



---

### The AES 125kHz Transponder with Open Immobilizer Software Stack

---

#### PRELIMINARY DATASHEET

---

#### Features

- AES crypto transponder in plastic brick package
  - Includes coil and capacitor for tuned circuit antenna
- Radio Frequency  $f_{RF} = 125\text{kHz}$
- Contactless power supply
- Contactless bidirectional data communication interface
- High performance AES encryption hardware unit
- Open immobilizer stack by Atmel®
- 2K EEPROM for secret key storage, field user data, and configuration data
- Error correction code support for NVM
- 32-bit unique ID
- Multiple configuration registers
- Modulation/coding: Biphase, Manchester, QPLM
- Configurable baudrate
- $-40^{\circ}\text{C}$  to  $+85^{\circ}\text{C}$  operation Temperature
- LGA like brick package

## 1. Description

The Atmel® ATA5580 is a smart transponder module with an AES encryption unit, customer EEPROM, an 125kHz LF front end and a LF ferrite antenna for wireless power supply and communication. All components are built up in a single pinless transponder package. The IC contains the highly configurable Open Immobilizer Software Stack by Atmel.

### 1.1 Module Schematic

The Atmel ATA5580 transponder contains an ultra low power Transponder IC with AES engine, an LF Antenna resonant circuit, and a buffer capacitor.

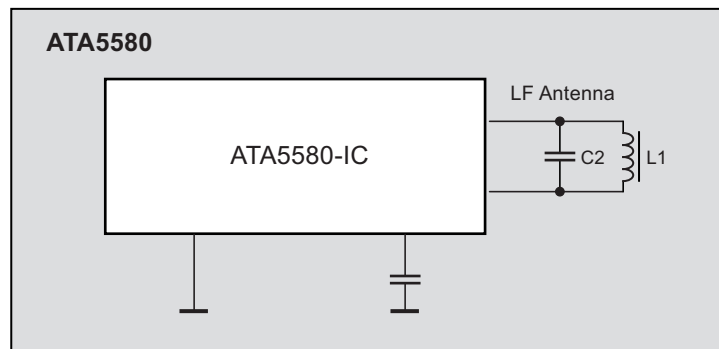
### 1.2 Functional Description

Atmel® ATA5580 is designed for automotive immobilization applications in RKE keys. The Atmel ATA5580 micro module consists of an ultra low power IC with AES encryption engine and Immobilizer frontend, an LF ferrite antenna and capacitors for the antenna and as supply buffer.

The small LGA -like package of the Atmel ATA5580 contains all the components required for the transponder application.

The IC needs no battery supply as it is powered by a 125kHz LF field. The communication with the chip is also done via an LF Field. A base station can request data via an LF telegram and the transponder responds with data from its memory or with cipher data via a damping modulation from the LF field. The transponder function is defined by a special Atmel immobilizer stack.

**Figure 1-1. Block Diagram**



## 2. Atmel Open Immobilizer Protocol Description

### 2.1 Overview

#### 2.1.1 Protocol Flexibility

The Atmel® Immobilizer protocol has been designed as a configurable software stack.

For instance, security levels, turn-around authentication time, and authentication schemes are all configurable at runtime while covering a wide range of car manufacturer requirements.

Additionally, Atmel defined three default configurations respectively targeting fast, standard, and high security for which analysis of bit security strength vs. turn-around time was carried out. Obviously, flexibility for tuning the Protocol stack to meet specific constraints is still a feature.

#### 2.1.2 Open Software Stack

Rather than developing its own proprietary cryptographic functions, Atmel selected and implemented the 128-bit AES global benchmark standard as its data encryption and decryption source. This standard is open source and freely available to the public for use and scrutiny. Because of this it continues to be favored by industry experts over private and proprietary crypto algorithms.

In addition to selecting an open source and public AES crypto function, the firmware includes user configurable options that enable the engineer to “build” an authentication protocol that meets user requirements. The complete documentation of the protocol configuration options are made publicly available. The encryption and configuration of the authentication protocol are open source and freely available to customers free of charge.

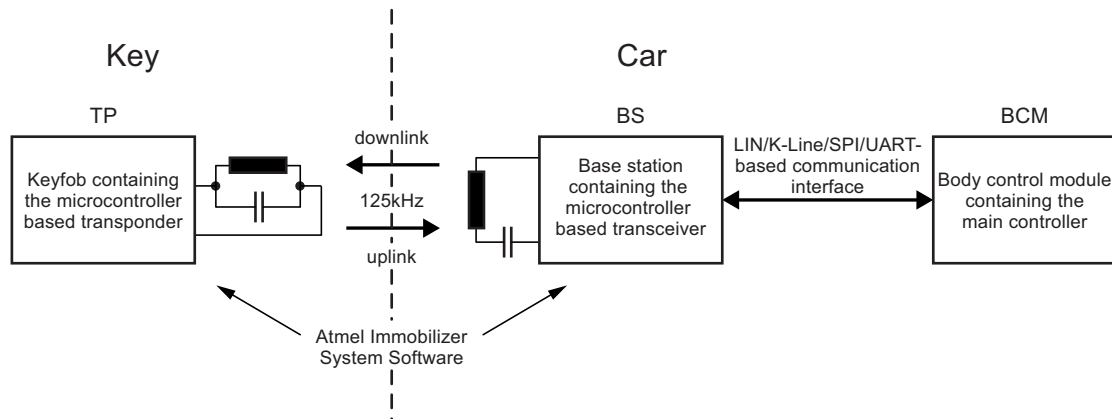
#### 2.1.3 Production-ready software implementation

Besides defining an Open Immobilizer Protocol stack, Atmel elected to implement it in all Car Access devices with an embedded LF frontend. This implementation complies with Automotive grade development standards (CMMI - Automotive Spice) and is production-ready.

### 2.2 System overview

As a sub-system of the general car access system, the immobilizer is not used for accessing the car but instead to allow the driver to start the engine. [Figure 2-1](#) visualizes system partitioning.

**Figure 2-1. System Overview**



## 2.3 Device Support

The firmware implementation developed by Atmel® uses specific hardware blocks that are found in our vehicle access product line. The transponder features are optimized to function seamlessly with the following devices:

- Atmel ATA5580 (TrAESure): Standalone transponder
- Atmel ATA5790 (Primus): Passive Entry/Go Microcontroller with 3D LF Receiver and Transponder interface
- Atmel ATA5794 (PrimeOne): Remote Keyless Entry Microcontroller with Transponder interface
- Atmel ATA5795 (Primatik): Remote Keyless Entry Microcontroller with Transponder interface and Frac-N RF Transmitter.

The vehicle consists of the Atmel ATA5272 base station device containing a matched firmware library for implementing the entire system.

The transponder's hardware and software layers have been specifically designed to be compatible with any FDX basestation available on the market by implementing the protocol described in this document on the host microcontroller.

## 2.4 Firmware Features

The purpose of this section is to provide an overview of the complete immobilizer features included with the Atmel firmware library. It also describes the information flow between the car side base station and the key side transponder. It includes definitions and requirements in terms of physical layer, protocol layer, and encryption.

## 2.5 Memory Partitioning

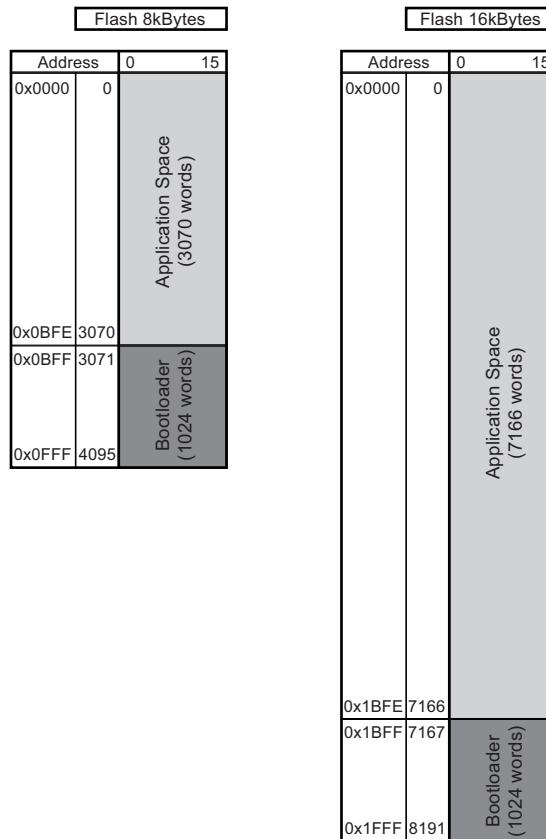
Except for the Atmel ATA5580, There are two types of memory on the Atmel devices are used by both the Immobilizer and the application. These memories need to be partitioned and some guidelines established to ensure reliable operation. Program code stored in Flash memory is typically used as read -only memory once initial programming has occurred. Non -volatile memory that supports multiple read/write access is provided through EEPROM memory structures.

### 2.5.1 Flash Memory

The Immobilizer firmware developed by Atmel is stored in the bootloader section of the Flash memory. This is shipped from Atmel with the bootloader section protected from overwriting through the use of fuse settings. This allows the application space to be programmed without corrupting the Immobilizer firmware.

Each Atmel device provides differing amounts of flash memory. The bootloader space is consistent across devices at 2kbytes. In the case of the Atmel ATA5580 all of the Flash memory (8K) is available for the Immobilizer Stack. [Figure 2-2 on page 5](#) shows how the flash memory is partitioned for various memory sizes.

**Figure 2-2. The Flash Memory Partition**



### 2.5.2 Non-volatile Memory

Non-volatile memory used for data storage is implemented in EEPROM structures. This is subdivided into two pages.

Page one provides read and write access for storage of application and immobilizer data. This includes four special access protection (AP0 - AP3) areas. The protection takes the form of requiring an intentional setting of the second register before programming is possible. The AP0 location has been selected for exclusive use by the Atmel® Immobilizer firmware. The application code should be audited to insure that this memory is not used to prevent corruption. [Figure 2-3 on page 6](#) shows the use of EEPROM Page 1.

**Figure 2-3. EEPROM Page 1**

| EEPROM 2kBytes |      |                                   |
|----------------|------|-----------------------------------|
| Address        |      | 0 7                               |
| 0x0000         | 0    | Application Space<br>(1920 Bytes) |
|                |      |                                   |
|                |      |                                   |
|                |      |                                   |
|                |      |                                   |
|                |      |                                   |
|                |      |                                   |
|                |      |                                   |
|                |      |                                   |
|                |      |                                   |
|                |      |                                   |
|                |      |                                   |
|                |      |                                   |
|                |      |                                   |
|                |      |                                   |
| 0x05FF         | 1535 |                                   |
| 0x0600         | 1536 |                                   |
| 0x067F         | 1663 | AP3                               |
| 0x0680         | 1664 | AP2                               |
| 0x06FF         | 1791 |                                   |
| 0x0700         | 1792 | AP1                               |
| 0x077F         | 1919 |                                   |
| 0x0780         | 1920 | AP0                               |
| 0x07FF         | 2047 |                                   |

Page 2 is locked from overwriting at the end of Atmel® manufacturing. This page contains a vast set of configuration and identification features. Once these have been set they are protected from any subsequent changes.

### 2.5.2.1 Secret Key Storage

Atmel makes provisions for a total of three secret keys that can be used. One of these is the fixed Default Secret Key which resides in the locked Page 2 of EEPROM and is intended for use during a secure key transfer process to establish the other two secret keys.

The other two secret keys are intended for use during normal operation. These are stored in the AP0 section of EEPROM when the supplied LF interface is used to pair the transponder to the vehicle. To ensure integrity, The LF interface for transferring secret keys also stores each of these two secret keys with two copies. When the secret key is accessed for the authentication process, all three copies are read out and checked against each other for errors. Any corruption of a single copy can be corrected automatically. [Figure 2-4 on page 7](#) shows the mapping of the AP0 section located in Page 1 of EEPROM.

The size of the Secret key is 16 bytes.

The Secret Keys for immobilizer and Application must be stored based on the configuration stored in Page 2.

Both Secret key 1 and Secret key 2 must be stored with 2 copies in their respective locations.

[Figure 2-4 on page 7](#) represents the allocation of Secret Key in the EEPROM Memory.

**Figure 2-4. The AP0 Memory Map**

| 128 bit                        |        |        |        |        |        |        |        |        |        |         |         |         |         |         |         |         | Physical Address | AP0 128 Bytes |
|--------------------------------|--------|--------|--------|--------|--------|--------|--------|--------|--------|---------|---------|---------|---------|---------|---------|---------|------------------|---------------|
| Secret Key                     | Data 1 | Data 2 | Data 3 | Data 4 | Data 5 | Data 6 | Data 7 | Data 8 | Data 9 | Data 10 | Data 11 | Data 12 | Data 13 | Data 14 | Data 15 | Data 16 |                  |               |
| 2                              |        |        |        |        |        |        |        |        |        |         |         |         |         |         |         |         | 0780 - 078F      |               |
| 2 (Copy 1)                     |        |        |        |        |        |        |        |        |        |         |         |         |         |         |         |         | 0790 - 079F      |               |
| 2 (Copy 2)                     |        |        |        |        |        |        |        |        |        |         |         |         |         |         |         |         | 07A0 - 07AF      |               |
|                                |        |        |        |        |        |        |        |        |        |         |         |         |         |         |         |         | 07B0 - 07BF      |               |
| 1                              |        |        |        |        |        |        |        |        |        |         |         |         |         |         |         |         | 07C0 - 07CF      |               |
| 1 (Copy 1)                     |        |        |        |        |        |        |        |        |        |         |         |         |         |         |         |         | 07D0 - 07DF      |               |
| 1 (Copy 2)                     |        |        |        |        |        |        |        |        |        |         |         |         |         |         |         |         | 07E0 - 07EF      |               |
|                                |        |        |        |        |        |        |        |        |        |         |         |         |         |         |         |         | 07F0 - 07FF      |               |
| 128 bytes of Secret Key memory |        |        |        |        |        |        |        |        |        |         |         |         |         |         |         |         |                  |               |

The unassigned locations of AP0 are reserved for general variable storage for the Immobilizer firmware.

