APPLICATION NOTE

Open Source Immobilizer Protocol Stack

ATAN0088

Introduction

Most of the immobilizer systems have historically been based on proprietary protocols and cryptography de facto limiting the interoperability of different vendor ICs.

Additionally, security was often misunderstood with secrecy leading to very limited peer reviewing and therefore inherently weak systems quickly cracked unfortunately after deployment.

The security industry has established peer reviewing as one of the conditions for reaching higher quality standards. AES (Advanced Encryption Standard) is an open source cryptography cipher that went through a thorough review by the industry before been approved by the US government for encrypting classified documents.

Since the communication protocol stacking on top of the cryptographic layer is a key contributor to the overall system security, it obviously needs to go through similar peer reviewing process.

Therefore, Atmel® has elected to release its Immobilizer Protocol under an Open Source License allowing the whole industry to review but also to contribute within a clear legal framework.

Additionally, system designers are allowed, under the terms of the license, to create derivative work to fit specific car manufacturer needs.

Specifically, this Protocol aims at the following goals:

- Security

Based on AES-128, the protocol layers have been designed following recommendations from independent automotive security experts.

- Compatibility and Interoperability

The physical layer is the well established 125kHz Full-Duplex used by all mainstream immobilizer ICs in the market today.

- Configurability

Most of the protocol parameters (challenge length, bit rate, modulation, authentication scheme, etc.) are configurable allowing the best compromise between authentication speed and power consumption for a given system.

Finally, this Immobilizer Protocol is implemented in all Atmel immobilizer devices.

- ATA5580: Standalone transponder
- ATA5795: Remote keyless entry microcontroller with RF transmitter and immobilizer function
- ATA5790: Passive entry / go microcontroller with 3D LF receiver and immobilizer function
- ATA5790N: Passive entry / go microcontroller with 3D LF receiver and immobilizer function
- ATA5791: Passive entry / go microcontroller with 3D LF receiver and immobilizer function
- ATA5291: Antenna Driver for Multiple Antennas

1. Licensing

THE PROTOCOL IS PROVIDED UNDER THE TERMS OF THIS LICENSE (“AGREEMENT”). ANY USE, REPRODUCTION OR DISTRIBUTION OF THE PROTOCOL CONSTITUTES RECIPIENT'S ACCEPTANCE OF THIS AGREEMENT.

1.1 Definitions

“Contributor” means any person or entity that distributes the protocol.

“Contribution” means:

i) any change to the protocol, and

ii) any addition to the protocol;

where such change and/or addition to the protocol originate from and are distributed by a particular contributor. A contribution 'originates' from a contributor if it was added to the protocol by such contributor or anyone acting on such contributor's behalf. Contributions includes additions to the protocol which:

(i) are separate from the protocol;

(ii) distributed in conjunction with the protocol, under their own license agreement (which the present protocol supersedes) or not; and

(iii) are necessary to use the protocol or the contribution to which they relate.

“(To) Contribute” means making a contribution to the protocol.

“Derivate Product” means a product, including but not limited to a vehicle immobilizer system, implementing a function which performance is based wholly or partly on the Protocol, or part of the Protocol, whether this product is designed or intended for commercial purpose, development purpose, or any other purpose.

“Protocol” means this document and any and all subsequent contribution made or distributed in accordance with this agreement.

“Recipient” means anyone who receives the protocol under this agreement, including all contributors.

1.2 Grant of Rights

a. Subject to the terms of this agreement, each contributor hereby grants recipient a non-exclusive, worldwide, royalty-free copyright license to reproduce, contribute, disclose under a confidentiality agreement, distribute and sublicense the contribution of such contributor, if any.

b. Recipient understands that although each contributor grants the licenses to its contributions set forth herein, no assurances are provided by any contributor that the protocol does not infringe the patent or other intellectual property rights of any other entity. Each contributor disclaims any liability to recipient for claims brought by any other entity based on infringement of intellectual property rights or otherwise. As a condition to exercising the rights and licenses granted hereunder, each recipient hereby assumes sole responsibility to secure any other intellectual property rights needed, if any. For example, if a third party patent license is required to allow recipient to distribute the protocol, it is recipient's responsibility to acquire that license before distributing the protocol.

c. Each contributor represents that to its knowledge it has sufficient copyright rights in its contribution, if any, to grant the copyright license set forth in this agreement.

1.3 Requirements

Each contributor must inform Atmel® before distributing the protocol to a recipient at the latest at the moment of such distribution.

A contributor may choose to distribute the protocol under its own license agreement, provided that:

a. the Contributor informs Atmel about such distribution;

b. it complies with the terms and conditions of this agreement; and

c. its license agreement:

   1. effectively disclaims on behalf of all contributors all warranties and conditions, express and implied, including warranties or conditions of title and non-infringement, and implied warranties or conditions of merchantability and fitness for a particular purpose;
   2. effectively excludes on behalf of all contributors all liability for damages, including direct, indirect, special, incidental and consequential damages, such as lost profits;
   3. states that any provisions which differ from this agreement are offered by that contributor alone and not by any other party.

Each contributor must identify itself as the originator of its contribution, if any, in a manner that reasonably allows subsequent Recipients to identify the originator of the contribution.

Contributors may not remove or alter any copyright notices contained within the protocol.

1.4 Commercial Distribution

Commercial distributors of derivate products may accept certain responsibilities with respect to end users, business partners and the like. The contributor who uses the Protocol in a derivate product offering should do so in a manner which does not create potential liability for other contributors. Therefore, if a contributor offers or sells derivates product, such contributor (“Commercial Contributor”) hereby agrees to defend and indemnify every other contributor (“Indemnified Contributor”) against any losses, damages and costs (collectively “Losses”) arising from claims, lawsuits and other legal actions brought by a third party against the Indemnified contributor to the extent caused by the acts or omissions of such commercial contributor in connection with its distribution of derivates products. In order to qualify, an indemnified contributor must: a) promptly notify the commercial contributor in writing of such claim, and b) allow the commercial contributor to control, and cooperate with the commercial contributor in, the defense and any related settlement negotiations. The indemnified contributor may participate in any such claim at its own expense.

For example, a contributor uses the protocol in derivates product X. That contributor is then a commercial contributor. If that commercial contributor then makes performance claims, or offers warranties related to derivates product X, those performance claims and warranties are such commercial contributor's responsibility alone. Under this section, the commercial contributor would have to defend claims against the other contributors related to those performance claims and warranties, and if a court requires any other contributor to pay any damages as a result, the commercial contributor must pay those damages.

1.5 Warranty Terms of Coverage

1.5.1 Atmel Protocol Implementation

Atmel stands behind the implementation of the protocol when it is included as a standard feature in any applicable Atmel device. In this case, standard warranty terms terms apply.

1.5.2 Recipient Protocol Implementation

When the recipient implements the protocol in any device, the recipient voids Atmel warranty.

EXCEPT AS EXPRESSLY SET FORTH IN THIS AGREEMENT, THE PROTOCOL IS PROVIDED ON AN “AS IS” BASIS, WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, EITHER EXPRESS OR IMPLIED INCLUDING, WITHOUT LIMITATION, ANY WARRANTIES OR CONDITIONS OF TITLE, NON-INFRINGEMENT, MERCHANTABILITY OR FITNESS FOR A PARTICULAR PURPOSE. Each recipient is solely responsible for determining the appropriateness of using and distributing the protocol and assumes all risks associated with its exercise of rights under this agreement, including but not limited to the risks and costs of protocol errors, compliance with applicable laws, damage to or loss of data, protocols or equipment, and unavailability or interruption of operations.

