



X-line X10 / X10i / X15 - Quick Start Guide

Document No. 80-18392 Issue 4

HEBER LTD

Current Issue: - Issue 4 – 6th November 2007

Previous Issue: - Issue 3 – 9th October 2007
Issue 2 – 12th October 2006
Issue 1 – 15th December 2005

©HEBER Ltd. 2007. This document and the information contained therein is the intellectual property of HEBER Ltd and must not be disclosed to a third party without consent. Copies may be made only if they are in full and unmodified.

HEBER LTD

Belvedere Mill
Chalford
Stroud
Gloucestershire
GL6 8NT
England

Tel: +44 (0) 1453 886000
Fax: +44 (0) 1453 885013
Email: support@heber.co.uk
<http://www.heber.co.uk>

CONTENTS

| | | |
|----------|--|----------|
| 1 | IMPORTANT INFORMATION | 1 |
| 1.1 | X10 CUSTOMERS | 1 |
| 1.2 | X15 | 1 |
| 2 | INSTALLING THE SOFTWARE..... | 2 |
| 2.1 | WINDOWS® 2000 AND WINDOWS® XP | 2 |
| 2.1.1 | <i>Installation.....</i> | 2 |
| 2.2 | LINUX..... | 2 |
| 2.2.1 | <i>Extracting the Development Kit</i> | 2 |
| 2.2.2 | <i>Building the kernel module under Linux Kernel Version 2.6.*</i> | 2 |
| 2.2.3 | <i>Installing the fflyusb Shared Library.....</i> | 3 |
| 2.2.4 | <i>Adding X-line HotPlug Support</i> | 3 |
| 2.2.5 | <i>Starting and closing the X-line Driver</i> | 3 |
| 3 | DEMONSTRATION PROGRAMS | 5 |
| 3.1 | COMPILING A SAMPLE PROGRAM..... | 6 |
| 3.1.1 | <i>Building under Windows®</i> | 6 |
| 3.1.2 | <i>Building under Linux</i> | 6 |
| 4 | CONNECTING DEVICES TO THE X-LINE BOARD..... | 7 |
| 4.1 | P2, I/O 1..... | 8 |
| 4.2 | P3, I/O 2..... | 9 |
| 4.3 | P4, ccTALK CHANNELS A & B..... | 10 |
| 4.4 | P6, I/O 3..... | 11 |
| 4.5 | P7, AUDIO IN..... | 11 |

LIST OF TABLES

| | |
|--|----|
| Table 1 - Table of Connectors | 7 |
| Table 2 – P2, I/O 1 Connector..... | 8 |
| Table 3 – P3, I/O 2 Connector..... | 9 |
| Table 4 - P4, ccTalk Channels A & B..... | 10 |
| Table 5 – P6, I/O 3 Connector..... | 11 |
| Table 6 – P7, Audio-In Connector | 11 |

This page intentionally left blank.

1 IMPORTANT INFORMATION

1.1 X10 Customers

This section is intended for current users of the X10 board. It can be safely ignored by all new X10i customers.

The X10 board has been enhanced and replaced by the X10i. This was necessary for two reasons:

1. The 8051 chip utilised on the X10 was discontinued.
2. Due to the EU RoHS directive, which comes into operation in 2006, it became necessary to ensure that all components on the X10 board are lead-free. The X10 previously included several components containing lead.

The change provided the opportunity to improve the X10 by adding new functionality. The processor on the X10i is also faster so there are performance improvements. All X10 and X10i differences are outlined in the X-line X10 / X10i / X15 Software User Manual, 80-18374.

1.2 X15

This section is intended for all customers that use the X10 or X10i

Heber has introduced an additional product into its range of X-line boards; this product is the X15. As the X15 is based upon the core design of the X10i product, and it shares much of the functionality of the X10i a single development kit CD package has been produced.

This allows users of the X10i and X15 products to install one set of drivers and use one set of manuals for both products. Where there are differences in the two products, these shall be stated at the appropriate place in the documentation.

2 INSTALLING THE SOFTWARE

2.1 Windows® 2000 and Windows® XP

2.1.1 Installation

Double-click on 'setup.exe' and follow the on screen instructions to complete the installation. The final screen asks for permission to install the drivers onto the current PC – to do this ensure that the “Install the X-line drivers now” option is ticked and click “Finish”.

When the X-line board is plugged in, Windows® might ask for the location of the X-line drivers. Direct Windows® to look in the Xline driver folder – by default this is C:\Heber\Xline\driver. It should only be necessary to perform this task once.

