zooming can only occur as long as there are at least two sample points present in the zoomed area. If two sample points are not present, the plot will appear blank.

Processing:

- FIR/IIR Filter – Calculate FIR and IIR filter coefficients according to user specified low/high frequency cutoff and noise reduction.
- Logical Trigger - Detects whether a trigger condition on the module input has been satisfied and output a logical value.
- Zero Crossing - Can count the number of transition on a multi-channel data stream connected to its input.

Demonstration Module:

- Audio In/Out – Modules to read analog data from the microphone or line in on the sound card and output analog data to the sound card.
  Note: This module, while fully functional, is provided for demonstration purposes only.

Module Enhancements:

- Record/Playback
  - Recording of data is now controlled by either a push button on the Control Button Bar or by a logical value present on the “enable” input of the record module. A circular buffer (available only when record mode is binary) has been added to allow recording of data acquired by the instrument before the record button has been depress or a logical true

value has been presented to the enable input. Data can now be scaled when saving in ASCII format.

- Recording of data performance improvements. Separate thread is created to write data to disk preventing running model from blocking on disk I/O.
- Playback of data can be controlled via "VCR" like controls on the modules pop-up menu. Playback rate controls exist for continuous, single-step, time delay, and simulation modes. Automatic and reverse replay is also available.
- Playback also allows browsing data and jumping to any position within a recorded session with a simple slider movement.
- Playback module now provides the capability to display data message timestamps. In the simulation mode, the playback will run based on the timestamp of individual records.
- Data record files now allowed past 2Gb boundary.

- Digital Meter – The digital meter has been greatly enhanced. Multiple channels can now be displayed in a single top-level window. The format for the numerical data and the description can be modified via the module's configuration menu.
- Signal Generator – Can change frequency while instrument is running. Can add NOISE to generated wave and control the rate of wave generation.
- Constant – Constant value can now be entered directly into a pop-up dialog.
- Low Pass Filter – Can now enter frequency value directly into pop-up dialog.
- Shell Command – The "Unix Command" has changed name to the Shell Command to be more representative of its functionality.
- Multiplexer – Now provides the capability to perform Async operations. In the non-async operation, no message is sent out until all message have reached connected inputs. In the Async mode, message arriving on any of the inputs will be multiplexed out immediately without waiting for messages from other connected inputs to arrive. In Async mode, inputs that did not have pending messages bill be NULL filled.

5.2 LWB 4.1.1 Features

The following features have been added to the **LWB 4.1.1** release:

- IVP (Installation Verification Procedure) initial release version of LWB
- Separate rpm files for full LWB w/ I/O and IVP version w/o I/O
- New installation script – *ccur_install* for a more automated installation process.
- Enhanced Display Module - MScope: Faster processing through use of IPP library to demultiplex and convert channel data.
- New Process Module - Channel Selector: Provides ability to pick and reorder a combination of channels in a multiplexed data stream.
- New Process Module - Statistics: Elementary statistical calculation on data buffer (Average, Standard Deviation, Maximum, Minimum, & Median)
- New Display Module - CrossSpectrum: Calculate Cross Spectrum / coherence / phase of 2 input data streams.
- New Display Module Campbell plot: display Campbell plot of multiplexed data stream.
- New IVP Module - RCIM Intr: read and time external interrupts received via RCIM
- New IVP Module - RCIM RTC: Program RCIM 's RTC to generate interrupts / read and time interrupts intervals.

6 LWB Corrections and Changes.

This section will be used to describe corrections and changes made beyond release 4.1.1.

7 Known Limitations & Issues

This section describes know issues and problems with LWB V4.1.1. If you should discover any problems not covered in this section, please report them to dac@ccur.com to report these issues.

7.1 Audio In/Out supported for demo purposes only.

7.2 User Functions: threshold and stripchart are provided as example User Functions with the sole intention of being an aid for the development of custom User Functions written by the end user of this product.

8 Additional Utilities (Japan Only)

This section describes new features added in this release.

LWB V4.1.0 was an initial release providing support for the 16-bit data acquisition products (and LCAIO), SP50, X11R6, and Open/Motif V2.2.2.