1.6 Disclaimer of Liability

EXCEPT AS EXPRESSLY SET FORTH IN THIS AGREEMENT, NEITHER RECIPIENT NOR ANY CONTRIBUTORS SHALL HAVE ANY LIABILITY FOR ANY DIRECT, INDIRECT, INCIDENTAL, SPECIAL, EXEMPLARY, OR CONSEQUENTIAL DAMAGES (INCLUDING WITHOUT LIMITATION LOST PROFITS), HOWEVER CAUSED AND ON ANY THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT LIABILITY, OR TORT (INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY OUT OF THE USE OR DISTRIBUTION OF THE PROTOCOL OR THE EXERCISE OF ANY RIGHTS GRANTED HEREUNDER, EVEN IF ADVISED OF THE POSSIBILITY OF SUCH DAMAGES.

IN NO EVENT SHALL ATMEL BE LIABLE TO ANY CONTRIBUTOR OR RECIPIENT OR ANY THIRD PARTY FOR ANY CONSEQUENTIAL, SPECIAL, INDIRECT OR INCIDENTAL DAMAGES ARISING OUT OF THE USE OF THE MATERIALS EVEN IF ATMEL HAS BEEN ADVISED OF THE POSSIBILITY OF SUCH DAMAGES. IN NO EVENT SHALL THE LIABILITY OF ATMEL RELATING TO THIS AGREEMENT EXCEED THE PRICE PAID TO ATMEL HEREUNDER.

1.7 General

If any provision of this agreement is invalid or unenforceable under applicable law, it shall not affect the validity or enforceability of the remainder of the terms of this agreement, and without further action by the parties hereto, such provision shall be reformed to the minimum extent necessary to make such provision valid and enforceable.

If recipient institutes patent litigation against a contributor with respect to a patent applicable to software (including a cross-claim or counterclaim in a lawsuit), then any patent licenses granted by that contributor to such recipient under this agreement shall terminate as of the date such litigation is filed. In addition, if recipient institutes patent litigation against any entity (including a cross-claim or counterclaim in a lawsuit) alleging that the protocol itself (excluding combinations of the protocol with other software or hardware) infringes such recipient's patent(s), then such recipient's rights granted under Section 1.2 “Grant of Rights” on page 3 b) shall terminate as of the date such litigation is filed.

All recipient's rights under this agreement shall terminate if it fails to comply with any of the material terms or conditions of this agreement and does not cure such failure in a reasonable period of time after becoming aware of such noncompliance. If all recipient's rights under this agreement terminate, recipient agrees to cease use and distribution of the protocol as soon as reasonably practicable. However, recipient's obligations under this agreement and any licenses granted by recipient relating to the protocol shall continue and survive.

Everyone is permitted to copy and distribute copies of this agreement, but in order to avoid inconsistency the agreement is copyrighted and may only be modified in the following manner. Atmel® reserves the right to publish new versions (including revisions) of this agreement from time to time. No one other than Atmel has the right to modify this agreement. Atmel may assign the responsibility to publish new versions of this Agreement to a suitable separate entity. Each new version of the agreement will be given a distinguishing version number. The protocol (including contributions) may always be distributed subject to the version of the agreement under which it was received. In addition, after a new version of the agreement is published, contributor may elect to distribute the protocol (including its contributions) under the new version. Except as expressly stated in Section 1.2 “Grant of Rights” on page 3 a) and Section 1.2 “Grant of Rights” on page 3 b) above, recipient receives no rights or licenses to the intellectual property of any contributor under this agreement, whether expressly, by implication, estoppel or otherwise. All rights in the protocol not expressly granted under this agreement are reserved.

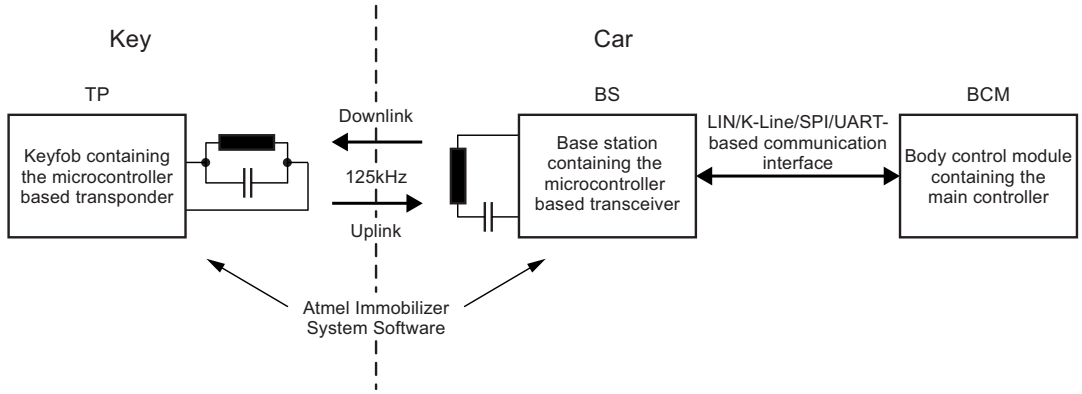
This agreement and all transactions concluded hereunder shall be governed by the laws of the state of California, as such laws are applied to contracts entered into and performed entirely in California by California residents. Any litigation relating to this Agreement shall be subject to the exclusive jurisdiction of the state courts located in Santa Clara County, California, or the federal courts located in the Northern District of California. If any provision of this agreement is held to be invalid, illegal or unenforceable, that provision shall be construed in such a manner that it becomes valid and enforceable and so as to reflect most closely the intent of the parties in agreeing upon the provision in the first place, and the remaining provisions of this agreement shall continue in full force and effect and shall not in any way be affected or impaired by any such determination of invalidity, illegality or unenforceability.

2. System Overview

The immobilizer, as a sub-system of the general car access system, is not used to gain access into the car but to enable the driver to start the engine.

It consists of the key side transponder, the car side base station with transceiver, and a body control module that controls its operation. The picture below visualizes the system partitioning.

Figure 2-1. System Representation



3. Device Support

The Immobilizer Protocol Stack described in this document has been implemented by Atmel® for all its Car Access devices:

- ATA5580: Standalone transponder
- ATA5795: Remote keyless entry microcontroller with RF transmitter and immobilizer function
- ATA5790: Passive entry / go microcontroller with 3D LF receiver and immobilizer function
- ATA5790N: Passive entry / go microcontroller with 3D LF receiver and immobilizer function
- ATA5791: Passive entry / go microcontroller with 3D LF receiver and immobilizer function
- ATA5291: Antenna Driver for Multiple Antennas

Additionally, any 125kHz full duplex (FDX) immobilizer device with load modulation data transfer can be used in conjunction with this protocol to interoperate with any of the devices listed above. All the mainstream immobilizer devices today support this kind of physical layer. If unsure, please contact your Atmel representative for further interoperability investigation.

4. Transponder Features

The intention of this section is to describe the features of the transponder that are specifically used in the development of the immobilizer functionality. It also includes an explanation of the configuration settings that are possible for achieving the required system performance.

4.1 Memory Partitioning

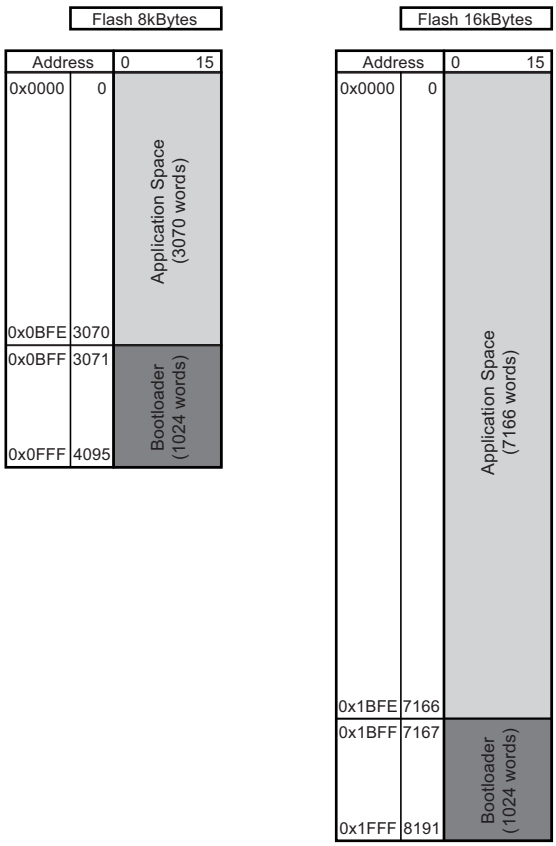
There are two types of memory in the transponder devices that will be used by the immobilizer and the application; Flash and EEPROM. These memories will need to be partitioned and some guidelines established to insure reliable operation. Program code stored in Flash memory typically is used as read only once initial programming has occurred. Non volatile memory that supports multiple read/write access is provided through EEPROM memory structures.

