

Aardvark Embedded Solutions

PayLink Linux User Guide

Issue	Date	Author	Comments
D1	19/04/2006	Len Meakin	Initial release for review.
D2	30/04/2006	Len Meakin	Improved installation script.
1.1	09/10/2008	Dave Bush	Addition of Firmware Loader
2.0	24/02/2011	Dave Bush	Update for Source Distribution

Table of Contents

1	Distribution	3
1.1	The Package.....	3
1.2	Automatic Installation.....	3
2	Manual Installation.....	4
2.1	Executables	4
2.2	Shared Libraries (DLL).....	4
3	Firmware Updating	5
4	AESWDriver	6
4.1	Reprogramming Kernel	6
5	Examples	7
5.1	Coin Read.....	7
5.2	Pay Out.....	7
5.3	Lumina Serial Number	7
5.4	AESDemo	8

1 Distribution

1.1 The Package

The Linux support software for Paylink is distributed as a Linux archive (PaylinkSourceDist.tar.gz) containing the source for all the PC support components. This has the following advantages (over the old distribution).

1. This distribution is totally as source files, and is compiled on the user's system. This means that all the software is guaranteed compatible, and it will even run on non-Intel platforms.
2. This distribution uses libusb to access USB devices. This is available as source, is included in this package, and also avoids using the deprecated usbfs package.
3. This distribution uses libftdi for the device specific access. . This is available as source, is included in this package, and is required to use the libusb package.

1.2 Automatic Installation

1. For all the installation steps, you should run as root (su).
2. This package installs into **the** `/usr/local/bin` directory. Before starting, you should ensure that this is on your standard path.

If they are not, one way of fixing this is to add the lines

```
# set up local paths for packages
pathmunge /usr/local/bin after
towards the end of /etc/profile
```

If you had to do this, then there is a good chance that your system is not set up for `/usr/local/lib`. To achieve this, add the line

```
/usr/local/lib
to the file /etc/ld.so.conf and update the cache by running
ldconfig
```

3. Unpack the distribution directory.

```
tar -zxvf PaylinkSourceDist.tar.gz
```
4. Swap to the new directory and issue the `./Install.sh` command.

The script should proceed to install libusb, libftdi and the AES Paylink support programs. If you do not have libusb / libftdi installed, the script will automatically install them. If they are already present, the script will offer to rebuild them from the source packages included in the Paylink distribution.

2 Manual Installation

For Paylink operation, the Linux system will require the following files, either in the indicated directories, or in directories that are functionally equivalent:

2.1 Executables

You need to have:

/usr/local/bin/AESWDriver	
/usr/local/bin/showtraf.sh	
/usr/local/bin/USBProgram	
/usr/local/bin/libusb-config	Needed to build libftdi

Also ensure the files are executable, if necessary by issuing the command:

```
# chmod +x /usr/local/bin/*
```

2.2 Shared Libraries (DLL)

You need to have the following:

```
/usr/local/lib/libaes_access.so
/usr/local/lib/libftdi.a
/usr/local/lib/libftdi.la
/usr/local/lib/libftdi.so
/usr/local/lib/libftdipp.a
/usr/local/lib/libftdipp.la
/usr/local/lib/libftdipp.so
/usr/local/lib/libusb.a
/usr/local/lib/libusb.la
/usr/local/lib/libusb.so
/usr/local/lib/libusbpp.a
/usr/local/lib/libusbpp.la
/usr/local/lib/libusbpp.so
```

Many of these will be symbolic links to more specific files in the same directory.

3 Firmware Updating

As well as the main release archive, there is a Firmware archive available - [PaylinkFirmware.tar.gz](#).

The main distribution has compiled the `USBProgram`, which performs the download itself, the firmware can therefore be distributed as text files, and used without requiring a compilation system on the target.

This Firmware archive contains a number of self contained scripts, each of which will load a new copy of the firmware into Paylink.

The naming conventions used in this folder are that:

- the first part is "Genoa", the internal name of the firmware
- the 2nd Part specifies the note reader protocol supported by the RJ45 serial connector, either:
 - The `ID003` protocol, originating with JCM
 - The `GPT V2.2` Serial protocol
 - The `Ardac 2` protocol for the MCL Ardac 2 acceptor.
 - a 4th, blank, option does not use the RJ45 for a protocol and outputs program diagnostics on the RJ45.
- the final part is `Vn-n-n-n` - the release number of the firmware.
- and there is then a suffix of `.sh`

This firmware updating program, when run, will automatically disable the Driver program (if running) and then download the new firmware to the Paylink. If there is any error in the programming, the program exits with a code of to exit with 1 for an error and 0 is everything is OK.