The installation creates several directories beneath the install directory (e.g. c:\Heber\Xline):

- **bin:** The X-line Windows diagnostic program 'x10diag' is located in here.
- **docs:** All X-line documentation is located within this directory. The documents are saved in Adobe PDF format. The PDF reader can be obtained from the Adobe website at <http://www.adobe.com>.
- **driver:** The Windows drivers required to drive the X-line boards are located in this directory.
- **include:** Contains the API header files. Only 'fflyusb.h' and 'unlockio.h' need to be included in your X-line programs.
- **lib:** Contains the API library files. These must be linked against your X-line programs.
- **samples:** Contains various demonstration programs along with the corresponding source code.

A 'Heber' entry will also be added to your 'Start' menu. This contains shortcuts to several useful X-line features, e.g. documentation, diagnostics and demonstration source code.

2.2 Linux

2.2.1 Extracting the Development Kit

First find a suitable location to install the X-line Development Kit (/usr/src is recommended). Move to the required directory and then unzip the supplied tar archive:

- `cd /usr/src`
- `tar -xzvf xlinedevkit.tar.gz`

For the remainder of this guide it will be assumed that the X-line Development Kit is placed in directory /usr/src.

2.2.2 Building the kernel module under Linux Kernel Version 2.6.*

Ensure that the path /usr/src/linux contains the kernel sources for the kernel version you intend to use. These sources must be configured by issuing the following commands:

- `cd /usr/src/linux`
- `make config` (Note this can be replaced with “make menuconfig” or “make oldconfig”)
- `make`

To build the X-line drivers under a 2.6.* kernel then follow these instructions:

- `cd /usr/src/xlinedevkit/driver`

- `make -C /usr/src/linux SUBDIRS=`pwd` modules`

If the build succeeds a file called X10.ko will have been created in the local directory. To install this file issue the command:

- `make -C /usr/src/linux SUBDIRS=`pwd` modules_install`
- `depmod` (Note the `depmod` command may not be needed - see `man depmod` for details.)

This should copy the newly created X10.ko file to its correct place in the `/lib/modules` directory tree and make the module loader aware of the X10 module.

2.2.3 Installing the fflyusb Shared Library

To access the X-line board a user library `fflyusb.so` is supplied - this needs to be copied to a suitable library directory (e.g. `/usr/lib` or `/usr/local/lib`). A symbolic link should also be created:

- `cp -a fflyusb.so /usr/lib`
- `ln -s /usr/lib/fflyusb.so /usr/lib/libfflyusb.so`

This library must be linked with your code.

2.2.4 Adding X-line HotPlug Support

The kernel driver has been built to use the 2.6 `sysfs` file system. The driver creates a new `sysfs` class (in `/sys/class`) called `xline`. When an X-line board is detected the driver adds a new entry to this class. Each entry contains a symbolic link to the USB bus entry in `/sys` associated with that particular device. The `fflyusb` library uses this to detect whether an X10, X10i or X15 is connected and whether it is running in full or high speed.

Most 2.6 based systems mount a ram based file system over `/dev` early in the boot process. A user space demon `udev` then walks the `sysfs` tree and populates this new `/dev` with entries specific to the hardware present.

Because the X-line driver adds entries to `/sys`, if `udev` is being used then `udev` can create `/dev` entries. To do this add the following rule to the `udev` rules file (i.e. `/etc/udev/rules.d/50-udev.rules`).

- `KERNEL="X10_*",NAME="%k",MODE="666"`

The `/dev` entry can be assigned to a group by adding a `GROUP="my group name"` field to the line above.

If this line is added `udev` will create an `xline` `/dev` entry when the board is plugged in and delete it when it is removed (this process can take up to 10 seconds).

2.2.5 Starting and closing the X-line Driver

If the X-line driver has been installed to the correct `/lib/modules` directory and the `depmod` command has been executed then the X10 kernel module can be loaded as follows:

- `modprobe X10`

An alternative mechanism that doesn't demand the driver to be installed in the correct `/lib/module` directory is to issue the command.

- `insmod X10.ko`

Once the X10 driver is loaded and a board is plugged in, it takes a couple of seconds for the board to download the required firmware and initialise. In order to check that the board is ready to go you can

look at the kernel log by using the command 'dmesg'. One of the last entries should read "Heber X* Board ready."

To remove the X10 driver, issue the following command:

- `rmmod X10`

3 DEMONSTRATION PROGRAMS

A number of demonstration programs are supplied with the X-line Development Kit. They give examples of how to use the API calls described in the software user manual.