4.1.1 Flash Memory

The immobilizer firmware implementation developed by Atmel® will be stored in the bootloader section of Flash memory. Prior to shipment from Atmel, the bootloader section will be protected from overwriting by use of fuse settings. This allows the application space to be programmed without corrupting the immobilizer firmware.

Atmel devices come with different amounts of Flash memory. However, the bootloader space is consistent across devices at 2Kbytes. Figure 2-1 on page 6 shows how the Flash memory is partitioned for various memory sizes.

Figure 4-1. Flash Memory Partition



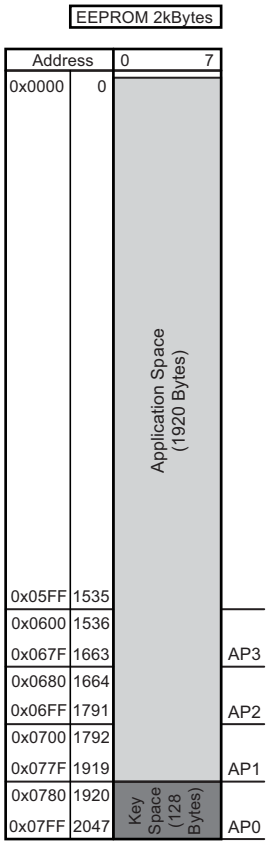
4.1.2 Non Volatile Memory

Non volatile memory used for data storage shall be 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 areas of special access protection (AP0 – AP3). The protection takes the form of requiring an intentional setting of a second register before programming is possible. The AP0 location has been selected for exclusive use by the immobilizer. To prevent corruption, the application code should be audited to insure that this memory is not used.

Figure 4-2 shows an example usage of EEPROM page 1.

Figure 4-2. EEPROM Page 1



Page two shall be permanently locked from overwriting at the end of the Atmel® manufacturing process. This page contains a vast set of configuration and identification features. Once these have been set they are protected from any future changes.

4.1.2.1 Secret Key Storage

The protocol makes provisions for the use of a total of three secret keys. One of these is the fixed Default Secret Key. This 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. Please see Section 6.3.4 “Learn Secret Key1” on page 26 and Section 6.3.5 “Learn Secret Key2” on page 27 for more details.

The other two secret keys are intended for use during normal operation. These will be stored in the AP0 section of EEPROM when the LF interface is used to pair the transponder to the vehicle. The LF interface for transferring secret keys also stores each of these secret keys with two additional copies to insure integrity. When the secret key is accessed for the authentication process, the first key copy is read and checked for consistency. If an error is detected the second copy of the key is read out and checked as well for correctness. If the second copy is also corrupted the third copy is then read out and checked. There is no automatic correction of a corrupted key.

Figure 4-2 on page 8 shows the mapping of the AP0 section located in page 1 of EEPROM.

- The size of the secret key is 16bytes.
- The secret keys for immobilizer shall be stored based on the configuration stored in page 2.
- Both secret key 1 and secret key 2 shall be stored with two copies with the respective locations.

The following table represents the allocation of secret key in the EEPROM memory.

Table 4-1. AP0 Memory Map

	128-bit																
Secret Key	Data1	Data2	Data3	Data4	Data5	Data6	Data7	Data8	Data9	Data10	Data11	Data12	Data13	Data14	Data15	Data16	Physical Address
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
128bytes of Secret Key Memory																	AP0 128 Bytes

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

4.1.2.2 Configuration Memory Options

The protocol provides highly configurable immobilizer features that allow the system design to be optimized. All configuration options must be selected during development, e.g., design testing and validation, and will be placed and locked in page 2 of EEPROM prior to production launch.

1. Data Check Disable

   EEPROM address 0x0815 Bit 0 allows the data CRC 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 the sending or receiving all commands.

Table 4-2. Data Check Disable Register

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

2. 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. This subject is discussed in greater detail in Section 7.1 on page 34.

Table 4-3. Authentication Format Register

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

3. Challenge and Response Length

   These two configuration registers deal with the number of bytes transferred during authentication. The length of the challenge that the transponder expects is stored in EEPROM address 0x0819. In response, the transponder will return an encrypted value with a length determined by the setting in address 0x081A. The start authentication command must have knowledge of these length settings in order to properly perform the authentication process.

Table 4-4. Challenge and Response Length Register

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

4. Uplink Coding and Data Rate

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

   Uplink modulation (MOD): 0 = Manchester, 1 = Bi-phase

   The baud rate setting (0x0817) sets the threshold for the Manchester/bi-phase 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. The T2 prescaler sets the scaled Field clock used to count out the number of steps set by the baud rate setting. Adding up these delays gives a time value. The time values should be set to equal to ½ the bit period of the desired data rate. Recommended values of 16 for the baud rate setting and 0 for the T2 prescaler correspond to a data rate of 3.906kB/s. This can be found using the following relationships.

   a. Base field clock = BFC = 1/(LF field frequency) = 1/125kHz = 8µs
   b. Scaled field clock = SFC = BFC/(2T2PS[2:0] = 1/(2000) = 1/1
   c. Manchester detection period = Half bit period = T = 1/(2 × data rate) = 1/2 × 3906) = 128µs
   d. Baud rate setting = T/(Scaled Field Clock) = 16

Table 4-5. Uplink Coding and Data Rate Registers

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
817	BD7	BD6	BD5	BD4	BD3	BD2	BD1	BD0	Baud rate setting
818	T2D1	T2D0	T2PS2	T2PS1	T2PS0				T2 prescaler

5. Downlink Coding and Data Rate

   EEPROM address 0x0815 Bits 3 and 4 allow the downlink coding type to be selected.

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

   PLM threshold (0x0816) sets the threshold used to decode BPLM data from the vehicle. The value in this register [PLM0:7] 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 could then be set to 24 to achieve accurate decoding, see previous section for relevant equations.

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

Table 4-6. Downlink Coding and Data Rate Register

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

6. 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 first during the start authentication command. This is NOT the location that the key will be stored. Also the type of key transfer process used to load the two secret keys into AP0 is decided through this configuration. The secret keys can be transferred either encrypted or plain text.

Table 4-7. Secret Key Selection Register

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

7. 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. See Section 4.2.5 “Transponder Initialization” on page 15 for more information on how the detection header is used.

Table 4-8. Fob Power Up Register

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

8. Default Secret Key

   A 128bit default secret key is programmed and locked into EEPROM address locations 0x0830 to 0x083F. The process used to generate this value will be defined by the user and will be used during fabrication and assembly. It is the expectation of Atmel® that the default secret key generation process is fixed, but the default secret keys will be unique to each device. This default secret key can not be read out of EEPROM by LF field commands. The default secret key is used for the secure key transfer process.