Options available for the download are as follows:

- Serial Number Specific mode (`-s <Serial>`): will cause the program to only update a Paylink set with the specific USB serial number (for us on a multiple Paylink installation.)
- Force mode (`-f`): regardless of the version already programmed, this mode will reprogram the Paylink.
- Check mode (`-c`): If this is set, then the programming will not happen if the Paylink already contains code with the same version number and program checksum.

4 AESWDriver

The driver runs in command line mode, no graphical user interface has been developed. However the driver is able to run in various modes:

Verbose mode (-v): certain information is printed to stdout (console window). This is analogical to running the Windows driver with the hide traffic option selected. Without the verbose mode flag set the driver will print very minimal information (if any).

ShowTraffic mode (-t): will display all data that is being sent / received to / from the PayLink interface (all messages are time stamped to millisecond resolution). This is analogical to running the Windows driver with the show traffic option selected. This option is only usually useful when no traffic at all appears to be taking place.

High Priority mode (-p): will cause the driver program to increase the priority of the communication process.

Serial Number Specific mode (-s <Serial>): will cause the driver program to only communicate with a Paylink set with the specific USB serial number (for us on a multiple Paylink installation.)

The ShowTraffic option may be set at driver start-up (-s) or by sending a SIGUSR1 signal to the driver. The driver will toggle the value of the ShowTraffic option upon receiving a SIGUSR1 signal. The supplied `showtraf.sh` shell script will perform this action for you.

The shared memory segment has been named `USBDriver` (or `USBDriver<Serial>` if the serial number option is being used), and can be viewed as part of the file system by issuing the following command (this is not recommended and should only be used for diagnosing driver issues).

```
# ls /dev/shm/USBDriver
```

To Start the AESWDriver at the root prompt issue the following command:

```
# AESWDriver -v
```

To force the driver to exit either send the driver a signal (other than SIGUSR1), or press CTRL+C on the console where the driver process was started.

Upon the driver starting it will attempt to open a link to the attached interface, if unsuccessful the driver will continue connection attempts every second until the device has been opened successfully.

4.1 Reprogramming Kernel

Firmware versions x-1-10-9 and later contain a reprogramming kernel. This greatly reduces the chances of a failed reprogramming exercise resulting a non-functional Paylink. If the Paylink already contains a reprogramming kernel then this is not re-written during an upgrade, meaning that following failures, the kernel will allow for further attempts.

Even if the Paylink does not already contain a reprogramming kernel, the first 5% of the full download contains the reprogramming kernel and, provided this is successful, subsequent failures during the same download are tolerated as above.

5 Examples

A number of example programs are included in the distribution that can be used / compiled "straight out of the box" to show that the installation is working.

5.1 *Coin Read*

An example CoinRead program has been provided, to compile this program issue the following commands:

```
# cd /root/LinuxPayLink/CoinRead
# make
# ./CoinRead
```

5.2 *Pay Out*

An example PayOut program has been provided, to compile this program issue the following commands:

```
# cd /root/LinuxPayLink/PayOut
# make
# ./PayOut
```

5.3 *Lumina Serial Number*

The Lumina Serial Number program has been provided, to compile this program issue the following commands:

```
# cd /root/LinuxPayLink/LuminaSerialNo
# make
# ./LuminaSerial
```

