

## Milan / Paylink Firmware Version 4.1.11.8 Release Notice.

This is a **Full** (4) release of the Milan / Paylink Interface firmware - code version **1.11.8** and PC support software.

This release packages a **major** release of the PC support software. This software is also compatible with any 4.1.10.x firmware release, the PC software associated with earlier firmware releases will also work perfectly with this release of firmware.

The maintenance firmware release adds a few minor updates and clears problems discovered since the release of version 1.11.7

### ***PC code versions.***

The PC software associated with this release is:

Aesimhei.dll	Version 1.5.1.1 or later
Aesimhei.so	Version 1.2.0 or later (Linux)
AESWDriver.exe	Version 1.1.4.3 or later
Aesimhei.h	Dated 10/03/11 or later (See side effects below)

### ***Windows PC Software upgrade.***

- Paylink now has fully signed USB drivers, suitable for XP, Vista and Windows 7. The drivers are in the distribution and are also automatically available via Windows Update. Paylink will therefore install without any messages if the Internet is available when the Paylink module is plugged in.
- This package supports both the new series 1.11.x release of Paylink and, for people who do not wish to upgrade the Paylink firmware, the older 1.10.x series.
- Support of the programming environment dot NET (C# / VB.NET) and Java: Interface modules for both environments are in the SDK Folder
- The dot NET environment is illustrated by a simple test harness and by re-workings of the standard Paylink "Demo" program as an equivalent program using the new interface module in C# and VB.NET.
- The Java environment is illustrated by a simple test harness illustrating the use of the JNI Java module and its associated DLL.

### ***Linux PC Software upgrade.***

This is a **major** upgrade of the Linux support software.

- The old deprecated usbfs driver system has been replaced by the new libftdi system, which is available in source, and hence can be re-compiled for any Linux system, regardless of the hardware on which it is run.
- The Linux support software is now supplied *entirely* as source files. The PaylinkSourceDist.tar.gz package unpacks to a directory structure containing a script called Install.sh, which compiles and installs all the support components.
- The firmware upgrade programs for Linux are now shipped as executable source shell scripts in the PaylinkFirmware.tar.gz archive. Running one of these scripts carries out the programming function.
- The Linux installation also includes a Linux Java JNI environment which is illustrated by a simple included test harness.
- This package supports both the new series 1.11.x release of Paylink and, for people who do not wish to upgrade the Paylink firmware, the older 1.10.x series.

## **64 bit Operation.**

This release introduces support for 64 bit operating systems. The expectation is that all the utility programs will continue to be 32 bit as they have no need of the added memory, but that the dll / shared library is 64 bit compatible and can be used with 64 bit applications.

### **64 bit Windows.**

- The Windows part of the distribution contains the file **AESImhei64.dll**, which should be incorporated into the application build..
- Customers using 'C++' can use the associated library **AESImhei64.lib**.
- Customers using 64 bit Java will need to use **AESImhei64.dll** (leaving the name alone) but will also need to use a 64 bit Java interface dll. Java *insists* that this is called **AesImheiJava.dll** so they will need to ignore the 32 bit **AesImheiJava.dll** in the release and rename **AESImheiJava64.dll** to **AesImheiJava.dll**.
- Customers using 64 bit dot net will need to use **AESImhei64.dll** (leaving the name alone) but will also need to use a 64 bit dot net interface dll.

### **64 bit Linux**

64 bit Linux is complicated by the integer length specifications associated with 64 bit compilations. The size of a **long** data type is now 64 bits when compiling on a 64 bit Linux system, and 32 bits when compiled on all other systems (Linux 32, Windows 32 & 64, Paylink embedded.)

This change has meant that the interfaces have had to be re-declared as **int** rather than **long**. The overall result is that there are two Linux packages:

- PaylinkSourceDist.tar.gz - which uses the traditional **long** data type, and will not work for 64 compilations.
- PaylinkSourceDist64.tar.gz - which uses the **int** data type, and so will work both for 32 bit and 64 compilations.

## **Legacy Protocol support.**

- GPT V2.2 serial protocol acceptors are no longer supported
- Ardac II / WACS protocol acceptors are no longer supported

If you have a critical requirement for these, please contact Aardvark.

## **New devices handled by 1.11.8.**

- This release fully supports the Innovative NV200 note acceptor. (older releases have subtle problems)

## **Minor enhancements in 1.11.8.**

- Older versions would cease reading out the cctalk coin / note information as soon a coin number was not recognised by the acceptor. This release examines all possible slots for coin / note information.
- The F56 handler supports a bill with a value of 2, against a "magnet pattern" of 0001.
- MDB Handlers now return string information (see the user manual.)