Table 4-9. Default Secret Key Registers

Byte Address	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	Remarks
830	SK127	SK126	SK125	SK124	SK123	SK122	SK121	SK120	Default secret key
831	SK119	SK118	SK117	SK116	SK115	SK114	SK113	SK112	
832	SK111	SK110	SK109	SK108	SK107	SK106	SK105	SK104	
833	SK103	SK102	SK101	SK100	SK99	SK98	SK97	SK96	
834	SK95	SK94	SK93	SK92	SK91	SK90	SK89	SK88	
835	SK57	SK86	SK85	SK84	SK83	SK82	SK81	SK80	
836	SK79	SK78	SK77	SK76	SK75	SK74	SK73	SK72	
837	SK71	SK70	SK69	SK68	SK67	SK66	SK65	SK64	
838	SK63	SK62	SK61	SK60	SK59	SK58	SK57	SK56	
839	SK55	SK54	SK53	SK52	SK51	SK50	SK49	SK48	
83A	SK47	SK46	SK45	SK44	SK43	SK42	SK41	SK40	
83B	SK39	SK38	SK37	SK36	SK35	SK34	SK33	SK32	
83C	SK31	SK30	SK29	SK28	SK27	SK26	SK25	SK24	
83D	SK23	SK22	SK21	SK20	SK19	SK18	SK17	SK16	
83E	SK15	SK14	SK13	SK12	SK11	SK10	SK9	SK8	
83F	SK7	SK6	SK5	SK4	SK3	SK2	SK1	S0	

9. Transponder Damping Level

   For the uplink, two different damping levels can be selected. The damping level is stored in page 2 of the EEPROM at address 0x081B.

Table 4-10. Transponder Damping Level Register

Byte Address	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	Remarks
81B	0	0	MD1	MD0	0	0	0	0	Transponder module level

4.1.2.3 Charge Pump Selection

This allows the option of selecting between two charge pump blocks. The selection is stored in page 2 of the EEPROM at address 0x081C. Setting this bit to 1 enables the low power charge pump and disables the standard charge pump. We recommend that this bit be set during immobilizer operation.

Table 4-11. Charge Pump Register

Byte Address	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	Remarks
81C	0	EELP	0	0	0	0	0	0	EECR pre-setting

4.1.2.4 Watchdog Timer

This provides the configuration for the watchdog timeout interval. The value stored in page 2 of the EEPROM at address 0x81D is loaded into the watchdog timer and sets the time after which an unserviced watchdog will result in a reset. This sets the prescaler value used to generate the watchdog reset. For exact time values, the datasheet for the device must be consulted but will typically fall between 1ms and 268s.

Table 4-12. Watchdog Timer Register

Byte Address	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	Remarks
81C	0	0	0	0	0	WDP2	WDP1	WDP0	Watchdog timer

4.1.2.5 Voltage Monitoring Default

This configures the voltage monitoring circuit used to protect against EEPROM corruption due to insufficient power supply. The options set in this register are used to set the minimum threshold. EEPROM page 2 address 0x81E contains these default values. The range is typically between 2.0V and 3.4V but the datasheet for the device must be consulted for exact values.

Table 4-13. Voltage Monitoring Register

Byte Address	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	Remarks
81E	BODLS	BODPD	VWPS	0	VMLS3	VMLS2	VMLS1	VMLS0	VMCR default setting

4.1.2.6 Fixed Identification

Fixed identification contains data that has been programmed and locked during the device manufacturing. This data is provided for use in the immobilizer application as well as supply chain management.

1. Unique ID

   The Unique ID or serial number consists of 32 bits of non-sequential, unique values. Each transponder will be assigned a unique value at the end of the manufacturing process and will be stored in EEPROM address locations 0x0800 through 0x0803. This value can be accessed very efficiently through the use of the “Read UID” command. Optionally, the user could append the customer ID stored at address 0x0804 with the unique ID through the use of the read user memory command. This would be useful in determining that this is a genuine part sold to that customer.

Table 4-14. Unique ID Register

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

2. Traceability

   Traceability data contains information that can be used to determine where and how this device has been processed. The following information uniquely identifies this device:

   Address - Value
   0x0808 - Device type
   0x0809 to 0x080B - Lot number
   0x080C - Wafer number
   0x080D to 0x080E - Die number

Table 4-15. Traceability Register

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	

3. Software Revision

   The software revision is contained in EEPROM address 0x080F and provides knowledge of the current version loaded into Flash memory.

Table 4-16. Software Revision Register

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

4. Radio ID

   The Radio ID (RID) is used to identify the RF device uniquely. This consists of 35 bits of information that is unique, non-sequential, and customer specific. This value is stored in EEPROM address 0x0810 to 0x0814.

Table 4-17. Radio ID Registers

Byte Address	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	Remarks
810	RID34	RID33	RID32	RID31	RID30	RID29	RID28	RID27	
811	RID26	RID25	RID24	RID23	RID22	RID21	RID20	RID19	RID number
812	RID18	RID17	RID16	RID15	RID14	RID13	RID12	RID11	
813	RID10	RID9	RID8	RID7	RID6	RID5	RID4	RID3	
814	RID2	RID1	RID0	0	0	0	0	0	

4.2 Device Initialization

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

4.2.1 Power-up Mode

This mode 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 register, ports, and SRAM to initial conditions. The program counter is always set to the reset vector located in the bootloader section. This insures the priority of the immobilizer over all other functions. After a fixed delay, code is executed to check the conditions described as follows.

4.2.2 LF Field Detection

The very first item checked after exiting the Power-Up Mode is the presence of LF field. If present, the immobilizer function will be executed and the other conditional checks will be skipped.

If the LF field is NOT present, then the initialization routine will exit to the application code section if the enhanced mode is not activated. Transponder initialization will not occur.

4.2.3 Enhanced Mode Detection

Once the determination has been made that transponder mode is not needed and that control will be passed to the application there is one last condition to test. This is to determine if the NEXT LF field event will be process using battery power (if available).

If the flag to use “Enhanced Mode” is set, then the power switching will be disabled and the flag cleared to force this to be a one time only event.

Once this is resolved, control will be passed to the application code section. This allows the application to begin with all settings, except possibly the power switching control, in the default reset state.

The return back to the normal application mode could be handled by the application code section or by sending the LF command to leave the Enhanced Mode.

Note: If enhanced mode is activated, the application code MUST make provisions for acknowledging LF field presence and determining how the LF communication will be handled through the use of an interrupt service routine.

4.2.4 Self-Test Mode

For the completely contained transponder several provisions have been made to enable a self test routine. This routine will be activated by the final test system by the IC manufacturer and insures the proper operation of this device in the end application. This test routine is NOT intended to be used in the field and the command to begin this routine should not be used.

4.2.5 Transponder Initialization

Once all conditions have been met to enter transponder mode, the following configurable items (pre-defined by the customer) will be used to prepare for communication.

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

4.2.6 Reliable Communication Channel Indication

If selected, an indication of transponder readiness can be conveyed to the base station. This is achieved through the transmission of a detection header that will insure that the communication channel is open and reliable with a high probability. Both the uplink and downlink paths are verified 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.

5. Base Station Features

The intent of this section is to describe the features of the base station that are specifically used in the development of the immobilizer functionality. This also involves the configurations settings that are possible for achieving the required system performance.

5.1 Memory Partitioning

There are no other functions enabled in the base station firmware other than given in this document. Additional functions can be added but they require the user to recompile the entire source and flash the new project into the device as a whole. Unlike the transponder, there is not a separation of memory structures that would allow flashing of new programs into the device while retaining the existing base station firmware.

The following sections discuss memory usage related to implementation of the immobilizer protocol. Many of the memory structures and addresses described pertain to the specific devices listed in Section 3. “Device Support” on page 6. These devices use hardware specific features to ease the implementation of portions of the protocol. This includes memory segmentation and hardware computational blocks.

While it is possible to implement the protocol in many other devices special care must be taken to insure that sufficient resources exist in the device to contain the object code, configuration settings, and variable storage.