## 2.5.2.2 Configuration Memory Options

The Atmel® firmware includes highly configurable immobilizer features allowing the system design to be optimized. All configuration options must be selected during design testing and validation and are placed and locked in Page 2 of EEPROM.

### Data Check Disable

EEPROM address 0x0815 bit 0 allows the CRC data to be disabled for both the Request Frame and the Response Frame.

Data Check Disable (DCD): 0 = CRC enabled, 1 = CRC disabled

This configuration bit is checked when sending or receiving all commands.

| Byte Address | Bit 7 | Bit 6 | Bit 5 | Bit 4 | Bit 3 | Bit 2 | Bit 1 | Bit 0 | Remarks       |
|--------------|-------|-------|-------|-------|-------|-------|-------|-------|---------------|
| 815          | TDH   | SKT   | KS    | DLP1  | DLP0  | CM    | MOD   | DCD   | Configuration |

### Authentication Format

EEPROM address 0x0815 bit 2 allows the type of authentication protocol to be selected.

Crypto Mode (CM): 0 = Unilateral, 1 = Bilateral

This configuration bit is checked when the Start Authentication and Memory Access commands are executed. Details of this interaction are provided in the LF Command Set section.

| Byte Address | Bit 7 | Bit 6 | Bit 5 | Bit 4 | Bit 3 | Bit 2 | Bit 1 | Bit 0 | Remarks       |
|--------------|-------|-------|-------|-------|-------|-------|-------|-------|---------------|
| 815          | TDH   | SKT   | KS    | DLP1  | DLP0  | CM    | MOD   | DCD   | Configuration |

### Challenge and Response Length

These two configuration registers deal with the number of bits transferred during authentication. The length of the challenge that the transponder expects is stored in EEPROM address 0x0819. In response the transponder returns an encrypted value with a length determined by the setting in address 0x081A. The start authentication command must have knowledge of these length settings to be used in the authentication protocol.

| Byte Address | Bit 7 | Bit 6 | Bit 5 | Bit 4 | Bit 3 | Bit 2 | Bit 1 | Bit 0 | Remarks          |
|--------------|-------|-------|-------|-------|-------|-------|-------|-------|------------------|
| 819          | CH7   | CH6   | CH5   | CH4   | CH3   | CH2   | CH1   | CH0   | Challenge length |
| 81A          | RS7   | RS6   | RS5   | RS4   | RS3   | RS2   | RS1   | RS0   | Response length  |

## Uplink Coding and Data Rate

EEPROM address 0x0815 bit 1 allows the uplink coding type to be selected.

Uplink Modulation (MOD): 0 = Manchester, 1 = BiPhase

The Baud Rate Setting (0x0817) sets the threshold for the Manchester/BiPhase encoder. This works in combination with the T2 Prescaler (0x0818) to provide a very accurate and flexible transmission of data from the transponder to the vehicle. A typical value is recommended as 0x07 and 0x00 respectively to provide approximately 3.906kB/s.

| Byte Address | Bit 7 | Bit 6 | Bit 5 | Bit 4 | Bit 3 | Bit 2 | Bit 1 | Bit 0 | Remarks           |
|--------------|-------|-------|-------|-------|-------|-------|-------|-------|-------------------|
| 815          | TDH   | SKT   | KS    | DLP1  | DLP0  | CM    | MOD   | DCD   | Configuration     |
| 816          | PLM7  | PLM6  | PLM5  | PLM4  | PLM3  | PLM2  | PLM1  | PLM0  | PLM threshold     |
| 817          | BD7   | BD6   | BD5   | BD4   | BD3   | BD2   | BD1   | BD0   | Baud rate setting |
| 818          | T23   | T22   | T21   | T20   |       |       | T2D1  | T2D0  | T2 prescaler      |

## Downlink Coding and Data Rate

EEPROM address 0x0815 bits 3 and 4 allows the downlink coding type to be selected.

Downlink Protocol (DLP1:0): 00 = BPLM, 01 = QPLM (one of four codings), 10 = DPS

PLM Threshold (0x0816) sets the threshold used to decode BPLM data from the vehicle. The value in this register (PLM0 - PLM7) is used to determine if the number of field clock cycles received represents a logical zero or one. For example, a typical BPLM configuration uses 16 field clocks to represent a zero and 32 field clocks to represent a one. The threshold setting can then be set to 24 to achieve accurate decoding.

In QPLM mode the PLM Threshold becomes the reference value that is used to determine the four possible state values.

| Byte Address | Bit 7 | Bit 6 | Bit 5 | Bit 4 | Bit 3 | Bit 2 | Bit 1 | Bit 0 | Remarks       |
|--------------|-------|-------|-------|-------|-------|-------|-------|-------|---------------|
| 815          | TDH   | SKT   | KS    | DLP1  | DLP0  | CM    | MOD   | DCD   | Configuration |
| 816          | PLM7  | PLM6  | PLM5  | PLM4  | PLM3  | PLM2  | PLM1  | PLM0  | PLM threshold |

## Secret Key Selection and Transfer

EEPROM address 0x0815 bits 5 and 6 configure the handling of secret keys in the system.

Key Select (KS): 0 = Secret Key One, 1 = Secret Key Two

Secure Key Transfer (SKT): 0 = Off, 1 = On

The secret key selected in this option determines which key from the AP0 section of EEPROM is used during the Start Authentication command. In addition, the type of key transfer process used to load the secret keys into AP0 is specified using this configuration.

| Byte Address | Bit 7 | Bit 6 | Bit 5 | Bit 4 | Bit 3 | Bit 2 | Bit 1 | Bit 0 | Remarks       |
|--------------|-------|-------|-------|-------|-------|-------|-------|-------|---------------|
| 815          | TDH   | SKT   | KS    | DLP1  | DLP0  | CM    | MOD   | DCD   | Configuration |



## Fob Power Up

EEPROM address 0x0815 bit 7 allows the detection header functionality to be selected.

Detection Header (TDH): 0 = Off, 1 = On

This configuration determines if the detection header is included as part of the immobilizer initialization routine.

| Byte Address | Bit 7 | Bit 6 | Bit 5 | Bit 4 | Bit 3 | Bit 2 | Bit 1 | Bit 0 | Remarks       |
|--------------|-------|-------|-------|-------|-------|-------|-------|-------|---------------|
| 815          | TDH   | SKT   | KS    | DLP1  | DLP0  | CM    | MOD   | DCD   | Configuration |

## Default Secret Key

A 128-bit default secret key is programmed and locked into EEPROM address locations 0x081B to 0x82A. This is programmed identically for all devices that are shipped to the customer and includes the Customer ID address (0x081B). The remaining 15 bytes of data can be specified by the customer or assigned by Atmel®. This default secret key cannot be read out of EEPROM by LF field commands. The default secret key is used for the secure key transfer process.

| Byte Address | Bit 7 | Bit 6 | Bit 5 | Bit 4 | Bit 3 | Bit 2 | Bit 1 | Bit 0 | Remarks            |
|--------------|-------|-------|-------|-------|-------|-------|-------|-------|--------------------|
| 81B          | CID7  | CID6  | CID5  | CID4  | CID3  | CID2  | CID1  | CID0  | Customer ID        |
| 81C          | SK119 | SK118 | SK117 | SK116 | SK115 | SK114 | SK113 | SK112 | Default secret key |
| 81D          | SK111 | SK110 | SK109 | SK108 | SK107 | SK106 | SK105 | SK104 |                    |
| 81E          | SK103 | SK102 | SK101 | SK100 | SK99  | SK98  | SK97  | SK96  |                    |
| 81F          | SK95  | SK94  | SK93  | SK92  | SK91  | SK90  | SK89  | SK88  |                    |
| 820          | SK87  | SK86  | SK85  | SK84  | SK83  | SK82  | SK81  | SK80  |                    |
| 821          | SK79  | SK78  | SK77  | SK76  | SK75  | SK74  | SK73  | SK72  |                    |
| 822          | SK71  | SK70  | SK69  | SK68  | SK67  | SK66  | SK65  | SK64  |                    |
| 823          | SK63  | SK62  | SK61  | SK60  | SK59  | SK58  | SK57  | SK56  |                    |
| 824          | SK55  | SK54  | SK53  | SK52  | SK51  | SK50  | SK49  | SK48  |                    |
| 825          | SK47  | SK46  | SK45  | SK44  | SK43  | SK42  | SK41  | SK40  |                    |
| 826          | SK39  | SK38  | SK37  | SK36  | SK35  | SK34  | SK33  | SK32  |                    |
| 827          | SK31  | SK30  | SK29  | SK28  | SK27  | SK26  | SK25  | SK24  |                    |
| 828          | SK23  | SK22  | SK21  | SK20  | SK19  | SK18  | SK17  | SK16  |                    |
| 829          | SK15  | SK14  | SK13  | SK12  | SK11  | SK10  | SK9   | SK8   |                    |
| 82A          | SK7   | SK6   | SK5   | SK4   | SK3   | SK2   | SK1   | SK0   |                    |

### 2.5.2.3 Fixed Identification

Fixed identification contains data that has been programmed and locked by Atmel. This data is provided for use in the immobilizer application as well as part of supply chain management.

#### Unique ID

The ID or serial number consists of 32 bits of non-sequential, unique values. Each transponder is assigned this value at the end of the manufacturing process. The value is stored at EEPROM address locations 0x0800 to 0x0803. This value can be accessed very efficiently through the use of the Read UID command.

The Customer ID stored at address 0x0804 may optionally be added to the Unique ID.

| Byte Address | Bit 7 | Bit 6 | Bit 5 | Bit 4 | Bit 3 | Bit 2 | Bit 1 | Bit 0 | Remarks              |
|--------------|-------|-------|-------|-------|-------|-------|-------|-------|----------------------|
| 800          | ID31  | ID30  | ID29  | ID28  | ID27  | ID26  | ID25  | ID24  | Unique ID / Serial # |
| 801          | ID23  | ID22  | ID21  | ID20  | ID19  | ID18  | ID17  | ID16  |                      |
| 802          | ID15  | ID14  | ID13  | ID12  | ID11  | ID10  | ID9   | ID8   |                      |
| 803          | ID7   | ID6   | ID5   | ID4   | ID3   | ID2   | ID1   | ID0   |                      |
| 804          | CID7  | CID6  | CID5  | CID4  | CID3  | CID2  | CID1  | CID0  | Customer ID          |

#### Atmel Traceability

Atmel traceability entails information that can be used to determine where and how this device has been processed. The following information completely identifies this device in the Atmel process chain:

|                  |                |
|------------------|----------------|
| Address          | - Value        |
| 0x0808           | - Device Type  |
| 0x0809 to 0x080B | - Lot Number   |
| 0x080C           | - Wafer Number |
| 0x080D to 0x080E | - Die Number   |

| Byte Address | Bit 7 | Bit 6 | Bit 5 | Bit 4 | Bit 3 | Bit 2 | Bit 1 | Bit 0 | Remarks      |
|--------------|-------|-------|-------|-------|-------|-------|-------|-------|--------------|
| 808          | DEV7  | DEV6  | DEV5  | DEV4  | DEV3  | DEV2  | DEV1  | DEV0  | Device type  |
| 809          | LOT23 | LOT22 | LOT21 | LOT20 | LOT19 | LOT18 | LOT17 | LOT16 | LOT number   |
| 80A          | LOT15 | LOT14 | LOT13 | LOT12 | LOT11 | LOT10 | LOT9  | LOT8  |              |
| 80B          | LOT7  | LOT6  | LOT5  | LOT4  | LOT3  | LOT2  | LOT1  | LOT0  |              |
| 80C          | WAF7  | WAF6  | WAF5  | WAF4  | WAF3  | WAF2  | WAF1  | WAF0  | Wafer number |
| 80D          | DIE15 | DIE14 | DIE13 | DIE12 | DIE11 | DIE10 | DIE9  | DIE8  | Die number   |
| 80E          | DIE7  | DIE6  | DIE5  | DIE4  | DIE3  | DIE2  | DIE1  | DIE0  |              |

#### Software Revision

The software revision is contained in EEPROM address 0x080F and provides information about the current version loaded into flash memory.

| Byte Address | Bit 7 | Bit 6 | Bit 5 | Bit 4 | Bit 3 | Bit 2 | Bit 1 | Bit 0 | Remarks     |
|--------------|-------|-------|-------|-------|-------|-------|-------|-------|-------------|
| 80F          | SW7   | SW6   | SW5   | SW4   | SW3   | SW2   | SW1   | SW0   | SW revision |

## 2.6 Device Initialization

This section describes how the transponder device handles the initial power up sequence. The outcome or determination from the initialization sequence depends on various conditional paths. These are described in the following sections. The system can guarantee that the immobilizer functionality is given the highest preference and can operate independently from the application code by means of this initialization sequence.

### 2.6.1 Power-up Scenarios

Power up occurs whenever there is a reset event. This can be power-on-reset (POR), external reset, watchdog reset, brown-out reset, and transponder reset. This reset sets all registers, ports, and SRAM to initial conditions. The program counter is always set to the reset vector located in the bootloader section. This ensures the priority of the immobilizer over all other functions. After a fixed delay, a code is executed to check the conditions described as follows.

