Impinj Application Note

Loss Prevention using Monza® R6-P

Rev 1.0 April 06, 2015

1 Introduction

When using a RAIN RFID tag for Loss Prevention or Electronic Article Surveillance (EAS) we are required to solve several core challenges:

- **Quickly discriminate sold from not-sold items:** An exit gate should be able to decide, within a few milliseconds, whether any particular tagged item is sold or not-sold.
- **Not alarm on foreign tags:** The solution should ignore tags from other retailers and from other applications, yet reliably alarm on not-sold store tags.
- **Allow seamless product returns:** The solution should allow a consumer to seamlessly return an item, and a retailer to quickly verify the sold status of the item before accepting the return.
- **Protect consumer privacy:** The solution should (1) minimize false alarms and (2) allow seamless product returns without storing consumer information on the tag.
- **Provide Gen2 compatibility:** The solution should not interfere with other uses of the tag, such as for inventory visibility.

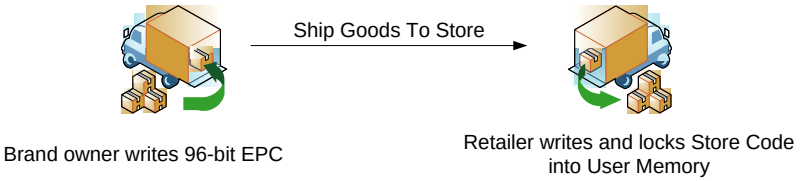
The three solutions described in this note have been designed to address different retail environments. All three solutions can be deployed using Monza R6-P (MR6-P) with the Max_User memory profile and they take into account the differing capabilities of point-of-sale and point-of-exit equipment.

All three solutions are simple and they share a common starting point with the concept of a **Store Code**.

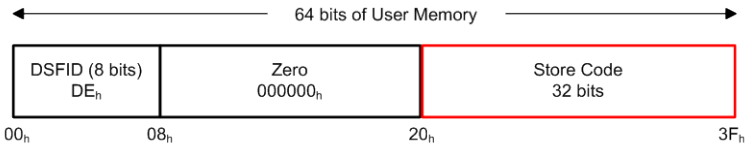
2 The Store Code

A **Store Code** links each tagged item to the store selling it. A brand owner, service bureau, or the retailer itself writes a Store Code into each item's tag. In most cases a reader can bulk-encode all tags in a box either on the outbound side of a distribution center or at store receiving, regardless of the SKU content.

- The Store Code can be unique to a chain (i.e. all ACME stores use a single Store Code that is different from that of all other retailers) or it can be unique to each store (i.e. each ACME store uses a unique Store Code that is different from that of all other stores). A 32-bit Store Code allows four billion stores and the Store Code should be permanently locked in tag memory.
- There are several approaches to assigning a Store Code. A retailer can encode a Private Enterprise Number (PEN) as the 32-bit Store Code. If a retailer does not have a PEN, they are readily available from the Internet Assigned Numbers Authority (IANA) at http://pen.iana.org/pen/app.



For the Monza R6-P with Max_User memory profile we recommend that the Store Code be written to the second word of user memory, *i.e.* locations 20_h–$3F_h$. A DSFID is required[1] as the first byte of user memory, set to DE_h, and the remaining 24 bits in the first 32-bit block of user memory can be set to zero. User memory should then be configured so that the first 32-bit block is perma-unlocked and the second 32-bit block is perma-locked. An exit-gate reader can *Select* on the Store Code or read the Store Code using a *Read* command.



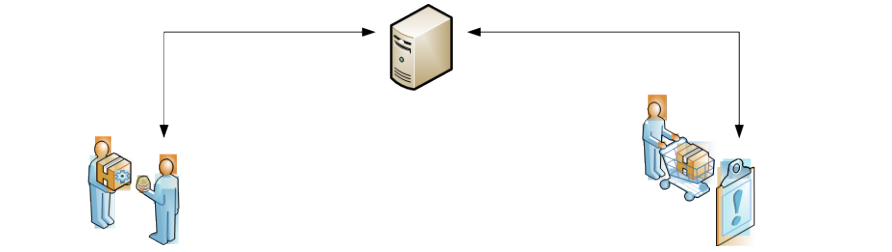
3 Three EAS Solutions using Monza R6-P

The solutions in this application note are designed to account for the different ways the point-of-sale and point-of-exit might be configured.