5.1.1 Non Volatile Memory

Non volatile memory for data storage is implemented in EEPROM structures.

5.1.1.1 Configuration Memory Options

The protocol provides highly configurable immobilizer features that enable the user to optimize system design. All configuration options must be selected and placed EEPROM. These configurations are very similar to those that exist in the transponder described in the sections above. However, the addressing is different and there are a few additional configurations that are unique to the base station.

1. Data Check Disable

   DCD (data check disable): EEPROM address 0x0000 Bit 0 allows the data CRC to be disabled for both the request frame and the response frame. If DCD = 0, CRC will be enabled and if DCD = 1, CRC will be disabled. This configuration bit is checked while sending or receiving all commands.
2. Uplink Modulation

   MOD (uplink modulation): EEPROM address 0x0000 Bit 1 indicates the uplink coding type used by transponder. If MOD bit is 0 = Manchester, 1 = Bi-phase.
3. Crypto Mode

   CM (crypto mode): EEPROM address 0x0000 Bit 2 allows the type of authentication protocol to be selected. If CM is 0 = Unilateral, 1 = Bilateral.
4. Downlink Protocol

   DLP0 and DLP1: EEPROM address 0x0000 Bits 3 and 4 allows the downlink coding type to be selected.

   Downlink protocol (DLP1:0): 00 = BPLM, 01 = QPLM (one of four coding), 10 = DPS
5. Secret Key Storage and Transfer

   KS and SKT (secret key selection and transfer): EEPROM address 0x0000 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
6. Detection Header

   TDH: EEPROM address 0x0000 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.

7. Challenge and Response Length

   CHALLENGE_LEN and RESP_LEN: These two configuration registers deal with the number of bits transferred during authentication, allowed values are multiples of eight bit. The length of the challenge that the transponder expects is stored in EEPROM address 0x0001. In response, the transponder will return an encrypted value with a length determined by the setting in address 0x0002. The start authentication command must have knowledge of these length settings for use in the authentication protocol.

8. Timing for Writing

   GAP_TIME: The gap time byte (0x000E) holds the information about the duration required to separate the bits in LF communication while sending commands and data to the transponder.

   ZEROPUL_LEN: The zero pulse length byte (0x0010) holds the information about the duration required to transmit bit 0 in LF communication while sending commands and data to the transponder using BPLM.

   ONEPUL_LEN: The one pulse length byte (0x0012) holds the information about duration required to transmit Bit 1 in LF communication while sending commands and data to the transponder using BPLM.

   QPLM_ZEROPUL_LEN: The QPLM zero pulse length byte (0x0014) holds the information about the duration required to transmit the Bits 00.

   QPLM_DELTA: The QPLM delta byte (0x0016) holds the information about delta value required to calculate the times for T_01, T_10 and T_11.

   Note: The base station does not have field clock information. The desired time interval is the decimal value of the applicable parameter register in µs.

9. Timing for Reading

   MANCHES_RANGE1_MIN: The Manchester range minimum byte (0x0006) holds the minimum damped/undamped time value (in µs) of a short pulse from the Manchester bit.

   MANCHES_RANGE1_MAX: The Manchester range maximum byte (0x0008) holds the maximum damped/undamped time value (in µs) of a short pulse from the Manchester bit.

   MANCHES_RANGE2_MIN: The Manchester range minimum byte (0x000A) holds the minimum damped/undamped time value (in µs) of a long pulse width of the Manchester signal.

   MANCHES_RANGE2_MAX: The Manchester range maximum byte (0x000C) holds the maximum damped/undamped time value (in µs) of a long pulse width of the Manchester signal.

Note: The base station does not have field clock information. The desired time interval is the decimal value of the applicable parameter register in µs.

10. LF Communication Timeout

    LF_COM_TIMEOUT: The LF communication time out byte (0x0003) holds the time out value for LF communication. This is the maximum time available for the transponder to respond for any command from the base station. If there is no response from the slave within this time limit, then the master will consider it as an error. Timeout duration in ms corresponds to the decimal value stored in this location. For example, [00000101] denotes a timeout of 5ms.

    Note: The setting for the LF communication Timeout will be extended if the base station has sent the command to learn a secret key to the transponder. This is necessary because writing 16bytes to the EEPROM on the transponder side will take much more time.

11. Detection Header Timeout

    DETECTHEAD_TIMEOUT: The detection header timeout byte (0x0004) holds the value of the detection header timeout (ms). If the value is '00' or 'FF' a default value of 100ms is used. Timeout duration in ms corresponds to the decimal value stored in this location. For example, [00000101] denotes a timeout of 5ms.

12. SPI Timeout

    SPI_COM_TIMEOUT: The SPI communication time out byte (0x0005) holds the time out value for SPI communication. This is the maximum time available to receive the complete SPI frame. If the entire SPI frame is not received within this time limit, then the BS will send the error message to BCM. Timeout duration in ms corresponds to the decimal value stored in this location. For example, [00000101] denotes a timeout of 5ms.

13. Configuration Checksum

    CHECK_SUM: The Check Sum byte (0x0018) holds the EEPROM check sum from address locations 0x0000 through 0x0017, which represents the entire device configuration.

5.2 Device Initialization

This section describes how the base station device handles the initial power up sequence. The base station will be ready to receive commands from the vehicle controller (BCM or ECU) once initialization is complete.

5.2.1 Power-up Mode

This occurs whenever there is a reset event such as Power-On-Reset (POR), external reset, or watchdog reset. The Power-up Mode asserts all registers, ports, and SRAM to their initial conditions. The program counter is always set to the reset vector located at the start of Flash memory. After a fixed delay, code is executed to check the conditions described as follows.

5.2.2 Base Station Initialization

Once the power-up mode has occurred, the following configurable items (pre-defined by the customer) will be used to prepare for communication.

- MCU driver: To initialize the system clock, stack pointer and to control the global interrupt mask
- Timer driver: To initialize the general purpose timer
- Watchdog driver: To initialize and reload the watchdog
- Port Driver: To initialize the general purpose input/output pins
- EEPROM driver: To read the BS configuration data
- SPI driver: To initialize the SPI hardware and read the SPI communication idle time
- LF driver: To initialize the LF driver

5.2.3 LF Field Generation

Upon completion of the power-up mode, the LF field is generated. If a transponder is present when the LF field is generated, then the immobilizer function on the transponder will be activated.

5.2.4 Vehicle Communication Channel

The base station shall provide an SPI communication channel to place itself under the direction of the vehicle controller and enable seamless bi-directional communication with the transponder. The SPI communication frames are exactly identical to the communication frames used over the LF field and listed in the common firmware features section of this document. This means that the base station is in effect a transparent translator from digital to LF domain. The vehicle controller is the SPI master and must provide the clock source as well as initiate all communication events. There are a few additional commands and error modes used between only the base station and the vehicle controller. They are shown below.

5.2.4.1 Local Resource Command

This command allows the vehicle to use the local base station resources like I/O pins and ON/OFF control of LF field.

Table 5-1. BCM to BS (SPI Command)

Cmd + CRC4	DATA	CRC
1011 1110	4-bit local resource identification + 2-bit operation indicator [read (00)/write (11)] + 2-bit data (valid only for write operation)	CRC - 8

Table 5-2. BS to BCM (SPI Response)

Header	DATA	CRC
1111 1110	4-bit status indication [Success(1111)/Failure(0000)] + 4-bit data [1 = 1111 and 0 = 0000, valid only for read operation]	CRC - 8

5.2.4.2 Detection Header Timeout

After the LF field is turned ON, the base station waits for the detection header timeout period (DETECTHEAD_TIMEOUT) to elapse (if enabled TDH = ON). If the detection header is not received within the user defined timeout period, the base station will send the detection header timeout error message (Status byte [3:0] = b1001) to vehicle controller. If the 'DETECTHEAD_TIMEOUT' parameter value is '00' or 'FF', then the default value of 100ms is used.