Within the “bin” directory is an executable diagnostic program named **X10Diag.exe** as well as a XML script entitled `cctalk_4_1`. This easy to use program will allow you to test a subset of the X-line’s functionality. Please note that this program is currently only available under Windows®.

When ready to write software for the board, the user should refer to the “sample” directory. Twelve demonstration programs are provided each demonstrating specific features of the X-line API and it is hoped that the supplied source code will prove useful to customers. The programs are identical for both Windows® and Linux versions.

Once running, each program responds to key commands. The control keys are displayed when each program is first executed. Although the control keys are generally different for each program, all programs can be terminated by typing “C”.

Following is a list of all supplied demonstration programs along with a brief description:

- **cctalkdemo**: This program demonstrates the ccTalk mode 1 operation. A pre-defined message is sent to any connected ccTalk devices and it is possible to read or empty the ccTalk buffers.
- **eeepromdemo**: This demonstrates EEPROM reading and writing.
- **fadedemo**: This program demonstrates lamp fading by slowly turning on all output lights and then turning them off again. This results in a “caterpillar” effect. The fade speed can be controlled by turning on or off IP16 to IP23 (the speed is increased by turning on more inputs).
- **Inpmuxdemo**: This demonstrates input multiplexing. There are four channels (corresponding to OP12-OP15) each containing 24 inputs (corresponding to IP0-IP23). The X-line board internally multiplexes these inputs and outputs together to provide 96 inputs.
- **iodemo**: This demonstrates various IO functionality.
- **paralleldemo**: This demonstrates the ability to drive parallel devices (e.g. coin hoppers).
- **randomdemo**: This demonstrates the random number generation facility (this is not available under X15).
- **reels**: This demonstrates reel spinning.
- **serialedemo**: Demonstrates serial communications. A good way to test this is to connect an RS232 crossover cable between an X-line board and a PC and then use a program like HyperTerminal to also send and receive data.
- **spidedemo**: This demonstrates communication with a Starpoint Electronic Counter (SEC) device using the SPI protocol. The program displays “HEBER” plus a number (as incremented each time you press a key) on the SEC device.
- **sramdemo**: This demonstrates SRAM reading and writing.
- **timedemo**: This demonstrates communication with the PIC. Each time a key is pressed then a new Real Time Clock read will occur and it will be displayed.
- **timedemo**: This demonstrates communication with the PIC. Each time a key is pressed then a new Real Time Clock read will occur and it will be displayed.
- **authenticatedemo**: This demonstrates the unlocking procedure for the X15 board only.

3.1 Compiling a sample program

3.1.1 *Building under Windows®*

As an example, the reel project will be described but all samples will follow the same steps. Explanations apply to Visual C++ version 6 but similar steps should be followed for other versions.

- Copy the reel folder into your working directory.
- Create a new project in Visual C++ and add "reels.cpp" to the source file list.
- Access the setting of your project (project\setting). Select "C/C++" tab and pre-processor drop down menu from there. In the additional include directory, insert the path of X-line include files (C:\heber\Xline\include as well as C:\heber\Xline\samples)
- Select the "Link" tab and the input drop down menu from there. In the additional library path, insert the path of the X-line library files. (C:\heber\Xline\lib) and specify which library you need to identify in the project option (fflyusb.lib unlockio.lib)
- Press 'OK' to change settings. Rebuilding the project should now work correctly.

3.1.2 *Building under Linux*

Within the 'samples' directory is a file called 'makefile'. This makefile builds all of the demonstration program and can be altered to suite your needs. In order to build the programs, type 'make'.