3.1 A Networked Solution

In some retail environments the point-of-sale and the point-of-exit are fully networked. By this we mean that events at the point-of-sale result in real-time changes to a back-end database. Equipment at the point-of-exit would then have immediate access to any changes made to the same back-end database. For those concerned about network latencies, a variety of optimizations can be considered including a push-model whereby information on sold items is proactively pushed to points-of-exit. Whatever the details of an implementation, the net result is that legitimate sales can be immediately recognized.

- **Point-of-sale:** At the point-of-sale the retailer marks an item as sold in an online database.
- **Store exit:** The exit reader broadcasts a *Select* command to all tags, instructing only those with the store's Store Code to respond. All foreign tags remain silent. Readers inventory the departing store tags, verifying with the online database that items are sold. Readers sound an alarm on not-sold tags leaving the store.



- **Returns desk:** The returns desk verifies an item's Store Code and consults the online database to see the status of the item. After the return is processed the item's status in the online database is returned to *not sold* and the item can be returned to the shelf.

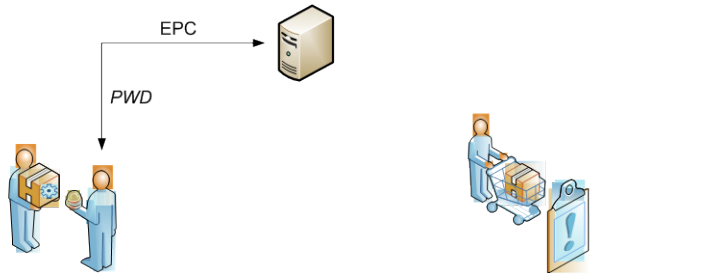
[1] The Data Storage Format Identifier (DSFID) specifies the data format for the user memory bank.

3.2 Using the Short-Range Privacy Feature on MR6-P

In some retail environments the point-of-sale and point-of-exit cannot easily be networked together. Or perhaps the online database cannot be consulted quickly enough for a networked EAS solution to be viable. In this situation Impinj recommends that the following solution be considered with MR6-P.

Each tag should be allocated an *access password* as specified in the Gen2 specifications. This is a process that can be managed at a store, or passwords can be provisioned earlier in the tag life-cycle and provided to the store.

- **Point-of-sale:** At point-of-sale the retailer writes a **Sold Code** to the tag and updates an online database that an item has been sold. A Sold Code can be generic or unique. For example, the Sold Code can be a generic code-of-the-day or it can be a derived from the tag's EPC. Methods to generate a Sold Code are discussed in Section 3.4. The retailer then uses the per-tag *access password* to change MR6-P from long-range to short-range mode.
- **Store exit:** The exit reader broadcasts a *Select* command to all tags, instructing only those with the store's Store Code to respond. All foreign tags remain silent. All store-owned tags that have been sold should remain silent[2]. Readers inventory the store tags that respond and check the Sold Code. Unless a legitimate Sold Code is stored in the tag the alarm is triggered.



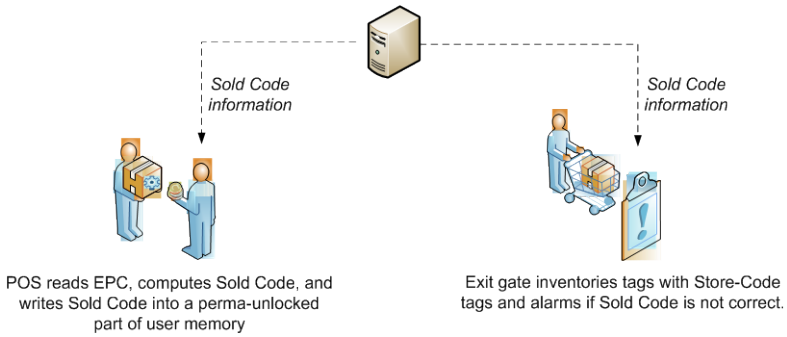
- **Returns desk:** The returns desk checks whether an item has been sold either by verifying a Sold Code on the tag or consulting an online database. After the return has been processed, the reader erases the Sold Code and/or changes the item's status in the online database. The reader changes MR6-P from short- to long-range and the item is returned to the shelves.

3.3 An Off-Line Solution

There may well be circumstances where the retailer does not wish to manage per-tag access passwords and where the point-of-sale and point-of-exit cannot be networked together. In this situation the following off-line solution may be used. A pre-requisite is that points of sale and exit are all provisioned with the means of verifying a Sold Code. There are many different ways of implementing a Sold Code and these offer different levels of security and convenience.