5.2.4.3 LF Communication Timeout

After the base station sends a command, it waits for a response from the transponder. If the response is not received within the allowed timeout period (see subheading 10 “LF Communication Timeout Period LF_COM_TIMEOUT” under Section 5.1.1.1 “Configuration Memory Options” on page 16), the base station will send the LF communication timeout error message (Status byte [3:0] = b1010) to the vehicle controller.

5.2.4.4 SPI CRC Error

If a CRC error is present in the received SPI message, the base station will send the SPI CRC error message (Status byte [3:0] = b1011) to vehicle controller.

5.2.4.5 SPI Communication Timeout

If the entire SPI data of a message is not received with the configurable SPI communication timeout period (SPI_COM_TIMEOUT), the base station will send the SPI timeout error message (Status byte [3:0] = b1100) to vehicle controller.

6. Common Firmware Features

The intent of this section is to give an overview of the complete immobilizer feature set provided by the protocol stack. 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.

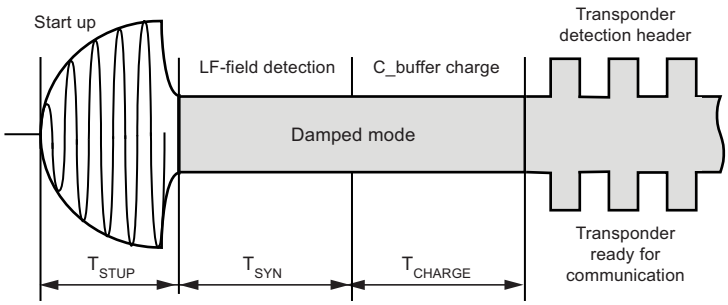
6.1 LF Physical Layer

All communication between the base station and the transponder occurs using the LF field as the signal carrier. If configured to use the detection header, the LF communication link is established when the transponder transmits LF channel detection header which consists of a Manchester coded sequence of “1010…” that is imposed on the 125kHz carrier. This signal continues until the base station interrupts the 125kHz carrier signal, during a damped phase, with a gap.

The 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 6-1 on page 20 depicts a transponder start-up sequence showing the establishment of an LF communication channel.

Figure 6-1. Transponder Startup Sequence



6.1.1 Downlink

A downlink channel is present when data is transmitted from the base station to the transponder. The downlink communication uses Amplitude Modulation (AM) in the form of On-Off-Keying (OOK). 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)
- Quad-pulse length modulation (QPLM): single pulse length is decoded into dual binary logic state (2-bit value). Also known as 1-of-4 encoding.
- 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 “1”s or “0”s.

Figure 6-2. BPLM Downlink

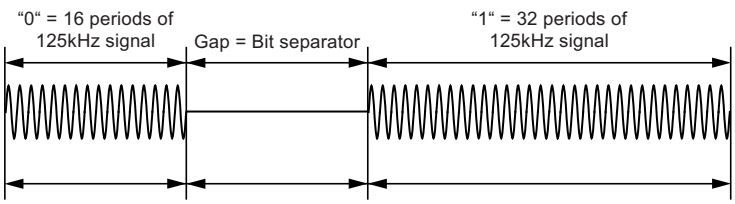


Figure 6-3. QPLM Downlink

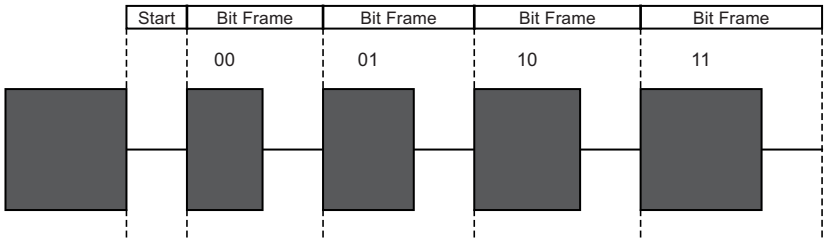
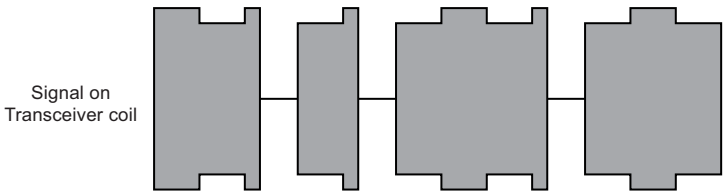


Figure 6-4. DPS Downlink



6.1.2 Uplink

An uplink channel is present when data is transmitted from the transponder to the base station. The uplink communication utilizes AM by reducing the induced voltage in the transponder coil (by loading the 125kHz magnetic field) to 50% or 40% (configurable; see Section 4.1.2.2 “Configuration Memory Options” on page 9 number 9 on page 9) of its un-damped amplitude (50% modulation depth). Binary data is either bi-phase or Manchester encoded.

Figure 6-5. Manchester Uplink

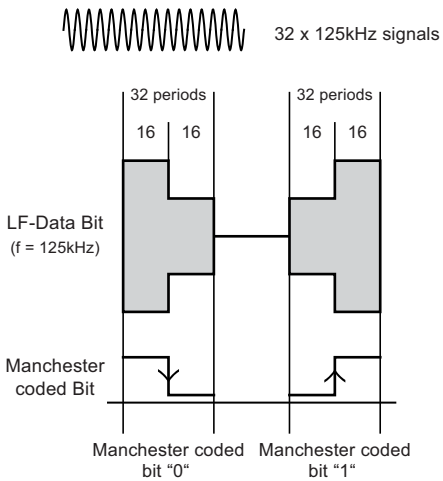
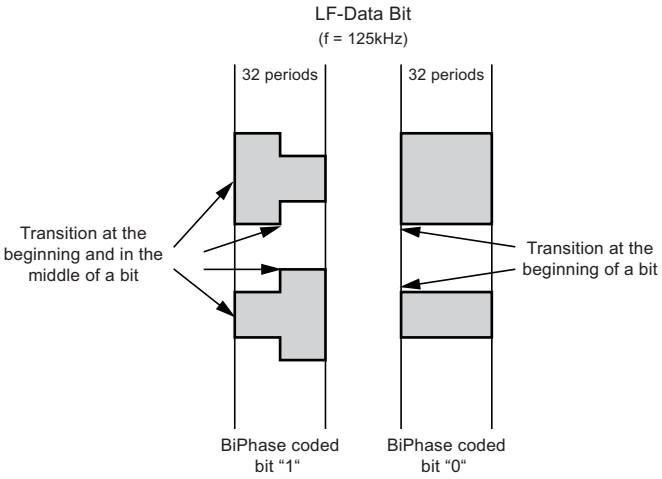


Figure 6-6. Bi-phase Uplink



6.2 LF Communication

The protocol layer 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, at least 2ms delay, and a transponder response. All communication will follow this process and create functionality by executing a series of communication sessions. All commands have a defined response that will be returned from the transponder. The command set only defines a successful response. Any errors that occur will cause the transponder to signal the base station in a unique manner by sending a fixed 1kHz AM modulated signal on the 125kHz carrier. This allows very rapid detection of a problem. The exact cause of the error is stored in the error status byte and can be accessed by a dedicated command.

6.2.1 Request Frame Definition

All transactions are initiated by the base station and conform to the following format:

- 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)

Table 6-1. Request Frame Format

Command Field		Data Field	CRC Field
4-bits	CRC4	Variable	CRC8

6.2.2 Response Frame Definition

All responses from the transponder to the base station conform to the following format:

- Header field = recognizable pattern fixed at 0xFE
- Data field = Variable bit length payload (based on command)
- CRC field = Payload CRC8 (based on presence of payload data)

Table 6-2. Response Frame Format

Header Field	Data Field	CRC Field
8-bits	Variable	CRC8

6.2.3 Error Status Byte

The cause for all errors will be stored in a status byte and contain the error code as well as which command was issued. The status byte shall be initialized to 0xFF during the power-up sequence of the transponder.

