

Milan / Paylink Firmware Version 4.1.10.14 Release Notice.

This is a **Full** (4) release of the Milan / Paylink Interface firmware - code version **1.10.14**. This is a maintenance fix that adds a few minor updates and clears the problems discovered since the release of version 1.10.9.

PC code versions.

To obtain all of the new facilities described in this release document the following PC versions are required:

Aesimhei.dll	Version 1.4.0.0 or later
AESWDriver.exe	Version 1.1.3.2 or later
AESW98Driver.exe	Version 1.1.2.2 or later
Aesimhei.h	Dated 19/09/08 or later (See side effects below)

Minor enhancements since 1.10.9.

- Azkoyen hoppers have significant differences from Money Controls hoppers. This release is the first Paylink release to 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")
- Paylnik 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 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.
- 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 payout. 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. For more details, see the discussion below.
- 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 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.

Reprogramming Problems.

All versions of Paylink prior to release 4.1.10.11 have a fault in the flash reprogramming code.

The Fault

During reprogramming of the Paylink unit from the USB link, a single bit can be programmed with the wrong data. The fault occurs in about 1% of re-programming operations and is not detected during the flash reprogramming operation.

Symptoms of faulty reprogramming.

As the fault is in the reprogramming code, the error symptom is apparent in the newly loaded code, not the previously resident code with the fault.

Paylink without Kernel (4.1.10.8 and earlier)

The corruption is detected by the Paylink unit and reported by the programmer and AESWDriver programs as an error. As most of the code in Paylink is never actually run, the chances of a corruption causing a problem are very low. However, any unexpected operation may be due to a corruption and some errors in operation may not be obvious.

Paylink with Kernel (4.1.10.9 and later)

The kernel detects the corruption when the application is checksummed before being started and the application is *not* started and the unit goes *directly* into re-programming mode. In this situation, the AES driver program does not show the connection as operational and the windows re-programming utility shows the existing application version as 0.0.0.0.

Note that this symptom is not expected to be seen with releases 4.1.10.11 and later. (see below.)

Paylinks Affected

Any Paylinks that has been reprogrammed via the USB port, where the previous loaded code was 4.1.10.10 or earlier are at risk of corrupted flash locations.

The only way to guarantee that a Paylink is running uncorrupted code is to check the Driver / Programmer, or to load version 4.1.10.9 or later.

Updating problem Paylinks with 4.1.10.11 or later.

When the new version (4.1.10.11 or later) is downloaded to Paylink, the flash is reprogrammed with the flash programmer that is part of the previously loaded code. This means that re-programming a unit that was running old software (pre 4.1.10.11) still has a 1% chance of corrupting the flash.

If a corruption does occur when a unit is reprogrammed with 4.1.10.11 or later, the new Kernel detects it and the Paylink reprogramming mechanism will run again. This time though the new flash reprogramming algorithm is used and will be successful.

The loader program (Windows version 1.0.4.4) has been enhanced to detect a Paylink that has automatically restarted in this way and will automatically repeat the download that has failed. As this repetition will use the newly downloaded correct code, the reprogramming will work this time and the Paylink will remain operational.