### 2.6.2 LF Field Detection

The very first item checked after the reset delay is the determination of the presence of an LF field. If the LF field is present then the immobilizer function is used and the other conditional checks can be skipped and the immobilizer function executed.

If the LF field is NOT present, the initialization routine will eventually exit to the application code section after the next step. Transponder Initialization will not occur.

### 2.6.3 Enhanced Mode Detection

This command does not apply to the Atmel® ATA5580 and will be ignored.

### 2.6.4 Transponder Initialization

Once all conditions have been met to enter transponder mode, the following items are configured to prepare for communication:

- The presence of LF field has to be acknowledged in order to enable operation of the transponder
- System clocks are reconfigured
- System resources are configured for the lowest power consumption possible
- The interrupt vector table is mapped into bootloader space
- The watchdog timer is configured and activated
- System resources for Uplink and Downlink communication processing are initialized

### 2.6.5 Reliable Communication Channel Indication

Once the device has been initialized for transponder mode an indication of this readiness can be conveyed to the base station if selected during device configuration. This is achieved through the transmission of a detection header that ensures with high probability that the communication channel is open and reliable. Both the uplink and downlink paths are verified by this in the following manner.

For the downlink to be successful, the transponder must receive enough power to operate. Once this condition is satisfied for a long enough time to charge a buffer capacitor, the transponder can survive field gaps needed to transfer data. The fact that the initialization routine was successfully executed up to this point means this has been achieved.

For the uplink to be successful, the transponder must modulate the carrier field with sufficient coupling and modulation depth that the base station can recover the data from the carrier. By sending a modulated signal as defined by the detection header, the base station can make a determination that the uplink path is open once this header is visible on the demodulated output.

## 2.7 LF Physical Layer

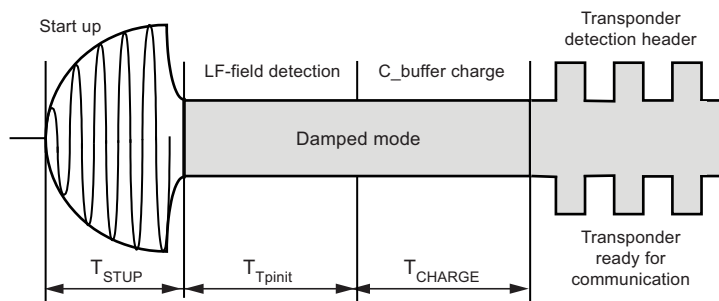
All communication between the Base Station and the Transponder occurs using the LF field as the signal carrier. The LF communication link is established when the Transponder transmits the LF channel detection header consisting of a Manchester coded sequence of “1010...” as a 125kHz signal which continues until the base station interrupts the signal during a damped phase with a gap.

**The physical layer (uplink and downlink) is compatible with all standard FDX basestations available on the market.**

LF channel consists of data communication sessions comprised of a downlink (Base Station to Transponder) and an uplink (Transponder to Base Station) data transfer.

Figure 2-5 shows a transponder start-up sequence after which the LF communication channel is established.

**Figure 2-5. LF Physical Layer**



## 2.7.1 Downlink

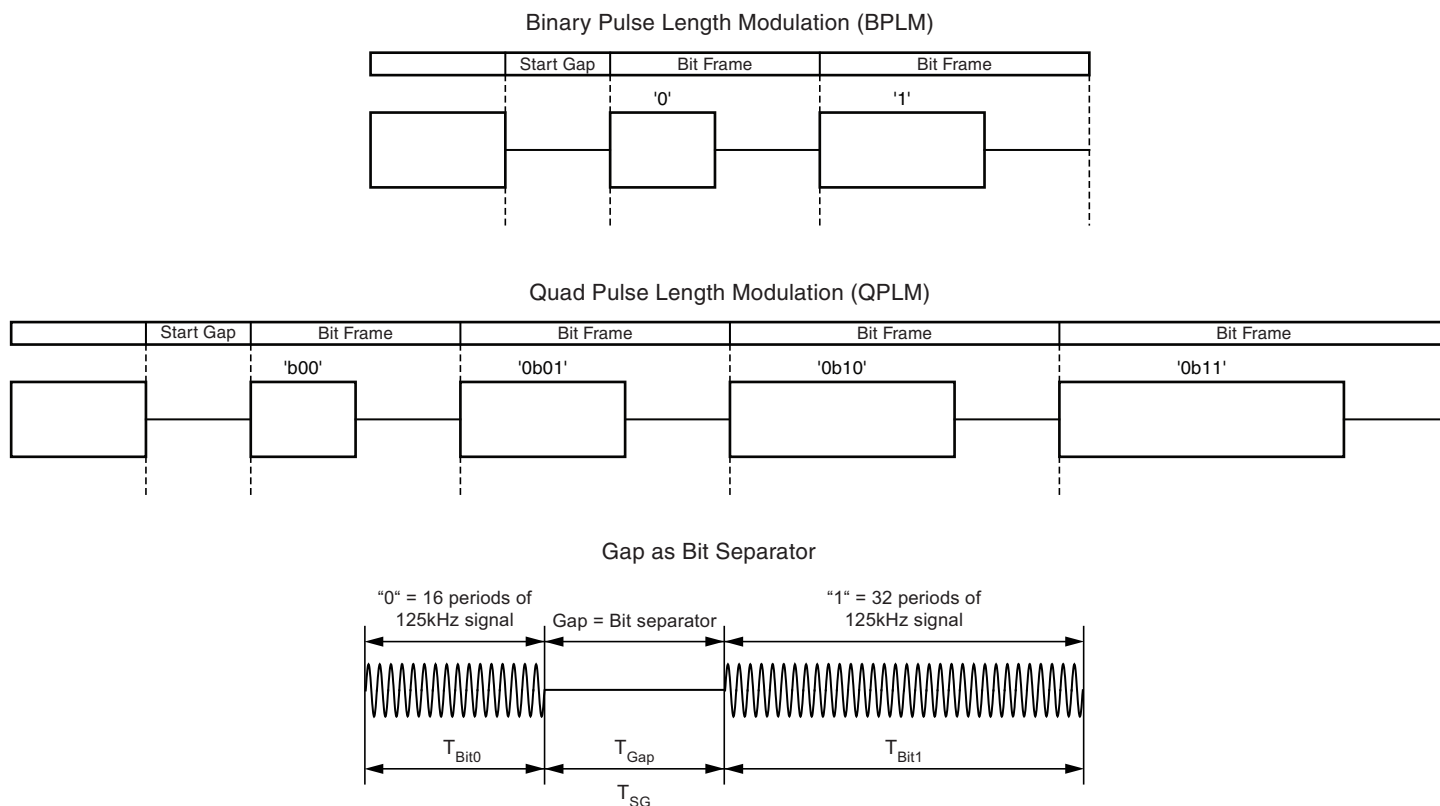
A downlink channel is established when the data is being transmitted from the base station to the transponder. The downlink communication uses Amplitude Modulation (AM) in the form of On-Off-Keying (OOK). To encode data pulse length coding is used. The pulses respectively LF bursts are separated by gaps. Data can be encoded in the following ways:

Binary Pulse Length Modulation (BPLM): Single pulse length is decoded to a single binary logic state (1-bit value).

QuadPulse Length Modulation (QPLM): Also known as 1-of-4 encoding, in this case a single pulse length is decoded into dual binary logic state (2-bit value).

Damped Phase Synchronized Modulation (DPS): While the transponder modulates the field with a sequential pattern of Manchester coded "0," the base station stops or continues sending the field during the second half of the bit (damped phase) to transmit "1s" or "0s".

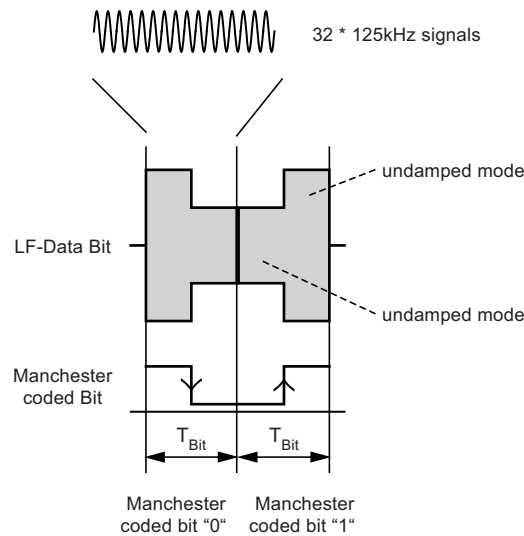
**Figure 2-6. Downlink**



2.7.2 Uplink

An uplink channel is established when the data is being transmitted from the transponder to the base station. The uplink communication utilizes AM by modulating the induced voltage on the transponder coil down to 50% of its un-damped amplitude (50% modulation depth). Binary data is either biphase or Manchester encoded.

Figure 2-7. Uplink (3.906 KB/s)



2.8 LF Communication

The protocol developed by Atmel® relies on two frame structures for the bi-directional communication. The downlink path from the base station to the transponder consists of a Request Frame. The uplink path uses the Response Frame defined below. Communication sessions consist of a base station request, a 2ms delay, and a transponder response. All communication follows this process and creates functionality by executing a series of communication sessions. The base station request contains the means to utilize the command set provided by the Atmel firmware. All commands have a defined response that is returned from the transponder. The command set indicates that this response only occurs if communication is successful. Any errors that occur cause the transponder to signal the base station in a unique manner by sending a fixed 1kHz waveform. This allows very rapid detection of a problem. The exact cause of the error is stored and can be accessed by a dedicated command.

2.8.1 Request Frame Definition

All transactions are initiated by the base station sending the following:  
Command Field = 4-bit command + 4-bit command CRC  
Data Field = Variable bit length payload (optional based on command)  
CRC Field = Payload CRC8 (optional based on presence of payload data)

| Command Field |      | Data Field | CRC Field   |
|---------------|------|------------|-------------|
| 4 bits        | CRC4 | Variable   | SW revision |

2.8.2 Response Frame Definition

All responses the transponder makes to the base station include sending the following:  
Header Field = recognizable pattern fixed at 0xFE  
Data Field = Variable bit length payload (optional based on command)  
CRC Field = Payload CRC8 (optional based on presence of payload data)

| Command Field |      | Data Field | CRC Field   |
|---------------|------|------------|-------------|
| 4 bits        | CRC4 | Variable   | SW revision |

2.9 LF Command Set

2.9.1 Read UID

The read UID command provides a very concise method for accessing the 32-bit unique serial number stored in the transponder. This serial number is assigned at the Atmel fabrication plant and provides a unique identity for use in the immobilizer system. The request from the base station is streamlined to provide a very rapid response consisting of only the 4-bit command and 4-bit CRC. The response contains the unique identifier. The EEPROM address designated for the unique identifier location starts from 0x810 and ends with 0x80D (4 bytes).

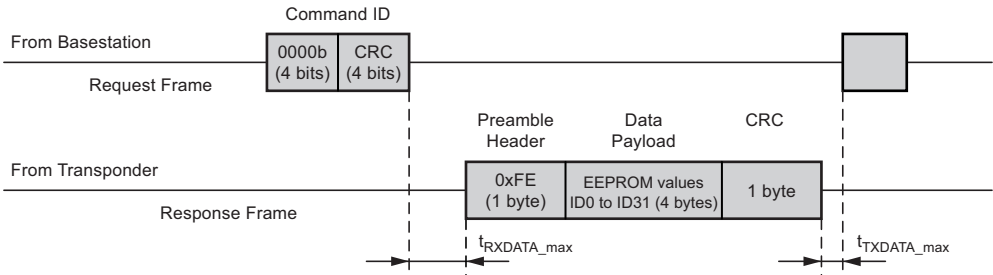
Table 2-1. The Read UID (Request Frame)

| Field        | Size       | Values           | Description |
|--------------|------------|------------------|-------------|
| Command ID   | 4 + 4 bits | 0000b + 0000 CRC | Read UID    |
| Data Payload | N/A        |                  |             |
| CRC          | N/A        |                  |             |

Table 2-2. The Read UID (Response Frame)

| Field           | Size    | Values       | Description                 |
|-----------------|---------|--------------|-----------------------------|
| Preamble Header | 1 byte  | 0xFE         | Synchronization             |
| Data Payload    | 4 bytes | EEPROM Value | Serial Number (ID0 to ID31) |
| CRC             | 1 byte  | Calculate    |                             |

Figure 2-8. The Read UID Sequence



2.9.2 Transponder Error Status

The Status byte contains both error information and command execution state information. By directly requesting this byte, the base station can determine the cause of an error or determine the last command executed. This allows a base station error to be remedied without complete loss of previously executed functions.

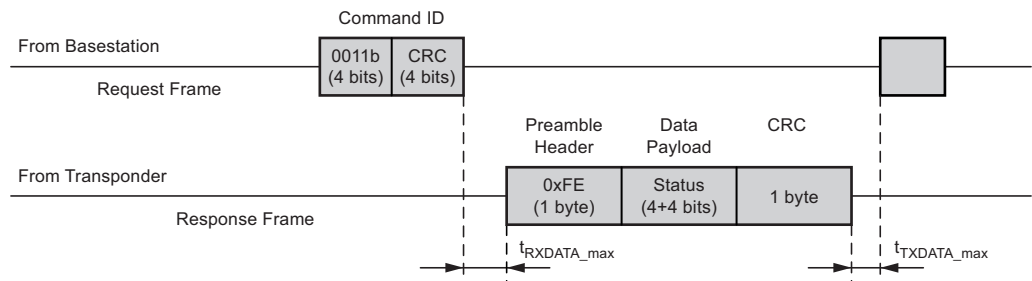
Table 2-3. The Transponder Error Status (Request Frame)

| Field        | Size       | Values           | Description         |
|--------------|------------|------------------|---------------------|
| Command ID   | 4 + 4 bits | 0010b + 0110 CRC | Request Status Byte |
| Data Payload | N/A        |                  |                     |
| CRC          | N/A        |                  |                     |

Table 2-4. The Transponder Error Status (Response Frame)

| Field           | Size   | Values    | Description     |
|-----------------|--------|-----------|-----------------|
| Preamble Header | 1 byte | 0xFE      | Synchronization |
| Data Payload    | 1 byte | Status    | Status          |
| CRC             | 1 byte | Calculate |                 |

Figure 2-9. The Transponder Error Status Sequence



2.9.3 Start Authentication

The immobilizer authentication protocol is to be based on Challenge - Response topology. This can be the Unilateral Authentication (UA) method or Bilateral Authentication (BA).