The status byte shall be stored in RAM and updated each communication session.

The status byte shall contain the last command received (4-bit) and an error flag (4-bit).

- Status_Byte[7:4]: Four MSBs of the field contain an echo of the command received in the last request frame
- Status_Byte[7:4] = 4b1111; reset occurred (not necessarily due to reset command)
- Status_Byte[3:0]: Four LSBs of the field contain status information encoded as follows:
- Status_Byte[3:0] = 4b0000; success (no error)
- Status_Byte[3:0] = 4b0001; address location locked
- Status_Byte[3:0] = 4b0010; address out of range
- Status_Byte[3:0] = 4b0011; command not supported
- Status_Byte[3:0] = 4b0100; CRC incorrect
- Status_Byte[3:0] = 4b0101; request frame error
- Status_Byte[3:0] = 4b0110; bilateral authentication failed
- Status_Byte[3:0] = 4b0111; AES block error
- Status_Byte[3:0] = 4b1000; generic error
- Status_Byte[3:0] = 4b1111; reset occurred

6.3 LF Command Set

6.3.1 Read UID

The Read UID command provides a very concise method to access the 32-bit unique serial number stored in the transponder. This serial number is assigned during the Atmel® manufacturing process and provides a unique transponder identity for use in the immobilizer system. The request from the base station is concise; consisting of only the 4-bit command and 4-bit CRC. This accelerates response time. In response, the transponder provides its unique 32-bit serial number. The EEPROM address designated for the unique identifier location starts from 0x800 and end with 0x803 (4 bytes).

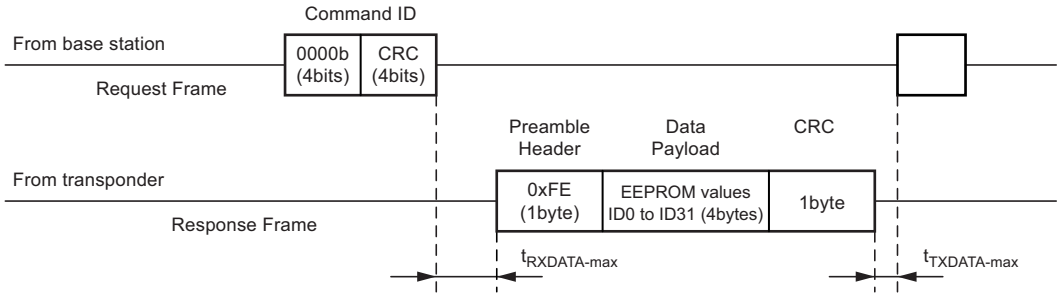
Table 6-3. Read UID (Request Frame)

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

Table 6-4. Read UID (Response Frame)

Field	Size	Values	Description
Preamble header	1byte	0xFE	Synchronization
Data payload	4bytes	EEPROM value	Serial number (ID0 to ID31)
CRC	1byte	Calculate	

Figure 6-7. Read UID Sequence



6.3.2 Read 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 that was executed. This enables the base station to remedy a communication error without complete loss of previously executed functions. The status byte is defined in Section 6.2.3 “Error Status Byte” on page 23.

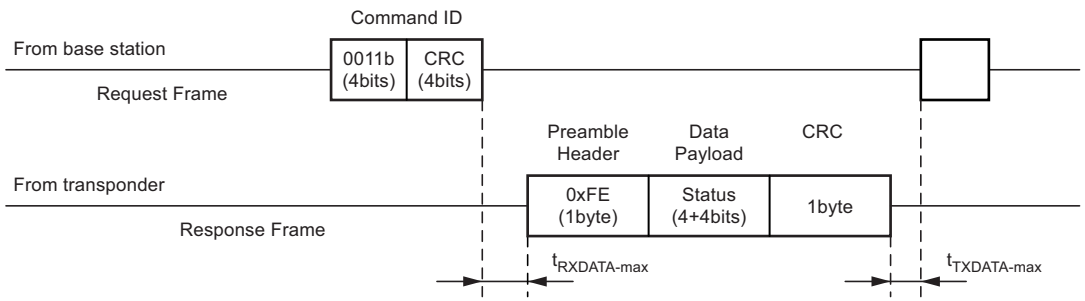
Table 6-5. Transponder Error Status (Request Frame)

Field	Size	Values	Description
Command ID	4 + 4bits	0010b + 0110b CRC	Request status byte
Data payload	N/A		
CRC	N/A		

Table 6-6. Transponder Error Status (Response Frame)

Field	Size	Values	Description
Preamble header	1byte	0xFE	Synchronization
Data payload	1byte	Status	Status
CRC	1byte	Calculate	

Figure 6-8. Transponder Error Status Sequence



6.3.3 Start Authentication

The immobilizer authentication protocol shall be based on challenge – response topology. This can be achieved using unilateral authentication (UA) or bilateral authentication (BA).

The start authentication command causes an authentication protocol to begin. The length of the request payload (challenge length) is dependent on the setting stored in EEPROM page 2 address 0x819 and the response length is dependent on the setting stored in EEPROM page 2 address 0x81A.

The type of protocol that is used will depend on the configuration stored in EEPROM page 2 at address 0x815. 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, bit lengths ciphers of various lengths are used.

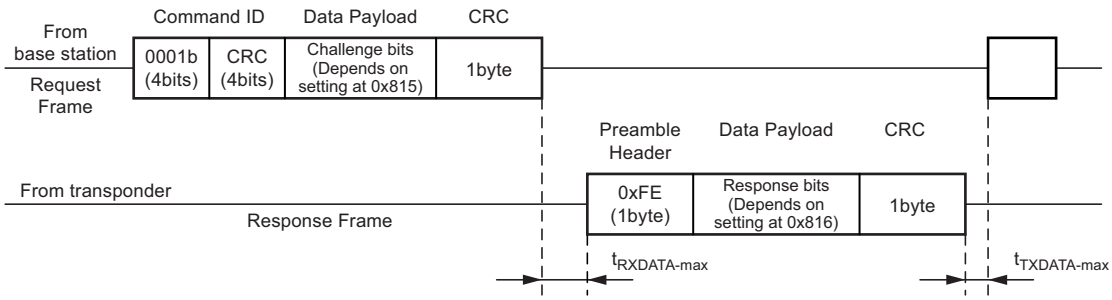
Table 6-7. Start Authentication (Request Frame)

Field	Size	Values	Description
Command ID	4 + 4bits	0001b + 0011b CRC	Start Authentication
Data payload	Varies (recommend 100 or 128bits)	Challenge bits	Depends on EEPROM page 2 setting
CRC	1byte	Calculate	

Table 6-8. Start Authentication (Response Frame)

Field	Size	Values	Description
Preamble header	1byte	0xFE	Synchronization
Data payload	Varies (recommend 56 or 80bits)	Response bits	Depends on EEPROM page 2 setting
CRC	1byte	Calculate	

Figure 6-9. Start Authentication Sequence



6.3.4 Learn Secret Key1

This command starts the Secret Key1 learn process. Either the open or secure transfer process will be followed, depending on the configuration setting stored in EEPROM at address 0x0815 (Bit 6). If this bit (SKT - secure key transfer bit) is 0, the transfer will be open mode and if the bit is 1, the transfer will be secure mode. The request frame contains a 128-bit secret key data payload (may be encrypted during secure transfer). The 128-bit key transferred through this command will be stored in AP0 key position 1 (0x7C0) along with two copies. The response frame consists of a status byte and the data payload. The status byte will be stored in RAM and updated each communication session. The status byte structure is defined in Section 6.2.3 “Error Status Byte” on page 23.