## **Bugs in version 1.11.7 & 1.11.6 fixed in this release.**

- There was a fault in the major error handling for the cctalk bus, which has been fixed.
- A fault in the handling of inhibits caused by missing ID003 note descriptions has been fixed.
- The handling of Merkur cashbox full / replaced has been redesigned, especially when a roll is being dumped.

## **New devices handled by 1.11.8 & 1.11.7.**

- This release supports the Merkur 100 note recycler, as a note acceptor and 3 dispenser.
- This release supports the Fujitsu F56 / F53 note dispenser as a number of dispenser devices, one per cassette.
- This release now fully supports the Azkoyen Hopper U series, including in encrypted mode.
- This release will automatically use the new level 3 encryption as used in the new Serial Compact Hopper 3E

### ***Minor enhancements in 1.11.8 & 1.11.7.***

- Coin table identity information is now retrieved from an SR5.
- An UNRECOGNISED\_COIN event is now queued when an acceptor reports an undefined coin.
- With this version an auto-route setting of Routed Path = 8 and Default Path = <other> now works with an SR5 - see note below.
- If a dispenser is inhibited whilst paying out, Paylink will now correctly handle the in-progress payout. This goes some way to allowing a payout to be abandoned by inhibiting all dispensers.
- Note acceptors can now be standard cctalk as well as BNV encrypted with CRC or unencrypted with CRC.

### ***Bugs in version 1.11.6 fixed in this release.***

- For acceptors that hold routing in RAM the coin routing would be lost on a power cycle. All routing is now restored following an acceptor reset.
- Acceptors that were reset before accepting any coins are now detected and re-set up.
- Some MCLoppers have the coin value held in bytes 0-5 of device Eeprom. Older release checked byte 1-8, but the SCH3 has the encryption level stored here. This resulted in hoppers erroneously acquiring the value 2 or 3.
- Many bugs fixed in the handling of Azkoyen hoppers. Especially bugs related to unencrypted hoppers with old style 4 byte dispense commands were wrongly treated following a NAK. This problem was introduced in 1.11.6 with the new hopper encryption code.
- Hoppers which do not respond with a NAK to 4 byte dispense are now handled. This particularly applies to Azkoyen hoppers in encrypted mode.
- If a hopper value is reassigned and the hopper is inhibited at almost the same time, then the new value assignment is lost from the Paylink without comment. The PC shows the value as changed, but the status DISPENSER\_VALUE\_REASSIGNED is never returned and the actual value is unchanged.
- The SEC meter update code will now allow unlimited increments - previous values only coped with increments < 32K
- The cctalk coin acceptor code now copes with acceptors whose path enables default to zero (off).

### ***Compatibility with 1.11.x / 1.10.x***

All software in this release can be freely mixed with that in 1.11.x and 1.10.x, except that the Aesimhei.h version shown above is required to obtain the new constants etc. (see below.)

### ***Upgrade / Downgrades***

Any earlier version of the firmware can be upgraded to this version without any problems. Downgrading to 1.11.x, 1.10.x or 1.9.x will not cause any problems.

## ***SR5 and Dynamic / Auto routing***

An SR5 stores internally both the “Routed Path” and the “Default Path”. These stored paths are retrieved by Paylink during start-up. The SR5 switches between these paths by using the concept of overriding the paths, an overridden Routed path is not used, and so the coin route falls back to the Default path.

As part of its design, an SR5 cannot have path 8 overridden. In previous versions this means that an auto-route setting of Routed Path = 8 and Default Path = <other> just does not work.

With this version such a setting now causes Paylink to turn off the “override capable” flag and update the coin routing dynamically - both for the coin involving path 8 and all other coins. Under these conditions, the path settings will no longer be saved by the SR5.

## ***Program Facilities changed.***

With the introduction of the new devices, some new statuses and constants have been introduced.

### **Merkur 100 note recycler**

The payout sequence for a Merkur recycler is slow, and requires the notes to be manually removed as they are delivered.

Because of this, Paylink supports the facility to dump a roll into the cash box. To dump a roll:

- the roll to be dumped should be found using ReadDispenserDetails(),
- the Dispenser.Status field changed to DISPENSER\_CASHBOX\_DUMP
- and WriteDispenserDetails() used to update the record to Paylink.

The dump will be complete when the CoinCount field for the roll is returned by ReadDispenserDetails() as zero.

There are also two new device identity constants:

DP_MERKUR_100	For the acceptor itself
DP_MERKUR_100_PAY	For each one of the recycler rolls.

### **Fujitsu F56 / F53 note dispenser**

The only change for this is the new device identity constants assigned to each of the dispenser’s cassettes:

DP_FUJITSU_F56
DP_FUJITSU_F53