The start authentication command causes an Authentication protocol to begin. The length of the request payload (Challenge length) is dependent upon the setting stored in the EEPROM page 2 address 0x815 and the response length is dependent upon the setting stored at the EEPROM page 2 address 0x816.

The type of protocol that is used depends on the configuration stored at the EEPROM page 2 register address 0x811. Bit 2 (CM) defines the crypto model selected (0=UA or 1=BA). The authentication protocol can be selected based on security level and authentication time requirement. Every protocol implementation utilizes AES-128 block cipher encryption and depending on security level uses different variable bit length ciphers.

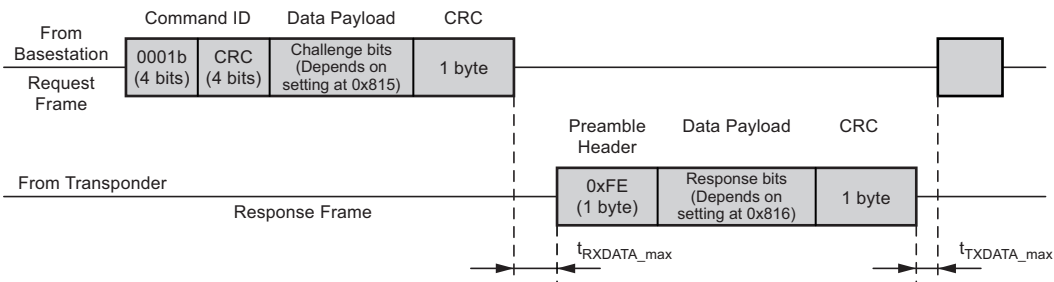
Table 2-5. Start Authentication (Request Frame)

| Field        | Size                                 | Values           | Description                     |
|--------------|--------------------------------------|------------------|---------------------------------|
| Command ID   | 4 + 4 bits                           | 0001b + 0011 CRC | Start authentication            |
| Data Payload | Varies (100 or 128 bits recommended) | Challenge bits   | Depends on EEPROM page2 setting |
| CRC          | 1 byte                               | Calculate        |                                 |

Table 2-6. Start Authentication (Response Frame)

| Field           | Size                               | Values        | Description                     |
|-----------------|------------------------------------|---------------|---------------------------------|
| Preamble Header | 1 byte                             | 0xFE          | Synchronization                 |
| Data Payload    | Varies (56 or 80 bits recommended) | Response bits | Depends on EEPROM page2 setting |
| CRC             | 1 byte                             | Calculate     |                                 |

Figure 2-10. The Start Authentication Sequence





## 2.9.4 Learn Secret Key1

This command starts the Learn Secret Key1 process for the first secret key. This is either open transfer or secure transfer depending on the configuration setting stored in EEPROM page 2 at address 0x811(bit 6). If this bit (SKT- Secure Key Transfer bit) is 0, the transfer is open mode and if the bit is 1, the transfer is secure mode. The request frame carries a 128-bit secret key data payload (may be encrypted during secure transfer). The 128-bit key transferred through this command is stored in AP0 key position 1 (0x7C0) along with two copies. The response frame consists of a status byte at the data payload. The Status byte is stored in RAM and updated with each communication session. The status byte consists of the last command received (MS 4bits) and an error flag (LS 4bits). Status Byte [7:4]: Four MSBs of the field contain an echo of the command received in the last Request Frame. Status Byte [3:0]: Four LSBs of the field contain status information in encoded form.

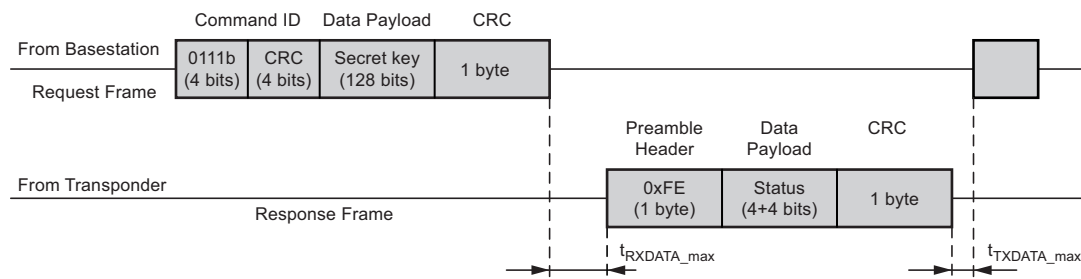
**Table 2-7. The Learn Secret Key1 (Request Frame)**

| Field        | Size       | Values           | Description                         |
|--------------|------------|------------------|-------------------------------------|
| Command ID   | 4 + 4 bits | 0111b + 1001 CRC | Learn Secret Key1                   |
| Data Payload | 128 bits   |                  | AES (possibly encrypted) secret key |
| CRC          | 1 byte     | Calculate        |                                     |

**Table 2-8. Learn Secret Key1 (Response Frame)**

| Field           | Size   | Values    | Description     |
|-----------------|--------|-----------|-----------------|
| Preamble Header | 1 byte | 0xFE      | Synchronization |
| Data Payload    | 1 byte | Status    | Status          |
| CRC             | 1 byte | Calculate |                 |

**Figure 2-11. The Learn Secret Key1 Sequence**



## 2.9.5 Learn Secret Key2

This command starts the secret key 2 learning process. This is either open or secure transfer depending on the configuration stored in EEPROM at address 0x811(bit 6). If this bit (SKT- Secure Key Transfer bit) is 0, the transfer is open mode and if the bit is 1, the transfer is secure mode. The request frame carries 128-bit secret key data payload (may be encrypted during secure transfer). The 128-bit key transferred through this command is stored in the AP1 key position 2 (0x780) along with two copies. The response frame consists of a status byte at the data payload. The Status byte is stored in RAM and updated with each communication session. The status byte consists of the last command received (MS 4bits) and an error flag (LS 4bits).

Status Byte [7:4]: Four MSBs of the field contain an echo of the command received in the last Request Frame. Status Byte [3:0]: Four LSBs of the field contain status information in encoded form.

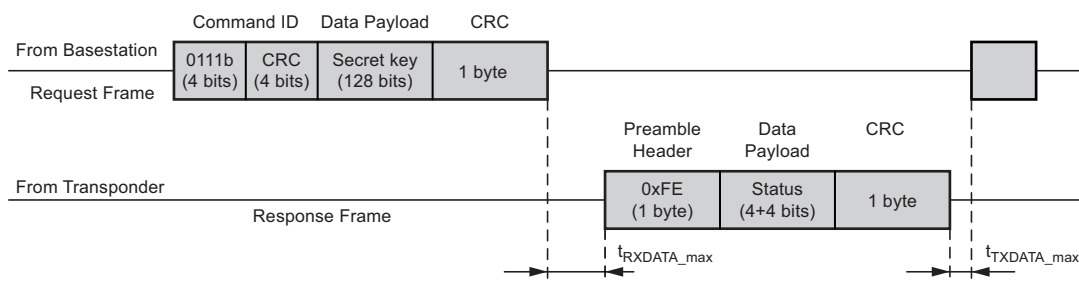
**Table 2-9. The Learn Secret Key2 (Request Frame)**

| Field        | Size       | Values           | Description                         |
|--------------|------------|------------------|-------------------------------------|
| Command ID   | 4 + 4 bits | 1000b + 1011 CRC | Learn Secret Key2                   |
| Data Payload | 128 bits   |                  | AES (possibly encrypted) secret key |
| CRC          | 1 byte     | Calculate        |                                     |

**Table 2-10. Learn Secret Key2 (Response Frame)**

| Field           | Size   | Values    | Description     |
|-----------------|--------|-----------|-----------------|
| Preamble Header | 1 byte | 0xFE      | Synchronization |
| Data Payload    | 1 byte | Status    | Status          |
| CRC             | 1 byte | Calculate |                 |

**Figure 2-12. The Learn Secret Key 2 Sequence**



2.9.6 Initiate Enhanced Mode

This command initializes the Enhanced Mode Command Structure. This command sets the transponder into enhanced Mode when it enters the VFLD the next time by setting the enhanced mode flag in EEPROM. This command begins a sequence to place the transponder into the enhanced mode where the battery supply is used during transponder communication. An EEPROM flag of value TBD is stored at address TBD. This address is checked at each POR to determine if the power switch should be disabled. Once the flag is set by this LF Command, the NEXT power cycle causes the following LF session to be operated using battery power. This will occur only once each time this LF Command is received.

The status byte consists of the last command received (MS 4bits) and an error flag (LS 4bits).

Status Byte [7:4]: Four MSBs of the field contain an echo of the command received in the last Request Frame. Status Byte [3:0]: Four LSBs of the field contain status information in encoded form.

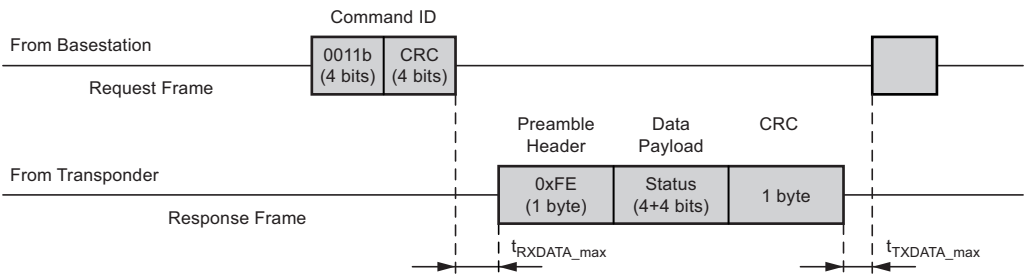
Table 2-11. The Initiate Enhanced Mode (Request Frame)

| Field        | Size       | Values           | Description            |
|--------------|------------|------------------|------------------------|
| Command ID   | 4 + 4 bits | 0011b + 0101 CRC | Initiate enhanced mode |
| Data Payload | N/A        |                  |                        |
| CRC          | N/A        |                  |                        |

Table 2-12. The Initiate Enhanced Mode (Response Frame)

| Field           | Size   | Values    | Description     |
|-----------------|--------|-----------|-----------------|
| Preamble Header | 1 byte | 0xFE      | Synchronization |
| Data Payload    | 1 byte | Status    | Status          |
| CRC             | 1 byte | Calculate |                 |

Figure 2-13. The Initiate Enhanced Mode Sequence



2.9.7 Repeat Last Response

This command requests that the last transmission be repeated. This quickly repeats the last response used. This enables a retry strategy that increases communication response time.

The response frame matches the response from the previous command.

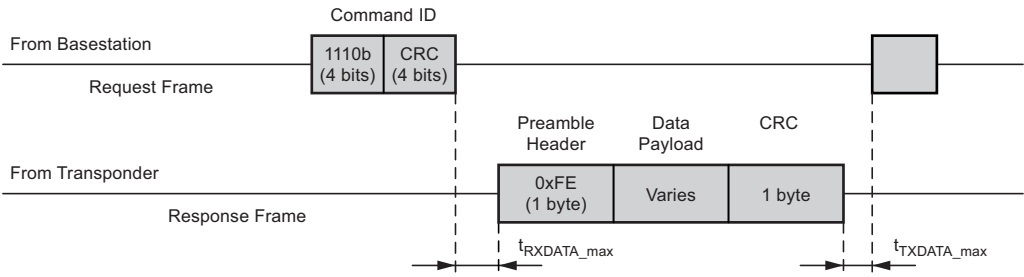
Table 2-13. Repeat Last Response (Request Frame)

| Field        | Size       | Values           | Description          |
|--------------|------------|------------------|----------------------|
| Command ID   | 4 + 4 bits | 1110b + 0001 CRC | Repeat last response |
| Data Payload | N/A        |                  |                      |
| CRC          | N/A        |                  |                      |

Table 2-14. Repeat Last Response (Response Frame)

| Field           | Size   | Values    | Description     |
|-----------------|--------|-----------|-----------------|
| Preamble Header | 1 byte | 0xFE      | Synchronization |
| Data Payload    | Varies | Status    |                 |
| CRC             | 1 byte | Calculate |                 |

Figure 2-14. The Repeat Last Response Sequence



## 2.9.8 Read User Memory

This command provides memory read operation from the user memory (EEPROM). The request frame data block provides the beginning address of the EEPROM as well as the Read Length (number of bytes that should be read). Addresses in the range of (0x0780 to 0x07FF) or (0x0817 to 0x0826) should NEVER be allowed access via the memory access commands. The transponder provides the status byte as well as requested number of EEPROM data bytes in the response frame. The Response Length specified does not exceed 16 bytes. The status byte consists of the last command received (MS 4bits) and an error flag (LS 4bits).

Status Byte [7:4]: Four MSBs of the field contain an echo of the command received in the last Request Frame. Status Byte [3:0]: Four LSBs of the field contain status information in encoded form.

