HEBER

X10i Quick Start Guide

Thank you for buying the powerful X10i real-time USB I/O controller from Heber. Please follow the following steps and examples to install the appropriate drivers and quickly set X10i to work for you.

What is X10i?

X10i is a universal, powerful and secure real-time controller that permits control over multiple inputs/outputs, via USB, from any PC system. Heber now offers support in other programming languages (like BlitzMax) to make it even easier for programmers, inventors and developers to connect with the outside world from their PC. X10i has already found a wide range of uses in many diverse markets such as:

- Security
- Gaming
- Kiosk
- Casino
- Vending
- Arcade
- Amusement
- Photo Booth
- Access Control
- Point of sale (POS)
- Products needing control of cash-in and cash-out
- Fire Systems and “Red Goods”
- Retail security and control
- Process control
- Home Automation
- Industrial machine control
- Cashless networked systems
- Robotics
- Force-feedback and telepresence
- Telemetry
- Sensor measurement for industry and test
- Functional test interface.
- CCTV
- Intruder Alarms
- Counter terror
- Medical
- Green technology interface (low-power)
- Education and Training
- IP and Network solutions
- For the inventor and system builder
- Hobby and Enthusiast
- Handheld and panel mount equipment
- Medical interfaces
- 3D plotter
- CAD/CAM, CNC interface – 3D Machining, 3D printing
- Motor speed control / Stepper motor
- Virtual Reality
- HID Interface

More information can be found here:

http://wwww.heber.ltd.uk/content.php?page=xline

And here:

http://wwww.heber.co.uk/pdfs/datasheets/x10i.pdf

Installing X10i Drivers.

Linux.

Extracting the Development Kit

First find a suitable location to install the XLine 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 XLine Development Kit is placed in directory /usr/src.

*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 XLine 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.

Installing the fflyusb Shared Library

To access the XLine 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.

Adding XLine 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 XLine 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.

For updates, new applications and lots more information about your X10i, regularly visit: ***www.heberforum.co.uk***

Heber Limited / Belvedere Mill / Chalford / Stroud / GL6 8NT

www.heber.co.uk

Because the XLine 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).

Starting and closing the XLine Driver

If the XLine 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

Windows XP.

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 XLine drivers now" option is ticked and click "**Finish**".

When the XLine board is plugged in, Windows® might ask for the location of the XLine 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 XLine Windows® diagnostic program 'x10diag' is located in here.
- `docs`: All XLine 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 XLine boards are located in this directory.
- `include:` Contains the API header files. Only 'fflyusb.h' and 'unlockio.h' need to be included in your XLine programs.
- `lib:` Contains the API library files. These must be linked against your XLine 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 XLine features, e.g. documentation, diagnostics and demonstration source code.

Windows Vista/7.

T.B.A

Installing BlitzMAX.

```
http://www.blitzbasic.com
```

Examples

The following Blitz source code shows creation and unlocking of an X10i board, it then goes on to interrogate USB speed multiple version and serial numbers and then closes the board releasing memory. This is one of the many samples supplied with the *Heber.X10i* module others include: Using I/O with read and write functions. Reading / writing to SRAM. Using the stepper motors. Read Write to SEC/SPI. Relock the X10i after a successful lock. cctalk demo. Serial comms. Reading multiplexed inputs. USB speed reporting. RTC. Preferred method of using OP0 - OP31 + AUX. Reading Inputs via callback.

```
SuperStrict
Framework       heber.x10i
Import          brl.retro

Global x10iBoard:TX10iBoard = New TX10iBoard.Create(
encryptedKeyGeneric )

If x10iBoard
    ' Read the version numbers.
    Print "board speed = " + x10iBoard.GetBoardSpeed()
    Print "Product Version = " + x10iBoard.
GetProductVersion()
    Print "Dll Version = " + x10iBoard.GetDllVersion()
    Print "8051 version = " + x10iBoard.Get8051Version()
    Print "PIC version = " + x10iBoard.GetPicVersion()
    Print "PIC serial number = " + x10iBoard.
GetPICSerialNumber()
    Local Dallas:Byte[8]
    Local CRCValid:Int
    x10iBoard.GetDallasSerialNumber( Dallas, Varptr
CRCValid )
    If CRCValid
        Print "Family code = " + Dallas[0]
        Print "Unique serial = " + Dallas[1] + Dallas[2] +
Dallas[3] + Dallas[4] + Dallas[5] + Dallas[6]
        Print "CRC = " + Dallas[7]
    Else
        Print "GetDallasSerialNumber returned invalid CRC
error"
    EndIf

    ' Housekeeping
    x10iBoard.Close()
    x10iboard = Null
    GCCollect

EndIf
End
```

For updates, new applications and lots more information about your X10i, regularly visit: ***www.heberforum.co.uk***

Document: 80-21233-1