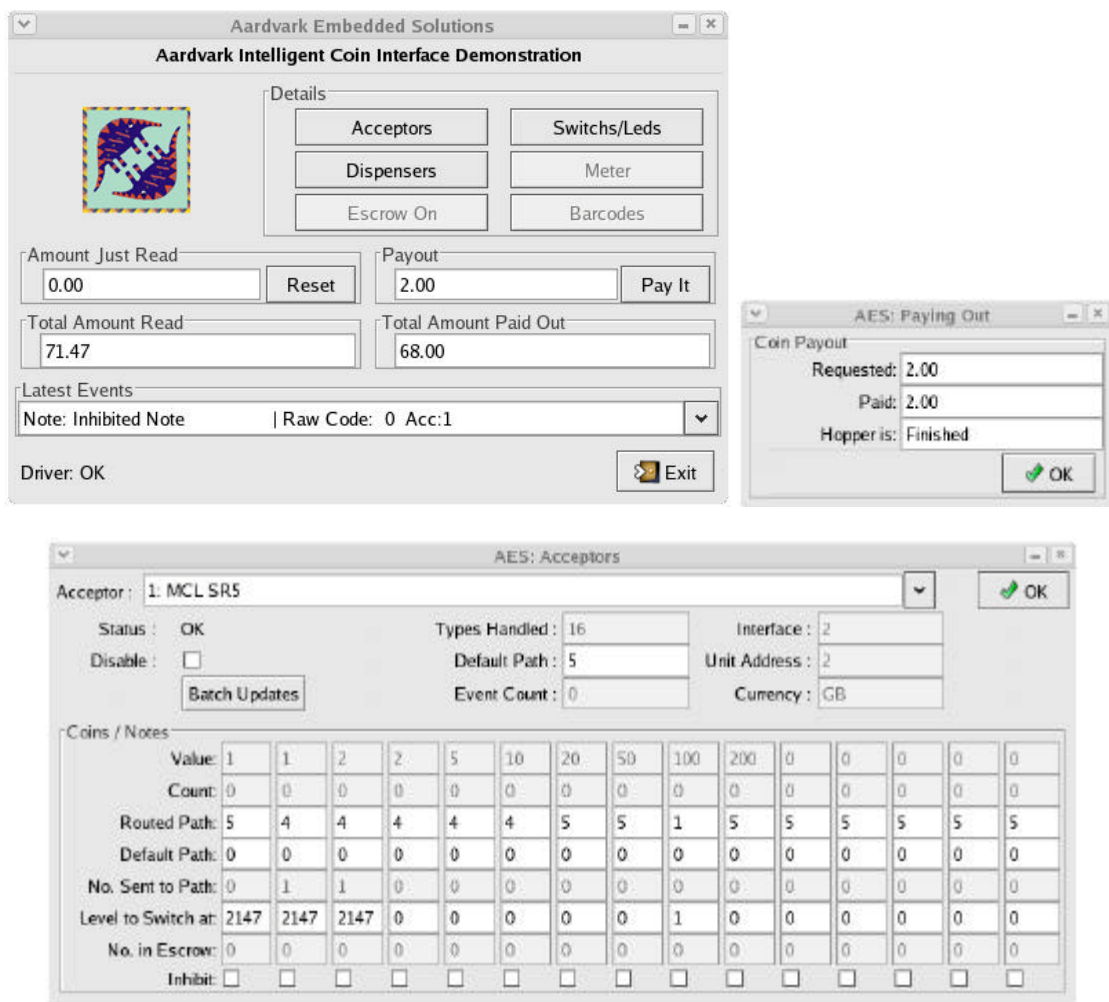
5.4 AESDemo

The AES Demo application has been provided, to compile this program issue the following commands:

```
# cd /root/LinuxPayLink/
# tar -zxvf AESDemo.tar.gz
# cd AESDemo
# ./configure
# make
# cd ./src
# ./AESDemo
```

This application uses the gtk+ graphical tool kit library. Please ensure that X Windows is running before issuing the above commands.

The application has been tested with GNOME 2.14 (Gentoo / FC5) and GNOME 2.8 (FC3).



AES: Switches / LEDs

Switch	LED
<input checked="" type="checkbox"/>	0
<input checked="" type="checkbox"/>	1
<input type="checkbox"/>	2
<input type="checkbox"/>	3
<input type="checkbox"/>	4
<input type="checkbox"/>	5
<input type="checkbox"/>	6
<input type="checkbox"/>	7
<input type="checkbox"/>	8
<input type="checkbox"/>	9
<input type="checkbox"/>	10
<input type="checkbox"/>	11
<input type="checkbox"/>	12
<input type="checkbox"/>	13
<input type="checkbox"/>	14
<input checked="" type="checkbox"/>	15

OK

AES: Dispensers

Dispensers

	Value	Address	Coins Paid	Contents	Status	Inhibit
MCL Serial Compact Hopper	100	3	0	-	Idle OK	<input type="checkbox"/>
						<input type="checkbox"/>
						<input type="checkbox"/>
						<input type="checkbox"/>
						<input type="checkbox"/>
						<input type="checkbox"/>
						<input type="checkbox"/>
						<input type="checkbox"/>
						<input type="checkbox"/>
						<input type="checkbox"/>
						<input type="checkbox"/>
						<input type="checkbox"/>
						<input type="checkbox"/>
						<input type="checkbox"/>
						<input type="checkbox"/>
						<input type="checkbox"/>
						<input type="checkbox"/>
						<input type="checkbox"/>
						<input type="checkbox"/>

OK