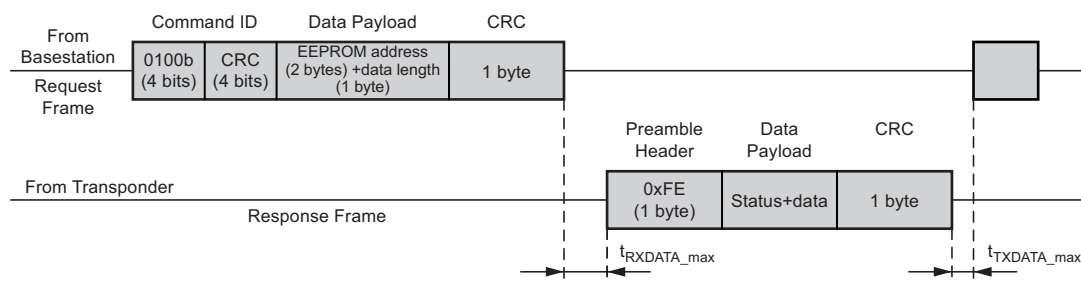
**Table 2-15. Read User Memory (Request Frame)**

| Field        | Size             | Values           | Description                  |
|--------------|------------------|------------------|------------------------------|
| Command ID   | 4 + 4 bits       | 0100b + 1100 CRC | Read User Memory             |
| Data Payload | 2 bytes + 1 byte |                  | EEPROM address + data length |
| CRC          | 1 byte           | Calculate        |                              |

**Table 2-16. Read User Memory (Response Frame)**

| Field           | Size          | Values        | Description          |
|-----------------|---------------|---------------|----------------------|
| Preamble Header | 1 byte        | 0xFE          | Synchronization      |
| Data Payload    | 1 byte + data | Status + data | Status + EEPROM data |
| CRC             | 1 byte        | Calculate     |                      |

**Figure 2-15. The Read User Memory Sequence**



## 2.9.9 Write User Memory

This command provides write operation to the user memory (EEPROM). The request frame data block provides the beginning address of the EEPROM followed by the data to be written. The transponder provides the status of the result in the response frame. Write commands that involve transponder EEPROM addresses with the AP1, AP2, and AP3 sections initially check the saved lock state for this section. If the section has previously been locked, the command is aborted and the transponder sends an error response. During Normal operation the number of EEPROM data bytes to be written should be 4 bytes at the most. During Enhanced mode the number of EEPROM data bytes to be written should not exceed 128 bytes. The EEPROM data is always sent as complete bytes.

The status byte consists of the last command received (MS 4bits) and an error flag (LS 4bits).

Status Byte [7:4]: Four MSBs of the field contain an echo of the command received in the last Request Frame. Status Byte [3:0]: Four LSBs of the field contain status information in encoded form.

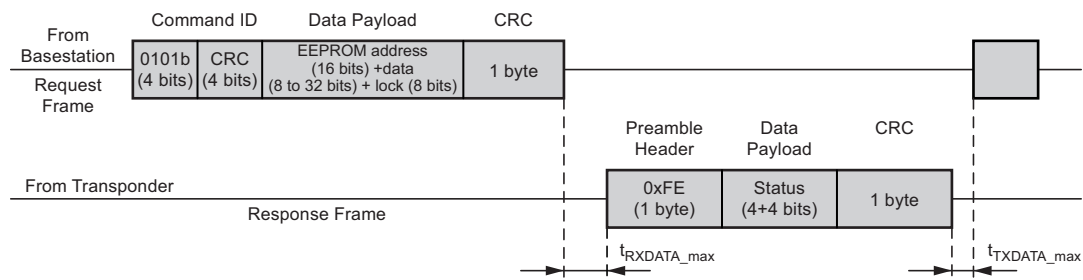
**Table 2-17. Write User Memory (Request Frame)**

| Field        | Size                            | Values           | Description                |
|--------------|---------------------------------|------------------|----------------------------|
| Command ID   | 4 + 4 bits                      | 0101b + 1111 CRC | Read User Memory           |
| Data Payload | 16 bits + 1 to 4 bytes + 8 bits |                  | EEPROM address + data lock |
| CRC          | 1 byte                          | Calculate        |                            |

**Table 2-18. Write User Memory (Response Frame)**

| Field           | Size   | Values    | Description     |
|-----------------|--------|-----------|-----------------|
| Preamble Header | 1 byte | 0xFE      | Synchronization |
| Data Payload    | 1 byte | Status    | Status byte     |
| CRC             | 1 byte | Calculate |                 |

**Figure 2-16. The Write User Memory Sequence**



## 2.9.10 Write Memory Access Protection

This command protects only the AP1, AP2, AP3 sections from being overwritten through transponder memory access commands (LF field commands). Once protection has been applied, it is not removed (sending 00b does not clear the locks). The request frame data block consists of binary 00+AP3+AP2+AP1 to create one byte. To lock each section the command transmits b11 in that section and b00 if section locking is not required (ex. 00110011 locks AP3 and AP1 and leaves section AP2 unlocked). The use of two bits for each memory section protects against accidental locking due to one-bit corruption.

The status byte consists of the last command received (MS 4 bits) and an error flag (LS 4 bits).

Status byte [7:4]: Four MSBs of the field contain an echo of the command received in the last request frame. Status byte [3:0]: Four LSBs of the field contain status information in encoded form.

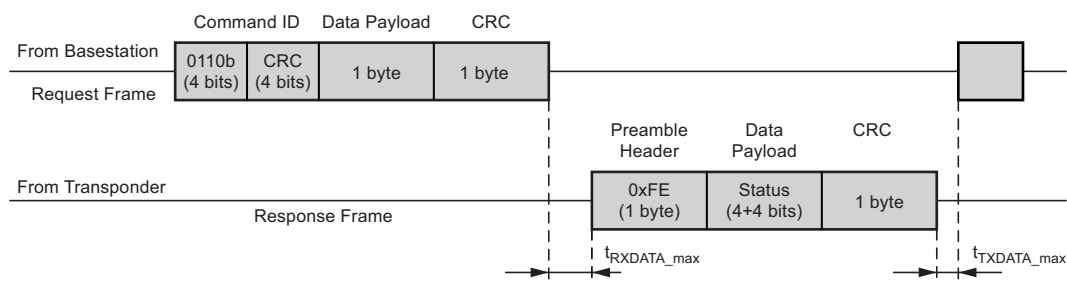
**Table 2-19. Write Memory Access Protection (Request Frame)**

| Field        | Size       | Values           | Description                    |
|--------------|------------|------------------|--------------------------------|
| Command ID   | 4 + 4 bits | 0110b + 1010 CRC | Write Memory Access Protection |
| Data Payload | 1 byte     |                  | Protection scheme              |
| CRC          | 1 byte     | Calculate        |                                |

**Table 2-20. Write Memory Access Protection (Response Frame)**

| Field           | Size   | Values    | Description     |
|-----------------|--------|-----------|-----------------|
| Preamble Header | 1 byte | 0xFE      | Synchronization |
| Data Payload    | 1 byte | Status    | Status byte     |
| CRC             | 1 byte | Calculate |                 |

**Figure 2-17. The Write Memory Access Protection Sequence**



### 2.9.11 Leave Enhanced Mode

This command will clear the enhanced mode flag from EEPROM.

If the transponder receives the command “Leave Enhanced Mode”, the internal power switch inside the transponder front end will be enabled. If the LF field is active the internal power management will switch automatically to the field supplied mode. This will then generate a power on reset and the immobilizer firmware will be executed.

**Table 2-21. Leave Enhanced Mode (Request Frame)**

| Field        | Size       | Values            | Description         |
|--------------|------------|-------------------|---------------------|
| Command ID   | 4 + 4 bits | 1010b + 1101b CRC | Leave Enhanced Mode |
| Data Payload | N/A        |                   |                     |
| CRC          | N/A        |                   |                     |

**Table 2-22. Leave Enhanced Mode (Response Frame)**

| Field           | Size   | Values    | Description  |
|-----------------|--------|-----------|--|
| Preamble Header | 1 byte | 0xFE      | Synchronization                                    |
| Data Payload    | 1 byte | Status    | [7:4] previous command<br>[3:0] encoded error info |
| CRC             | 1 byte | Calculate |  |

### 2.10 Communication Integrity and Error Mitigation

The commands are protected from transmission channel corruption by the use of a CRC nibble. This prevents accidental processing of an unintended command due to bit corruption. The data can be protected through a second CRC byte. This is true for communication in both the uplink and downlink direction. The use of this fast detection of bit level corruption allows a highly efficient retry strategy to be implemented. When this is combined with the Repeat Last Response command, uplink errors can be quickly and automatically mitigated.

The following is suggested as means of progressive retries for downlink errors:

- Error detected on downlink communication due to error signal response
- Request status byte to determine the cause of error
- Resend downlink request if error was due to failed downlink CRC
- If error still persists, reset transponder completely via command or removing of LF Field

The following is suggested as a means of progressive retries for uplink errors:

- Error detected on uplink communication via failed CRC check
- Request repeat transmission with Repeat Last Response command
- If error still occurs, repeat complete communication by resending the desired command request frame
- If error still persists, reset transponder completely via command or removing of LF Field



### 3. Immobilizer Functionality

This section describes the steps required to implement the immobilizer system functionality. This functionality can be achieved in the base station and vehicle controller by using features and commands provided by Atmel®. The following sections recommend how this can be achieved.

#### 3.1 Authentication

The heart of the vehicle immobilizer is the ability to identify the user as authorized to start the vehicle. There are many different authentication schemes. Each has different affects on response time and security. In order to provide the customer with a wide array of options, Atmel has developed a command and feature set that provides a high level of configurable authentication options including the choice of either unilateral or bilateral means of authentication.

##### 3.1.1 Unilateral Authentication

Unilateral authentication is a strategy where authentication is performed by only one entity in the system. The other entity simply responds to any command that it receives. In the case of a vehicle immobilizer system, the vehicle attempts to verify the identity of the key fob. The benefit of this approach is that a high level of security can be achieved without sacrificing system response time.

Unilateral authentication should be initiated by the base station and conform to the following sequence:

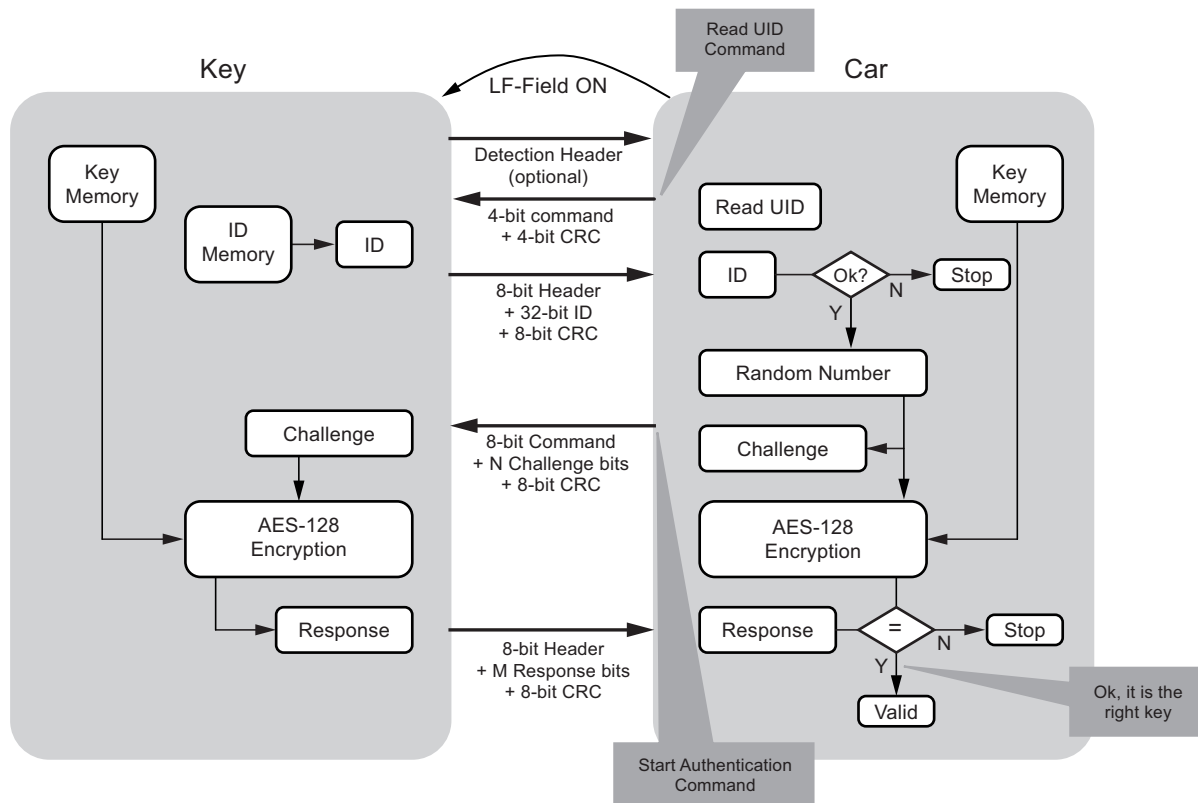
1. The base station sends the LF request “Read UID”.
2. The transponder responds by providing the 32-bit UID in its “Response Frame”.
3. The base station then sends the “Start Authentication” request which includes a random number “Challenge”.
4. The transponder returns an “encrypted Response” message to the base station.

Notes:

1. The “Challenge” will use bit length defined by configuration memory address 0x0819.
2. The Secret Key can be either Key1 or Key2 as defined by configuration memory address 0x0815 Bit 5.
3. The “Response” will use bit length defined by configuration memory address 0x081A.
4. When necessary for encryption, the challenge will be extended by first padding the upper bit positions with the 32-bit UID, then with “0”s as needed, and in this order, to attain 128 bits.