[2] Depending on the environment of use and the positioning of equipment at the point-of-exit, there might remain a small probability of detecting a sold tag in short-range mode. For this reason Impinj recommends a Sold Code be supported as part of the solution.

- **Point-of-sale:** At point-of-sale the retailer writes a Sold Code into each item. A Sold Code can be generic or unique. For example, the Sold Code can be a generic code-of-the-day or it can be a derived from the tag's EPC. Methods to generate a Sold Code are discussed in Section 3.4.
- **Store exit:** The exit reader broadcasts a Select command to all tags, instructing only those with the store's Store Code to respond. All foreign tags remain silent. Readers inventory the departing store tags, verify the Sold Codes, and alarm on not-sold tags leaving the store.

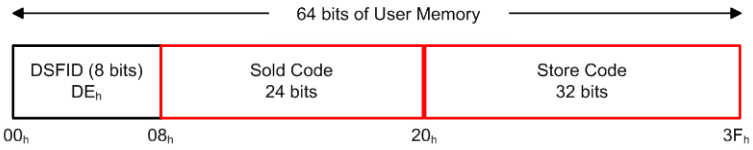


- **Returns desk:** The returns desk verifies an item's Store Code and Sold Code. After the return is processed the reader erases the Sold Code from its permanently unlocked region of memory, thereby returning the tag to its pre-sold state.

3.4 Generating a Sold Code

The Sold Code can be a generic code of the day distributed to all points of sale and exit. Alternatively it can be derived from the tag's EPC. Depending on which solution is deployed, some combination of the point-of-sale, the returns desk, and the point-of-exit would need the ability to use or verify a Sold Code.

Some stores may prefer to use cryptography to generate a Sold Code and equipment can be provisioned with a software-based AES-128 algorithm (http://en.wikipedia.org/wiki/AES_implementations) and a 128-bit key. To generate a Sold Code the AES algorithm and key might be used to encrypt the EPC. The 24 least significant bits of the output from the AES encryption constitute the Sold Code. However Sold Codes can also be generated without using cryptography and each retailer should choose a sold-code model appropriate for their needs. The Sold Code should be written into the first word of user memory, *i.e.* at location 08_h–$1F_h$.



4 Summary

The Loss Prevention solutions in this note are simple and meet the goals often specified in retailer requirements:

- ✓ **They quickly discriminate sold from not-sold items**
- ✓ **They allow seamless product returns**
- ✓ **They do not alarm on foreign tags**
- ✓ **They protect consumer privacy**
- ✓ **They are Gen2 compatible**

While the Loss Prevention solutions in this application note can be used with most of Impinj's Monza chip family, they are ideally suited to implementation using Impinj's retail-optimized Monza R6-P.

Notices

Impinj gives no representation or warranty, express or implied, for accuracy or reliability of information in this document. Impinj reserves the right to change its products and services and this information at any time without notice.

EXCEPT AS PROVIDED IN IMPINJ'S TERMS AND CONDITIONS OF SALE (OR AS OTHERWISE AGREED IN A VALID WRITTEN INDIVIDUAL AGREEMENT WITH IMPINJ), IMPINJ ASSUMES NO LIABILITY WHATSOEVER AND IMPINJ DISCLAIMS ANY EXPRESS OR IMPLIED WARRANTY, RELATED TO SALE AND/OR USE OF IMPINJ PRODUCTS INCLUDING LIABILITY OR WARRANTIES RELATING TO FITNESS FOR A PARTICULAR PURPOSE, MERCHANTABILITY, OR INFRINGEMENT.

NO LICENSE, EXPRESS OR IMPLIED, BY ESTOPPEL OR OTHERWISE, TO ANY PATENT, COPYRIGHT, MASK WORK RIGHT, OR OTHER INTELLECTUAL PROPERTY RIGHT IS GRANTED BY THIS DOCUMENT.

Impinj assumes no liability for applications assistance or customer product design. Customers should provide adequate design and operating safeguards to minimize risks.

Impinj products are not designed, warranted or authorized for use in any product or application where a malfunction may reasonably be expected to cause personal injury or death or property or environmental damage ("hazardous uses") or for use in automotive environments. Customers must indemnify Impinj against any damages arising out of the use of Impinj products in any hazardous or automotive uses.

Impinj, Monza, AutoTune, TagFocus, FastID, Enduro, Integra, ItemEncode, ItemSense, xArray, xPortal and Monza Self-Serialization are trademarks of Impinj, Inc. All other product or service names are trademarks of their respective companies.

These products may be covered by one or more U.S. patents. See www.impinj.com/patents for details.