The file conversion utility, *uconv*, can convert a recorded LWB data file into other “universal” file formats. It allows user to specify the parameter of universal function type directly to convert recorded data from LWB read from standard input.

```
# uconv -H
usage: /usr/lib/LWB/bin/uconv [-h or -H][-v][-d][-s][-n][-t template][-f function]
LWB_FILE
        This command uses the -h,-H,-v,-d,-s,-n,-t and -f options as described
        below LWB_FILE is the name of the file that you want to convert.
        if you do not specify the LWB_FILE name, the command reads
        the LWB_FILE name from standard input.

        -h              print out this message
        -H              print out long HELP message
        -v              print out /usr/lib/LWB/bin/uconv version info
        -s              send the result to the standard output
        -n              no output dataset 164
        -d              output forces the double precision(or complex)
        -t              template file(normally /usr/lib/LWB/universal.defs)
        -f function     convert to the function type
                        The legal 'function' for option -f are:
                         AUTO_Correlation(7)
                         AUTO_Spectrum(2)
                         COHErence(6)
                         CROSS_Correlation(8)
                         CROSS_Spectrum(3)
                         CUmulative_frequency_distribution(13)
                         EIGENVAlue(22)
                         EIGENVEctor(23)
                         ESD or ENergy_spectral_density(10)
                         FInite_impulse_responce filter(25)
                         FOrce_pattern(19)
                         FRequency_response_function(4)
                         General(0)
                         MOde_idicator_function(18)
                         MUltiple_chherence(26)
                         Orbit(17)
                         PARTIAL_Coherence(21)
                         PARTIAL_Power(20)
                         PEaks_valley(14)
                         PSD or POwer_spectral_density(9)
                         PRobability_density_function(11)
                         SHock_responce_spectrum(24)
                         SPectrum(12)
                         STRAin_cycles(16)
                         STREss_cycles(15)
                         TIme_response(1)
                         TRansmissibility(5)
                         Unknown(0)
When specifying the function type, the letters in upper case are required; underscores
are ignored. However, this command accepts either upper or lower case.
```

The following questions were compiled from Customer Service notes and general discussions with users.

? Why do some modules use data in engineering units, and others use raw data?

In general, computational modules perform their functions on raw data to prevent a loss of accuracy due to conversion errors. Display modules attempt to convert the raw data to some meaningful values by applying the scaling and engineering units conversions. Certain modules must transform the data types to provide meaningful output. The FFT/IFFT module output must be a complex number when performing a forward FFT, the input must be complex for the inverse FFT.

? If non-linear arithmetic is performed on the data in an Arithmetic module, the scaling factors are not transformed.

The output scaling factors will be incorrect unless they are modified by the user with the Message Editor module. LWB does not automatically perform this scaling transformation in these modules, as the intention of the user is impossible to know, and maximum flexibility is provided by the Message Editor.

? I get a message 'Incomplete transfer on path 6', but I don't understand what changes I should make to resolve the error.

This error message is generated by the data acquisition driver software indicating to the calling routine that the driver was unable to complete the transfer of data to the host memory buffers. This usually is the result of a processing step requiring more time than the sampling rate allows. Lowering the frame rate should eliminate the error message.
If a high frame rate is required, some adjustment to the number and sizes of the data buffers may allow the transfer to occur, at least for a finite time-period.
Eventually, a continuous transfer at a rate faster than the processing rate will swamp the pool of free data buffers and causes an error condition.

? Why do different modules produce output, or require input, in different data types?

Most LWB modules will handle data in all four standard LWB data types, (short, int, float, and complex). Restrictions are applied where necessary. Certain modules, such as the Analog Input and Analog Output Modules, can only handle short integers due to the restrictions of the hardware. Other modules, such as FFT/IFFT, restrict their data types depending upon the function. For example, only the complex type data makes sense as the input of the Inverse FFT. Other data types must first be converted to the complex type with the Type Conversion module. Multiplexer requires same data types and buffer sizes on all connected inputs.

? How can the btoa (binary to ASCII) utility be useful with data files?

The lwbtoa utility allows the conversion of data files from binary to ASCII form. It is usually more convenient to use the ASCII type files in a user-written program.