A graphical example is shown in [Figure 3-1 on page 26](#).

**Figure 3-1. Unilateral Authentication Protocol**



#### 3.1.1.1 Read UID

The Read UID command has been optimized to enhance the speed of the authentication. The request from the vehicle consists only of 8bits. The response contains a 32-bit unique serial number that can be used for rough authentication to determine if this key is potentially paired with the vehicle.

#### 3.1.1.2 Start Authentication

The encrypted authentication is initiated with the Start Authentication request that provides the challenge data. Atmel® recommends choosing 100 or 128 bits for the challenge length. The encrypted response should be chosen as 56 or 80 bits respectively. The reason for these choices would be to achieve a high level of security while optimizing the speed for the entire communication. The total number of bits transferred is 188 and 240 respectively. This works out to a bit security level of 50 and 64 bits for these two options. The attacker would need to attempt more than one trillion trials to break the security. The 128 bit secret key that is used can be chosen from one of two possible locations.

### 3.1.2 Bilateral Authentication

Bilateral authentication is a strategy where authentication is performed by both entities in the system. Each side attempts to ensure that they are only communicating with an approved and previously paired system entity. In the case of a vehicle immobilizer system, the transponder first verifies that the vehicle is approved. Once this has been established, the transponder provides the means for the vehicle to verify that the transponder is approved. The benefit of this approach is that a mutually secure system can be achieved within a reasonable system response time. It also provides the transponder a way to detect and defeat attacks from “unapproved” base stations.

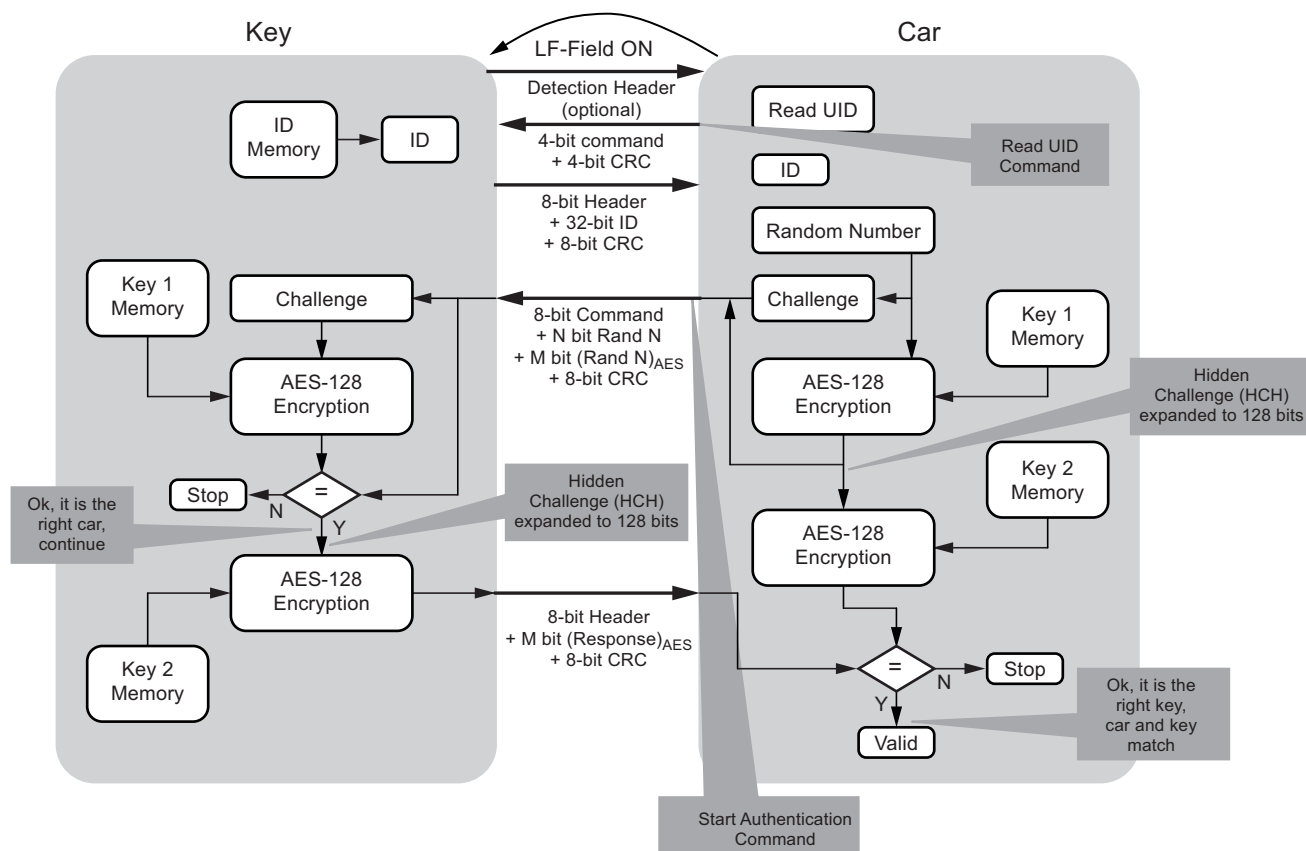
Bilateral authentication shall be initiated by the base station and conform to the following sequence:

1. The base station shall send the LF command “Read UID”.
2. The transponder shall respond by providing the 32-bit UID in its “Response Frame”.
3. The base station shall send the LF command “Start Authentication” which includes a random number “Challenge” followed by an AES encrypted version of the “Challenge” using one of its two Secret Keys.
4. The transponder will check the “encrypted Challenge” to verify it matches the transponder’s calculated value for “encrypted Challenge” (using the same Secret Key that created the “encrypted Challenge” in the base station).
5. The transponder will create an “encrypted Response” if the verification in step 4 was successful. It will use the full 128-bit “encrypted Challenge” not just the subset sent from the base station and the other of the two Secret Keys as AES block cipher inputs to form the encrypted “Response”.
6. The base station will compare transponder’s “encrypted Response” with its calculated value for encrypted “Response” following the same process used in step 5. If they match, bilateral authentication was successful.

- Notes:
1. The “Challenge” will use bit length defined by configuration memory address 0x0819.
  2. The initial Secret Key can be either Key1 or Key2 as defined by configuration memory address 0x0815 bit 5.
  3. The “encrypted Challenge” and “encrypted Response” will have bit length defined by configuration memory address 0x081A.
  4. The other secret key will be used to create the “encrypted Response”.
  5. When necessary, inputs for calculating the “encrypted Challenge” and “encrypted Response” will be extended by first padding the upper bit positions with the 32-bit UID, then with “0”s as needed, and in this order, to attain 128bits.

A visual representation is noted in [Figure 3-2 on page 28](#).

### Figure 3-2. Authentication BA



### 3.1.3 Read UID

The Read UID command has been optimized to enhance the speed of the authentication. The request from the vehicle consists only of 8bits. The response contains a 32-bit unique serial number that can be used for rough authentication to determine if this key is potentially paired with the vehicle.

### 3.1.4 Start Authentication

The Start Authentication command begins with sending a challenge followed by the output of an encryption of this challenge with an initial secret key. This “encrypted Challenge” serves to authenticate the vehicle identity to the transponder and prove that the vehicle is a valid partner with whom the transponder can communicate. The lengths of both of these are adjustable in the configuration options but Atmel recommends that a challenge length of 100 bits and encrypted challenge of 56 bits. If this fails, the transponder simply sends an error signal back. In the case of a successful vehicle authentication, the transponder would calculate the response to the vehicle using the hidden challenge and the remaining secret key. This is the same length as the “encrypted Challenge” and we recommend this be set to 56 bits.

This response can be evaluated by the vehicle to determine authenticity of the transponder. The total number of bits transferred is 244 and provides a bit security level of 50. This approach also is strengthened by the use of two separate 128-bit secret keys. Each secret key protects one direction of authentication meaning that compromising one secret key does not break the complete bilateral authentication protocol.

### 3.1.5 Hidden Challenge

Another aspect of this protocol is the use of a "hidden" challenge as the input to the second encryption stage. The reason this is considered "hidden" is that only a portion of this value is ever transmitted over the wireless interface. Using the recommended values from above, we see that the input to the second encryption block contains the 56 bit "encrypted Challenge" that was used to determine the authenticity of the vehicle. While this value was sent over the air and could be recorded, the second encryption block requires that the complete 128-bit output of the first encryption be known precisely. Since only 56 bits could be captured, this leaves 72 bits that are "hidden" from the attacker but are critical to producing the correct output. Through this scheme we are able to allow a truncated initial challenge to be expanded to a full 128-bit AES operation when producing the response used to validate the transponder identity. This final step is what protects against unauthorized vehicle starts and our system provides maximum protection in this area.

## 3.2 Memory Access

General purpose memory is a very important part of an immobilizer system. Atmel has provided a large EEPROM section in hardware and a very efficient means of accessing this through LF commands. The block size for access is flexible and allows the end system designer to build structures that are optimized for the data content. The only areas that are not accessed through the memory commands are the AP0 section used for secret keys and the Default Secret key stored in EEPROM page 2. All other memories providing an interface for the vehicle to interact with the application functionality can be accessed. For example, the vehicle could resynchronize with the RKE rolling code counter, read out user specific information, or store diagnostic trouble codes.

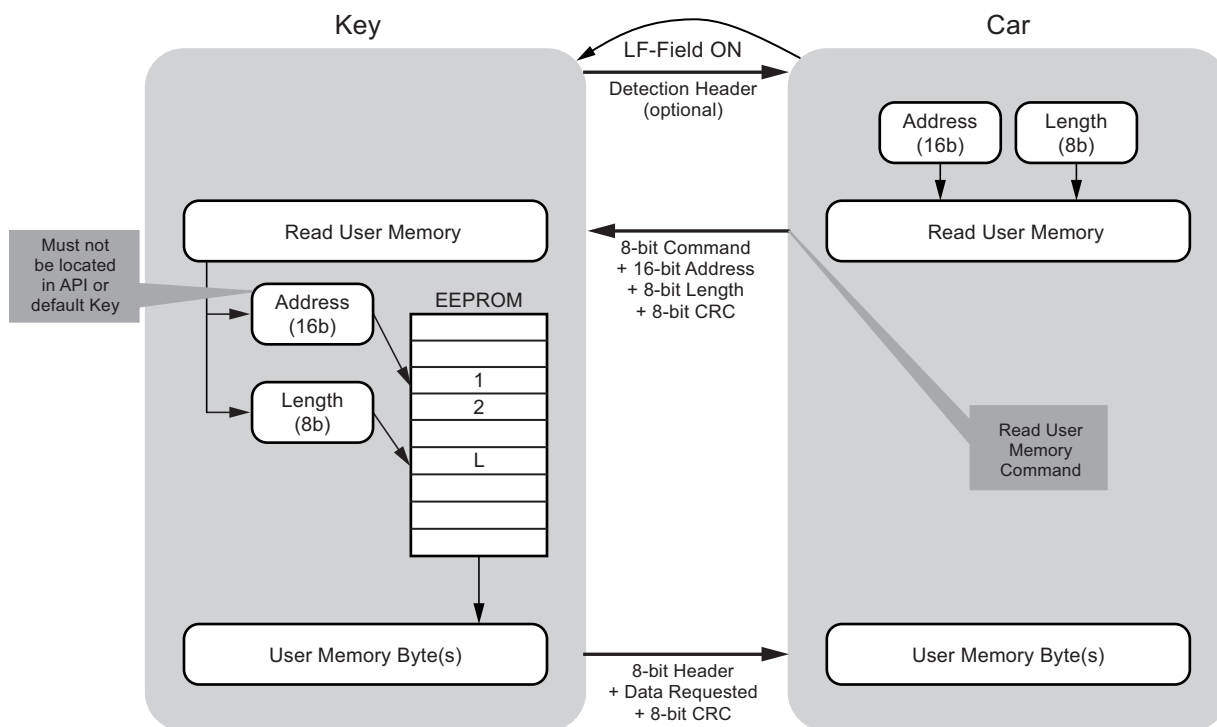
For higher security, if the transponder is configured to use Bilateral Authentication, an authentication session must be successfully accomplished before any memory access command is possible.

### 3.2.1 Read Memory

To read user memory only requires that the starting address and the number of bytes requested are provided. This allows block sizes from one to sixteen bytes to be accessed from the transponder non-volatile memory. The memory is accessed and the data returned starting with the first address and incrementing sequentially until all bytes are sent.

The flexibility of this command means it can be used for many functions that would normally require a dedicated LF Command. Examples of this are shown in other sections of this document.