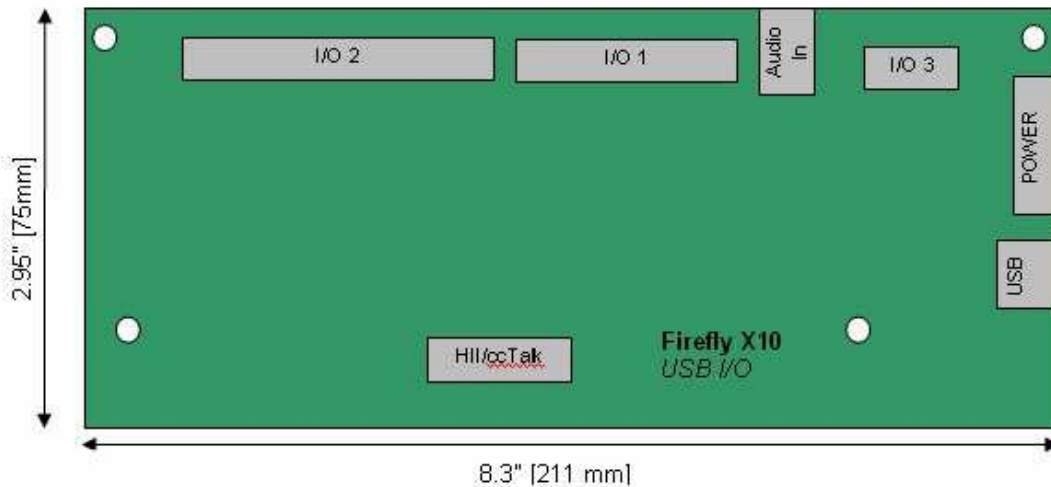
4 CONNECTING DEVICES TO THE X-LINE BOARD

The X-line board is fitted with the following connectors:

Table 1 - Table of Connectors

| Ident | Type | Function |
|-------|--------------------------|--|
| P1A | USB Type B | USB (connectors in parallel) |
| P2 | 34W Header | I/O 1 |
| P3 | 50W Header | I/O 2 |
| P4 | 20W Header | ccTalk Channels A & B |
| P5 | 4W AMP | +12V Power in. (Hard Disk style connector) |
| P6 | 16W Header | I/O 3 |
| P7 | 3.5mm Stereo Jack Socket | Audio In |

The connectors on the X-line board are arranged as follows:



Note:

The X10, X10i and X15 boards have the same set of connectors which are fully pin compatible, however the layout of the connectors is different on the X15 board.

4.1 P2, I/O 1

This is a 34w Header.

Table 2 – P2, I/O 1 Connector

Reference: P2
Type: 34W Header
Description: I/O 1

| | | | |
|----------------------------------|----|----|----------------------------------|
| Open Drain Output OP0/Input IP16 | 1 | 2 | Open Drain Output OP1/Input IP17 |
| Open Drain Output OP2/Input IP18 | 3 | 4 | Open Drain Output OP3/Input IP19 |
| Open Drain Output OP4/Input IP20 | 5 | 6 | Open Drain Output OP5/Input IP21 |
| Open Drain Output OP6/Input IP22 | 7 | 8 | Open Drain Output OP7/Input IP23 |
| Open Drain Output OP8 | 9 | 10 | Open Drain Output OP9 |
| Open Drain Output OP10 | 11 | 12 | Open Drain Output OP11 |
| Open Drain Output OP12 | 13 | 14 | Open Drain Output OP13 |
| Open Drain Output OP14 | 15 | 16 | Open Drain Output OP15 |
| Input IP0 | 17 | 18 | Input IP1 |
| Input IP2 | 19 | 20 | Input IP3 |
| Input IP4 | 21 | 22 | Input IP5 |
| +12V Current Sensed | 23 | 24 | +12V Power |
| Ground (0V) | 25 | 26 | Ground (0V) |
| Loudspeaker (left)+ | 27 | 28 | Loudspeaker (right)+ |
| Loudspeaker (left)- | 29 | 30 | Loudspeaker (right)- |
| Ground (0V) | 31 | 32 | Ground (0V) |
| Left Audio Line In | 33 | 34 | Right Audio Line In |

4.2 P3, I/O 2

This is a 50w Header. It may be fitted with a ribbon cable assembly to jump to a 50W 'D' Type on an I/O panel.

The high current outputs should use all three connections if the load will draw a high current. Otherwise, only one of the connections needs to be made. Similarly, sufficient ground connections should be used to meet the maximum load current expected.

