

Milan / Paylink Firmware Version 4.1.10.15 Release Notice.

This is a **Full** (4) release of the Milan / Paylink Interface firmware - code version **1.10.15**. This is a maintenance fix that adds a few minor updates and clears the problems discovered since the release of version 1.10.9.

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

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 ships with fully signed drivers, suitable for XP, Vista and Windows 7. These drivers will install without any messages if the Internet is available when the Paylink module is plugged in. The same drivers are also automatically available via Windows update.
- 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- The old deprecated usbfs driver system has been replaced by the new libftdi system, which is available in source, and hence can be re-compiled for any Linux system, regardless of the hardware on which it is run.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.

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.

Bug in versions 1.10.10 / .11 / .12 / .13 / 1.10.14 fixed in this release.

- Bug fixed whereby hardware problems with the communications could cause the Paylink unit to repeatedly crash. (The unit would revert to normal operation when the hardware problem was removed.)
In particular the “Data Line Frozen” diagnostic was not output by version 1.10.13 & 1.10.14.

Minor Firmware enhancements since 1.10.9.

- Azkoyen hoppers have significant differences from Money Controls hoppers. This release will comprehensively allow for these differences. The processing is triggered by a Manufacturers Id starting “Azk” - other replies will cause processing to default to the MoneyControls style.
- This release will automatically use the new Type 3 MCL encryption as used in the new Serial Compact Hopper 3E
- This release will automatically use the new Type 2 MCL encryption as used in the new Serial Compact Hopper 3.
- Serial Compact Hopper 3 is reported with an LS byte of 4, value DP_MCL_SCH3 in the above Aesimhei.h.
- Combi Hopper (SCH3A) is still reported with an LS byte of 3, but this value is now called DP_MCL_SCH3A in the above Aesimhei.h.
- This release will assign identify DP_MCL_SR3 to the new SR3I from MCL. (In fact any cctalk device string starting “SR3” - the old SR3 identified itself as “NXS”)
- Paylink will now detect and use note acceptors using no encryption plus checksum on cctalk (in addition to the current encryption plus CRC and no encryption plus CRC)
- This release correctly handles escrow hold on the latest Innovative (NV200) note acceptors. Older Innovative NV series acceptors did not handle the Escrow Hold command and so previous versions of Paylink never send the hold command to *any* Innovative acceptor.
- With this version an auto-route setting of Routed Path = 8 and Default Path = <other> now works with an SR5 - see note below.

Bugs in versions 1.10.10 / .11 / .12 / 1.10.13 fixed in this release.

- Bug fixed whereby hoppers with old style 4 byte dispense commands were wrongly treated following a NAK. This was especially problematic as some Azkoyen hoppers returned a NAK if a payout was requested when empty. This was introduced in 1.10.10 with the new hopper encryption code.

Bugs in versions 1.10.9 / .10 / .11 / .12 / 1.10.13 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 MCL hoppers have the coin value held in bytes 0-5 of device Eprom. 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.
- Bug fixed whereby if a cctalk hopper using the old 4 byte dispense command starts up in an abnormal state, then Paylink defaults to using a 7 byte message.
- If a dispenser is inhibited whilst paying out, Paylink will now correctly report the in-progress pay out. This goes some way to allowing a payout to be abandoned by inhibiting all dispensers.

Bugs in versions 1.10.9 / .10 / .11 / 1.10.12 fixed in this release.

- 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).

Bugs in versions 1.10.9 / .10 / 1.10.11 fixed in this release.

- If a hopper fails to respond to two successive status polls immediately following a payout then Paylink will pay twice as many coins as requested. The overpayment is actually reported as a fault and the system reports the number of coins actually paid.

Bugs in versions 1.10.9 / 1.10.10 fixed in this release.

- A significant fault was discovered and fixed in the USB firmware reprogramming code.
- A fault in the initialisation code was discovered that could cause very short pulse on the outputs following a reset on power up.

Bugs in version 1.10.9 fixed in this release.

- A fault in the cctalk note acceptor handler meant that all hoppers and coin acceptors would be disabled while the note acceptor was reporting a jam.
- A fault in the cctalk SR5 handler meant that if the main (hopper) path for a coin was changed, while that coin was overridden to the secondary (cashbox) path, future coins would route to the new main (hopper) path rather than to the secondary (cashbox) path.
- A fault in the initialisation code was discovered that could cause very short pulse on the outputs following a reset on power up.

Significant Side effects from version 1.10.9

- Older versions of Aesimhei.h equate a Combi Hopper ID, with LS byte **3**, as DP_MCL_SCH3 - i.e. this name is **not** the new Serial Compact Hopper 3.

Compatibility with 1.10.9 / 1.10.10 / 1.10.11 / 1.10.12 / 1.0.13

All software in this release can be freely mixed with that in 1.10.9 / 1.10.10 / 1.10.11 / 1.10.12 / 1.10.13

Upgrade / Downgrades

Any earlier version of the firmware can be upgraded to this version without any problems. Downgrading to 1.10.4 / 6 / 7 / 9 / 10 / 11 / 12 / 13 / 14 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.