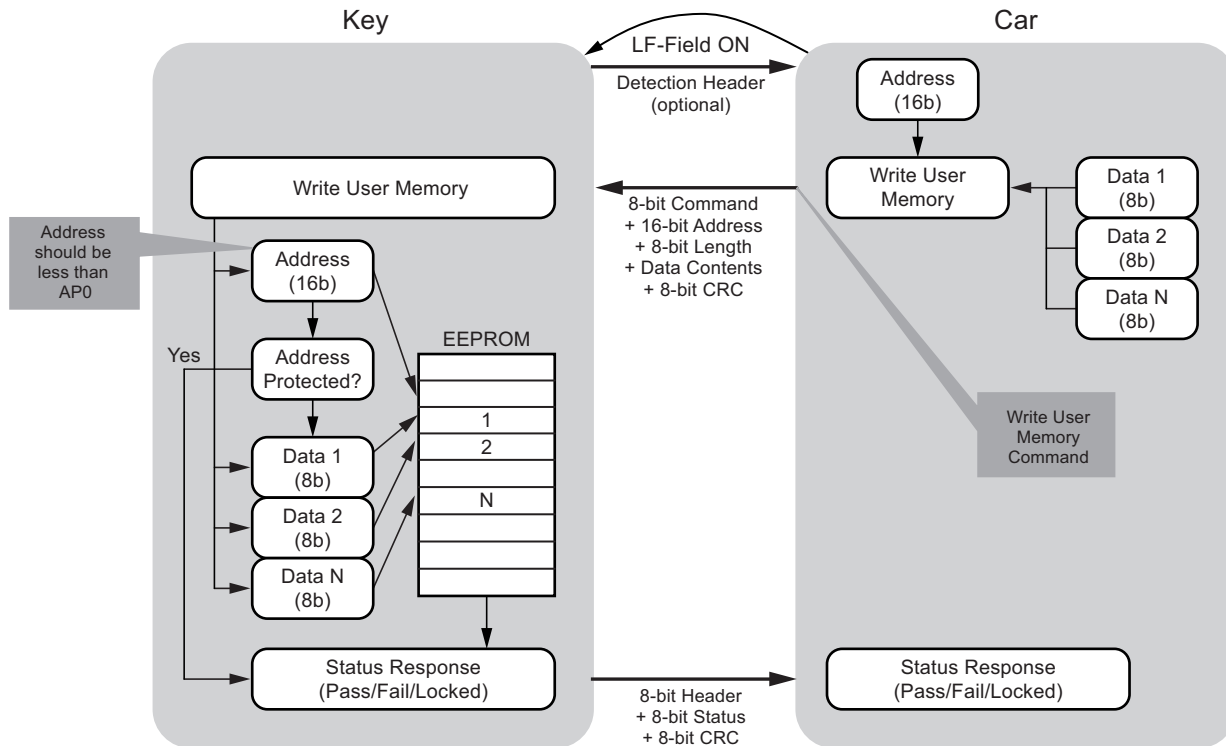
**Figure 3-3. Read Memory**



### 3.2.2 Write Memory

Writing data into the memory requires the starting address to be provided followed by the number of data bytes to be stored. The length of this block is limited to four bytes (128 bytes in enhanced mode) and must always be sent as full 8-bit multiples. Before the memory location is written, the firmware checks to see if access protection applies and determine if this command is allowed. Only if these checks are successful is the data written into EEPROM.

**Figure 3-4. Write Memory**



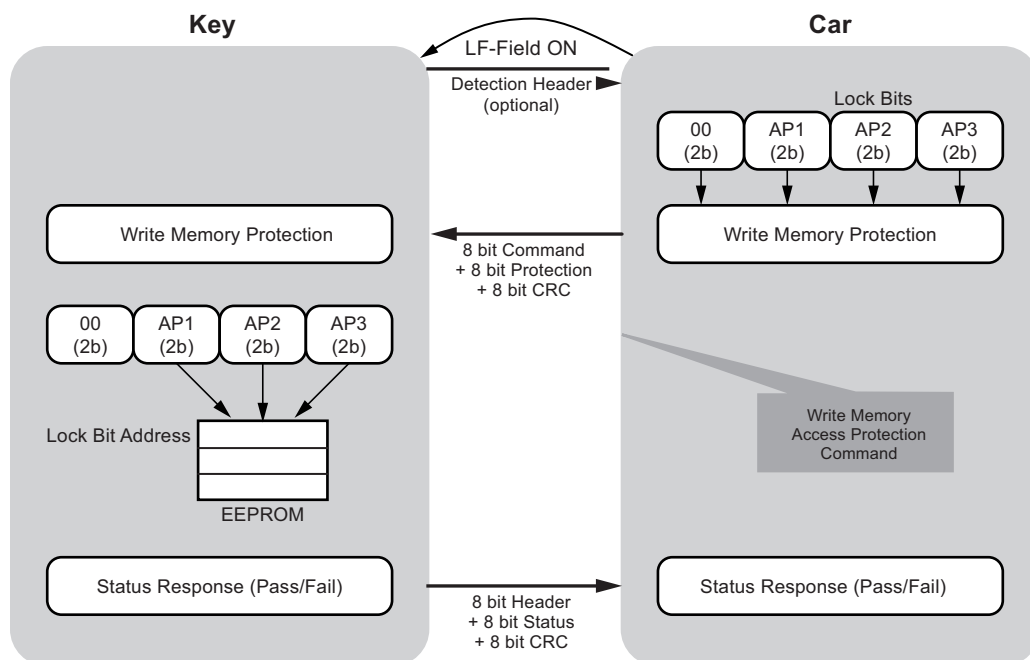
### 3.2.3 Memory Protection

Memory protection provides a means to prevent EEPROM data from being modified by future LF Commands. Once this protection is applied, it cannot be removed by a subsequent LF command. The protection applies to a complete section of EEPROM. There are three such sections that can be used. They are defined as the AP1, AP2, and AP3 portions of EEPROM and contain 128 bytes in each section. One example of this might be a block of manufacturing process information programmed into AP1 and locked so that this cannot be modified by LF commands. This allows a returned device to be traced back through the precise manufacturing chain.

The locking feature is implemented in firmware and does not contain any hardware components. The protection applies to the reaction to LF commands received by the immobilizer.

The write memory access protection command requires only one byte with the protection to be assigned to each section. Two bits are used for each memory section to add extra protection against false locking scenarios. Both bits must be set to logical one for the protection to be invoked. All unused locations should be set to logic zero which do not change the protection currently invoked.

**Figure 3-5. Write Memory Protection**



Note: Only when the lock bits are set "11" will be written to EEPROM.  
Lock bits set to "00" will NOT clear already set bits, i.e. XOR with current EEPROM data.

### 3.2.4 Memory Encryption

Encryption of the data is not provided through a special command or the immobilizer firmware. With the hardware encryption block, the need for this functionality can be implemented before the data is placed in the non-volatile memory by the application or before it is sent from the base station. Memory encryption can be easily decrypted by the application before it is used. For example, a rolling code counter can be encrypted and stored in memory. Each time this is needed, the application decrypts it, uses the counter, increments it, encrypts the new count, and then stores this back in memory.

### 3.3 Identification

One of the primary goals of the immobilizer is to establish the verified identity of the user. The firmware provided by Atmel® offers many identification options that allow the system to be optimized. The following sections describe how the fixed identification aspects can be used. It is possible to achieve customized identification scenarios by using the memory access commands and custom block sizes as needed.

#### 3.3.1 Serial Number

The serial number is a fixed value programmed and locked by Atmel during manufacturing. This value is a 32-bit, non-sequential, non-repeating number and is optimized for fast initial identification. A dedicated LF command (Read UID) allows this value to be accessed prior to authentication for a very rough screening of users.

#### 3.3.2 Atmel Traceability

Atmel provides manufacture traceability from our process flow to directly identify a given device. This information is fixed and locked at the end of our manufacture line. It provides very useful information about the device and also uniquely associates this with a physical die location on a wafer. Each of the following pieces of information can be accessed individually or as a unit with the Read Memory command.

- Device type: This contains information that specifies which Atmel device this is
- Lot Number: This specifies the Atmel facility and the production lot run that created this device
- Wafer Number: This designates the physical wafer in this lot
- Die Number: This locates the die on the wafer
- Software Rev: This indicates the firmware release version that is currently running



## 3.4 Personalization

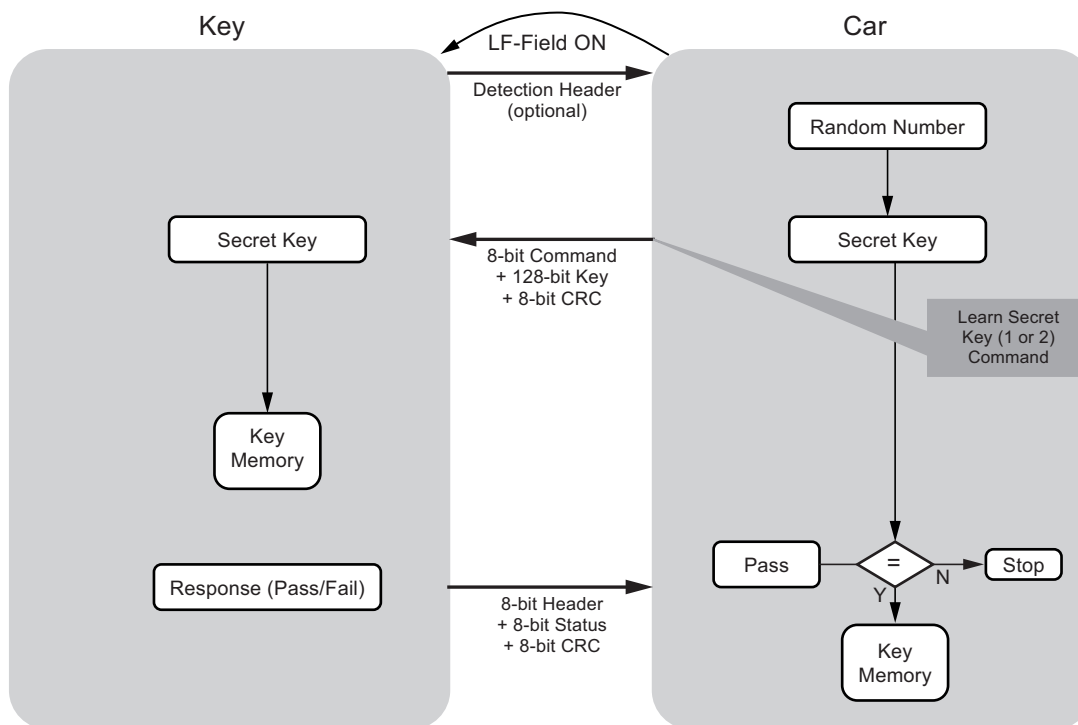
Personalization refers to the process of setting or resetting the initial parameters of the device. In the case of the immobilizer this involves pairing the transponder with the vehicle. The most common pairing scenario is the transfer of the secret key(s) from the vehicle to the transponder. Other personalization parameters can be set through the use of the Write Memory command. These could be the initial Roll Code, application Feature configuration, vehicle VIN, etc. The following section will look at the options possible for secret key transfer.

### 3.4.1 Open Key Learn

If the security of the key transfer can be ensured through physical or other security methods it may be desirable to send the secret key in plain text. The firmware can be configured to allow this and the following sequence would occur:

- The base station sends 128bits of secret key coding to be stored using the Learn Secret Key command.
- The transponder stores the encoded key in the AP0 section of EEPROM in key position 1 or 2.

Figure 3-6. Open Key Learn 1/2

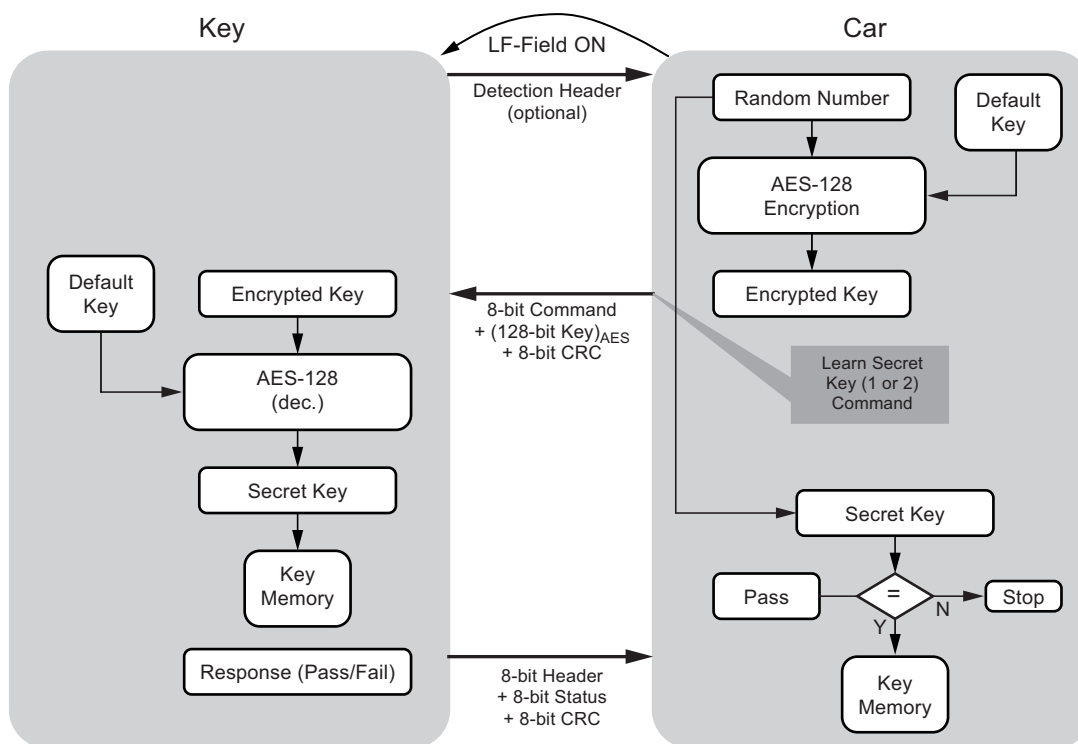


### 3.4.2 Secure Key Learn

Because the encryption key protects the integrity of the authentication process Atmel has provided a means to transfer the secret key in an encrypted manner. This involves the use of the default secret key stored in EEPROM page 2 and protects against eavesdropping by an attacker during key transfer. As a result, secure implementation of user-initiated personalization is possible where physical security cannot be ensured.

- The base station sends 128bits of data that have been encrypted using the default key stored in EEPROM page 2.
- The transponder decodes this to produce the secret key to be stored.
- The transponder then stores the encoded key in the AP0 section of EEPROM in key position 1 or 2.