Table 3 – P3, I/O 2 Connector

Reference: P3
Type: 50W Header
Description: I/O 2

| | | | |
|------------------------------|----|----|--------------------------|
| Open Drain Output OP16 | 1 | 2 | Open Drain Output OP17 |
| Open Drain Output OP18 | 3 | 4 | Open Drain Output OP19 |
| Open Drain Output OP20 | 5 | 6 | Open Drain Output OP21 |
| Open Drain Output OP22 | 7 | 8 | Open Drain Output OP23 |
| Open Drain Output OP24 | 9 | 10 | Open Drain Output OP25 |
| Open Drain Output OP26 | 11 | 12 | Open Drain Output OP27 |
| High Current Output OP28 | 13 | 14 | High Current Output OP28 |
| High Current Output OP29 | 15 | 16 | High Current Output OP28 |
| High Current Output OP29 | 17 | 18 | High Current Output OP29 |
| High Current Output OP30 | 19 | 20 | High Current Output OP30 |
| High Current Output OP31 | 21 | 22 | High Current Output OP30 |
| High Current Output OP31 | 23 | 24 | High Current Output OP31 |
| Input IP6 | 25 | 26 | Input IP7 |
| Input IP8 | 27 | 28 | Input IP9 |
| Input IP10 | 29 | 30 | Input IP11 |
| Input IP12 | 31 | 32 | Input IP13 |
| Input IP14 | 33 | 34 | Input IP15 |
| +12V | 35 | 36 | +12V |
| +12V | 37 | 38 | +12V |
| Ground (0V) | 39 | 40 | Ground (0V) |
| Ground (0V) | 41 | 42 | Ground (0V) |
| Ground (0V) | 43 | 44 | Ground (0V) |
| Security Switch SW1-4 common | 45 | 46 | Ground (0V) |
| Security Switch SW 1 | 47 | 48 | Security Switch SW 2 |
| Security Switch SW 3 | 49 | 50 | Security Switch SW 4 |

4.3 P4, ccTalk Channels A & B

This connector provides two ccTalk interface channels. It may be fitted with a 20W IDC header and the 20W ribbon split to provide two Industry Standard 10W Connections.

Table 4 - P4, ccTalk Channels A & B

Reference: P4
Type: 20W Header
Description: ccTalk Channels A & B

| | | | | | |
|-----------------------------------|----------------------------------|----|----|-------------|-----------------------------------|
| <i>ccTalk</i> CHANNEL A | DATA Channel A | 1 | 2 | Ground (0V) | <i>ccTalk</i> CHANNEL A |
| | BUSY Channel A | 3 | 4 | Ground (0V) | |
| | RESET Channel A (*Output OP8) | 5 | 6 | | |
| | +12V Power | 7 | 8 | Ground (0V) | |
| | Ground (0V) | 9 | 10 | +12V Power | |
| <hr/> | | | | | |
| <i>ccTalk</i> CHANNEL B | DATA Channel B | 11 | 12 | Ground (0V) | <i>ccTalk</i> CHANNEL B |
| | BUSY Channel B | 13 | 14 | Ground (0V) | |
| | RESET Channel B (*Output OP9) | 15 | 16 | | |
| | +12V Power | 17 | 18 | Ground (0V) | |
| | Ground (0V) | 19 | 20 | +12V Power | |

Note: The ccTalk receiver is configured for +5V operation. If the interface is operating at +12V levels, then R32 should be removed and fitted in R33 position instead. If this is not done, the ccTalk interface will work but with slightly reduced noise immunity and will be pulled up to 5V by the X-line board. This is unlikely to cause any problems.

4.4 P6, I/O 3

This is a 16w Header.

Table 5 – P6, I/O 3 Connector

Reference: P6
Type: 16W Header
Description: I/O 3

| | | | |
|----------------------------|----|----|----------------------------|
| Ground (0V) | 1 | 2 | Ground (0V) |
| Serial RS232 Input RXD A | 3 | 4 | Serial RS232 Output TXD A |
| Serial RS232 Input CTS A | 5 | 6 | Serial RS232 Output RTS A |
| * Serial TTL Input RXD B | 7 | 8 | * Serial TTL Output TXD B |
| * +12V output | 9 | 10 | * -12V output |
| Auxiliary CMOS Output AUX0 | 11 | 12 | Auxiliary CMOS Output AUX1 |
| Auxiliary CMOS Output AUX2 | 13 | 14 | Auxiliary CMOS Output AUX3 |
| Auxiliary CMOS Output AUX4 | 15 | 16 | Auxiliary CMOS Output AUX5 |

+12V and –12V outputs are only available if a +12V source has been connected to P5. However, RS232 signal levels are generated on the X-line board and it is not necessary to connect a +12V power source to use RS232 signals. The +12V and –12V outputs are intended for use by a BACTA port.

4.5 P7, Audio In

This is 3.5mm stereo jack Socket. It duplicates the Audio Line In signals on Pins 33 & 34 of Connector P2 to allow connection to Firefly 700 via a standard jack-jack lead.

The jack ground connections must be connected at both ends of the cable to obtain lowest background (digital) noise.

Table 6 – P7, Audio-In Connector

Reference: P7
Type: 3.5mm stereo jack Socket
Description: Audio In