Figure 3-7. Secure Key Learn



## 4. Abbreviations

FDX – Full Duplex  
AM – Amplitude Modulation  
BCM – Body Control Module  
ECU – Electronic Control Unit  
BPLM – Binary Pulse Length Modulation  
QPLM – Quad Pulse Length Modulation  
POR – Power on Reset  
TIC – Transmitter ID Code  
RKE – Remote Keyless Entry  
DPS – Damped Phase Synchronized  
VFLD – Field Voltage

## 5. Absolute Maximum Ratings

Stress beyond those listed under “Absolute Maximum Ratings” may cause permanent damage to the device. This is a stress rating only and functional operation of the device in these or any other conditions beyond those indicated in the operational sections of this specification is not intended. Exposure to absolute maximum rating conditions for extended periods may affect device reliability.

| Parameters   | Symbol    | Value       | Unit |
|--|-----------|-------------|------|
| Operating temperature range                        | $T_{amb}$ | –40 to +85  | °C   |
| Storage temperature range (data retention reduced) | $T_{amb}$ | –40 to +125 | °C   |
| Maximum assembly temperature, $t < 5\text{min}$    | $T_{ass}$ | 170         | °C   |
| Magnetic field strength at $f = 125\text{kHz}$     | $H_{pp}$  | 1000        | A/m  |

## 6. Operating Characteristics

$T_{amb} = +25^{\circ}\text{C}$ ;  $f_{coil} = 125\text{kHz}$ ; unless otherwise specified.

| No. | Parameters  | Test Conditions                                | Symbol                | Min.     | Typ.         | Max. | Unit          | Type* |
|-----|---|--|-----------------------|----------|--------------|------|---------------|-------|
| 1   | Coil inductance   |  | L                     |          | 2.38         |      | mH            | Q     |
| 2.1 | LC Circuit  | $H_{pp} = 14.5\text{A/m}$                      | $f_{res}$             | 119      | 125          | 131  | kHz           | T     |
| 2.2 |   | $H_{pp} = 1.5\text{A/m}$                       | $Q_{LC}$              | 15       | 20           | TBD  | 1             | T     |
| 3   | Min. field for Read Mode (Modulation)                         | Read mode                                      | $H_{pp\ mod}$         | 35       |              |      | A/m           | T     |
| 4   | Min. field for Write Mode                                     | Write mode                                     | $H_{pp\ prog}$        | 58       |              |      | A/m           | T     |
| 5   | Maximum field strength  | $-40^{\circ}\text{C}$ to $+85^{\circ}\text{C}$ | $H_{pp\ max}$         |          |              | 500  | A/m           | Q     |
| 6   | Data retention time EEPROM                                    | $T_{amb} = 25^{\circ}\text{C}$                 | $H_{pp\ mod}$         | 20       |              |      | Year          | Q     |
| 7   | Write endurance EEPROM  | $T_{amb} = 25^{\circ}\text{C}$                 | $H_{pp\ prog}$        | 100k     |              |      | Cycle         | Q     |
| 8   | Accuracy of internal timing references (SRC, FRC oscillators) | $-40^{\circ}\text{C}$ to $+85^{\circ}\text{C}$ | $H_{pp\ max}$         | $\pm 10$ |              |      | %             | Q     |
| 9   | Clock cycle   | $1 / f_{AFE}$                                  | $T_{AFE}$             |          | 8            |      | $\mu\text{s}$ |       |
| 10  | Field clock cycle   |  | $CLK_{FC}$            |          | 125          |      | kHz           |       |
| 11  | Transponder data rate   | Read mode                                      |                       |          | 3.9          |      | Kbit/s        |       |
| 12  | Transponder data rate   | BPLM – Write mode<br>QPLM – Write mode         |                       |          | 4.46<br>5.43 |      | Kbit/s        |       |
| 13  | Startup time  | $3.2V_P / 125\text{kHz}$                       |                       |          | 0.35         | 0.8  | ms            | A     |
| 14  | Transponder charge initial time                               |  | $t_{Tpininit}$        |          | 128          |      | $T_{AFE}$     | T     |
| 15  | Transponder mode voltage check interval                       |  | $t_{Charge}$          |          | 128          |      | $T_{AFE}$     | T     |
| 16  | Bit half period (damped – Talk Back mode)                     |  |                       |          | 16           |      | $T_{AFE}$     |       |
| 17  | Bit half period (undamped – Talk Back mode)                   |  |                       |          | 16           |      | $T_{AFE}$     |       |
| 18  | Bit period (Talk Back mode)                                   |  |                       |          | 32           |      | $T_{AFE}$     |       |
| 19  | Gap period (write mode)                                       | Gap time<br>Start gap time                     | $t_{GAP}$<br>$t_{SG}$ | 10       |              | 20   | $T_{AFE}$     | D     |
| 20  | Bit period (binary “0” – Write mode)                          | BPLM – Mode                                    |                       |          | 24           |      | $T_{AFE}$     |       |
| 21  | Bit period (binary “1” – Write mode)                          | BPLM – Mode                                    |                       |          | 32           |      | $T_{AFE}$     |       |
| 22  | Two bits period (binary “00” – Write mode)                    | QPLM – Mode                                    |                       |          | 28           |      | $T_{AFE}$     |       |
| 23  | Two bits period (binary “01” – Write mode)                    | QPLM – Mode                                    |                       |          | 40           |      | $T_{AFE}$     |       |
| 24  | Receive to transmit time                                      |  | $t_{RXData\_max}$     | 3.6      |              | 4.5  | ms            | Q     |
| 25  | Receive to receive time                                       |  | $t_{TXData\_max}$     | 280      |              | 350  | $\mu\text{s}$ | Q     |

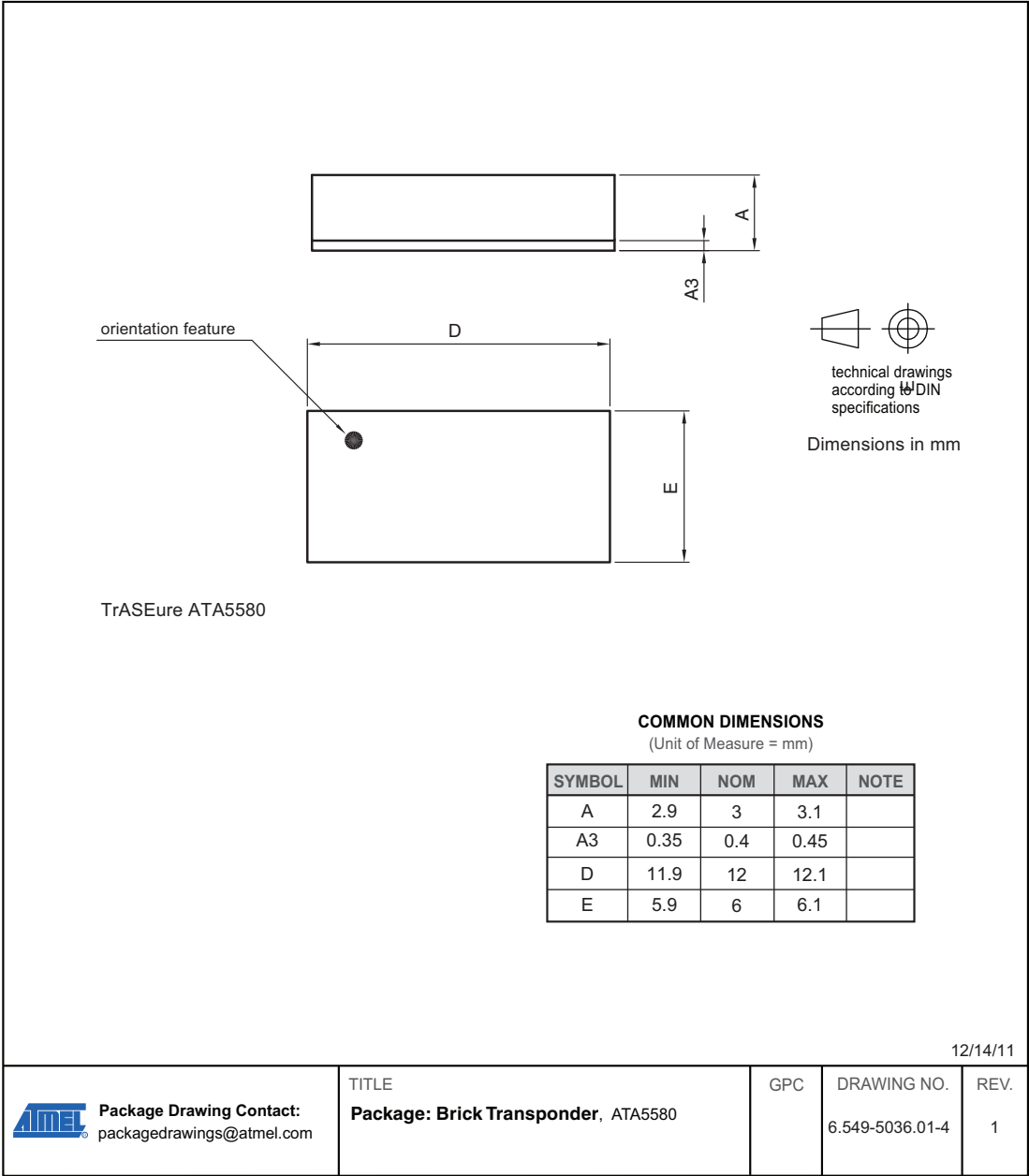
\*) Type means: A = 100% tested, B = 100% correlation tested, C = Characterized on samples, D = Design parameter, T: directly or indirectly tested during production; Q: guaranteed based on initial product qualification data

7. Ordering Information

Table 7-1. ATA5580 Ordering Information

| ATA5580M | nnn        | -TSMW | Package           | Remarks   |
|----------|------------|-------|-------------------|---|
|          | 132        |       | Brick tag package | UA, BPLM, Manchester, RF/32, 32b Ch, 32b Rs                             |
|          | 156        |       | Brick tag package | UA, BPLM, Manchester, RF/32, 100b Ch, 56b Rs                            |
|          | 264        |       | Brick tag package | BA, BPLM, Manchester, RF/32, 64b Ch, 64b Rs                             |
|          | 256        |       | Brick tag package | BA, BPLM, Manchester, RF/32, 100b Ch, 56b Rs                            |
|          | 300 to 999 |       | Brick tag package | Customer defined product. Must submit complete configuration memory map |

8. Package Information



## 9. Revision History

Please note that the following page numbers referred to in this section refer to the specific revision mentioned, not to this document.

| Revision No.     | History  |
|------------------|--|
| 9254B-RFID-05/12 | <ul style="list-style-type: none"><li>• Section 2.7 “LF Physical Layer” on pages 11 to 13 changed</li><li>• Section 6 “Operating Characteristics” on page 36 changed</li></ul> |



Enabling Unlimited Possibilities™

**Atmel Corporation**

2325 Orchard Parkway  
San Jose, CA 95131  
USA

**Tel:** (+1) (408) 441-0311

**Fax:** (+1) (408) 487-2600

[www.atmel.com](http://www.atmel.com)

**Atmel Asia Limited**

Unit 01-5 & 16, 19F  
BEA Tower, Millennium City 5  
418 Kwun Tong Roa  
Kwun Tong, Kowloon  
HONG KONG

**Tel:** (+852) 2245-6100

**Fax:** (+852) 2722-1369

**Atmel Munich GmbH**

Business Campus  
Parking 4  
D-85748 Garching b. Munich  
GERMANY

**Tel:** (+49) 89-31970-0

**Fax:** (+49) 89-3194621

**Atmel Japan G.K.**

16F Shin-Osaki Kangyo Building  
1-6-4 Osaki  
Shinagawa-ku, Tokyo 141-0032  
JAPAN

**Tel:** (+81) (3) 6417-0300

**Fax:** (+81) (3) 6417-0370

© 2012 Atmel Corporation. All rights reserved. / Rev.: 9254B–RFID–05/12

Atmel®, Atmel logo and combinations thereof, AVR®, Enabling Unlimited Possibilities®, and others are registered trademarks or trademarks of Atmel Corporation or its subsidiaries. Other terms and product names may be trademarks of others.

Disclaimer: The information in this document is provided in connection with Atmel products. No license, express or implied, by estoppel or otherwise, to any intellectual property right is granted by this document or in connection with the sale of Atmel products. EXCEPT AS SET FORTH IN THE ATMEL TERMS AND CONDITIONS OF SALES LOCATED ON THE ATMEL WEBSITE, ATMEL ASSUMES NO LIABILITY WHATSOEVER AND DISCLAIMS ANY EXPRESS, IMPLIED OR STATUTORY WARRANTY RELATING TO ITS PRODUCTS INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTY OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE, OR NON-INFRINGEMENT. IN NO EVENT SHALL ATMEL BE LIABLE FOR ANY DIRECT, INDIRECT, CONSEQUENTIAL, PUNITIVE, SPECIAL OR INCIDENTAL DAMAGES (INCLUDING, WITHOUT LIMITATION, DAMAGES FOR LOSS AND PROFITS, BUSINESS INTERRUPTION, OR LOSS OF INFORMATION) ARISING OUT OF THE USE OR INABILITY TO USE THIS DOCUMENT, EVEN IF ATMEL HAS BEEN ADVISED OF THE POSSIBILITY OF SUCH DAMAGES. Atmel makes no representations or warranties with respect to the accuracy or completeness of the contents of this document and reserves the right to make changes to specifications and products descriptions at any time without notice. Atmel does not make any commitment to update the information contained herein. Unless specifically provided otherwise, Atmel products are not suitable for, and shall not be used in, automotive applications. Atmel products are not intended, authorized, or warranted for use as components in applications intended to support or sustain life.