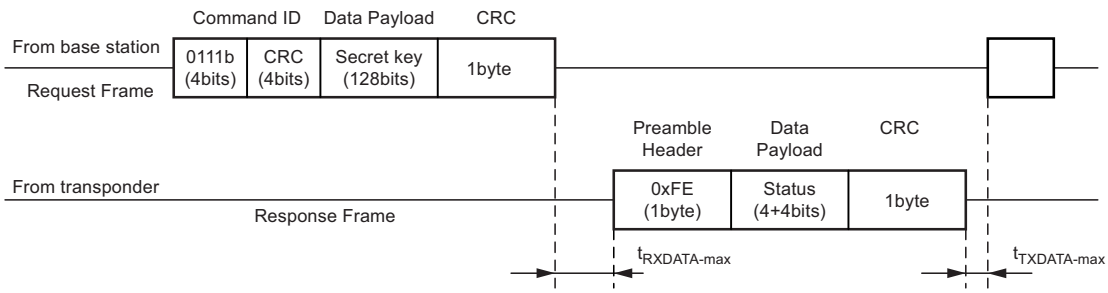
Table 6-9. Learn Secret Key1 (Request Frame)

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

Table 6-10. Learn Secret Key1 (Response Frame)

Field	Size	Values	Description
Preamble header	1byte	0xFE	Synchronization
Data payload	1byte	Status	[7:4] previous command [3:0] encoded error info
CRC	1byte	Calculate	

Figure 6-10. Learn Secret Key1 Sequence



6.3.5 Learn Secret Key2

This command starts the Secret Key2 learn process. Either the open or secure transfer process will be followed, depending on the configuration setting stored in EEPROM at address 0x0815 (Bit 6). If this bit (SKT - Secure Key Transfer bit) is 0, the transfer will be open mode and if the bit is 1, the transfer will be secure mode. The request frame contains a 128-bit secret key data payload (may be encrypted during secure transfer). The 128-bit key transferred through this command will be stored in AP1 key position 2 (0x780) along with two copies. The response frame consists of a status byte and the data payload. The status byte will be stored in RAM and updated each communication session. The status byte structure is defined in Section 6.2.3 “Error Status Byte” on page 23.

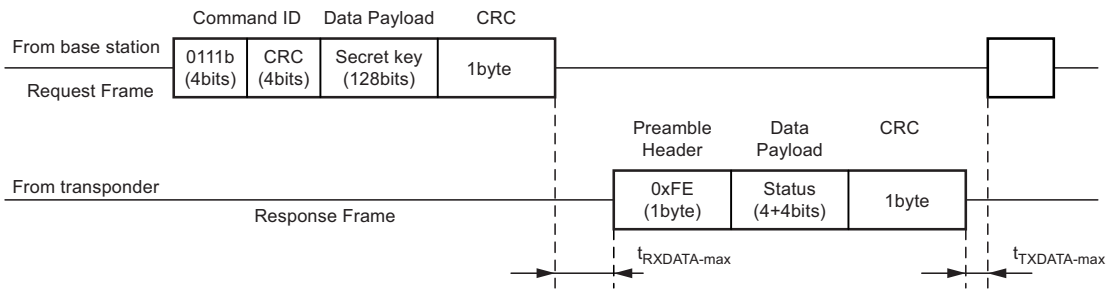
Table 6-11. Learn Secret Key2 (Request Frame)

Field	Size	Values	Description
Command ID	4 + 4bits	1000b + 1010b CRC	Learn Secret Key2
Data payload	128bits		AES (possibly encrypted) secret key
CRC	1byte	Calculate	

Table 6-12. Learn Secret Key2 (Response Frame)

Field	Size	Values	Description
Preamble header	1byte	0xFE	Synchronization
Data payload	1byte	Status	[7:4] previous command [3:0] encoded error info
CRC	1byte	Calculate	

Figure 6-11. Learn Secret Key2 Sequence



6.3.6 Initiate Enhanced Mode

This command initializes the enhanced Mode command structure. This command will begin a sequence to place the transponder into the enhanced mode where the battery supply is used during transponder communication by setting the enhanced mode flag in EEPROM. This flag will be checked at each POR to determine if the power switch should be disabled. Afterwards the flag is automatically cleared and the controller jumps to the application area. The application software could handle the LF communication then by itself or if the LF field is present it could jump back to the immobilizer firmware. To return back to the normal transponder mode the command “Leave Enhanced Mode” must be sent, which will then enable the internal power switch again.

The status byte structure is defined in Section 6.2.3 “Error Status Byte” on page 23.

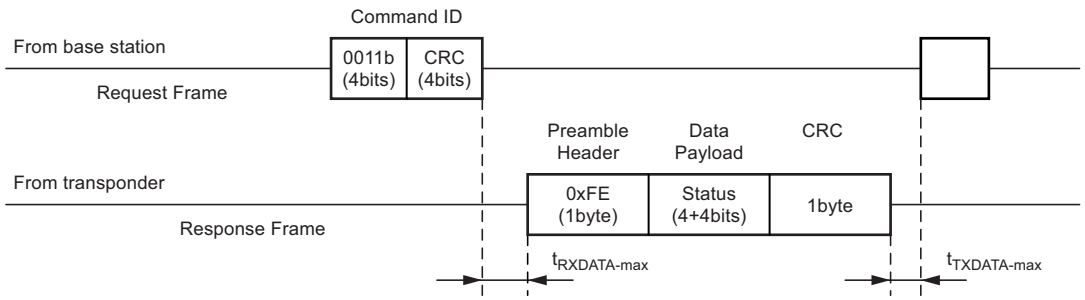
Table 6-13. Initiate Enhanced Mode (Request Frame)

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

Table 6-14. Initiate Enhanced Mode (Response Frame)

Field	Size	Values	Description
Preamble header	1byte	0xFE	Synchronization
Data payload	1byte	Status	[7:4] previous command [3:0] encoded error info
CRC	1byte	Calculate	

Figure 6-12. Initiate Enhanced Mode Sequence



6.3.7 Repeat Last Response

This command requests a repeat of the last transmission. This enables a retry strategy that significantly reduces communication response time.

The response frame matches with the response from the previous command.

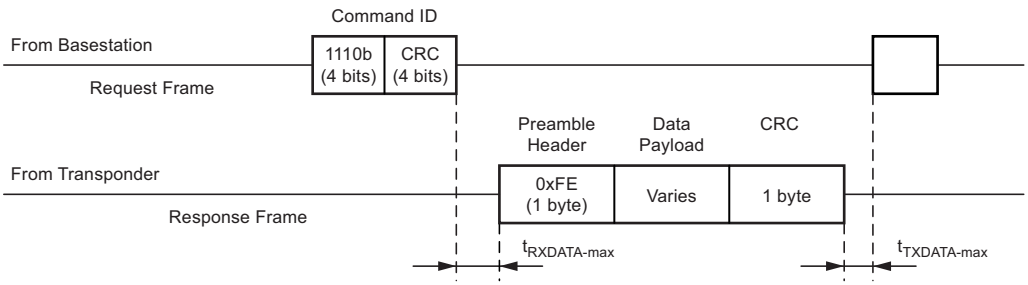
Table 6-15. Repeat Last Response (Request Frame)

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

Table 6-16. Repeat Last Response (Response Frame)

Field	Size	Values	Description
Preamble header	1byte	0xFE	Synchronization
Data payload	Varies	Varies	Previous Response
CRC	1byte	Calculate	

Figure 6-13. Repeat Last Response Sequence



6.3.8 Read User Memory

This command enables a read operation from the user memory (EEPROM). The request frame data block will provide 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 (0x0831 to 0x083F) shall NOT be allowed access by the memory access commands.

The transponder will provide the status byte as well as requested number of EEPROM data bytes in the response frame. The response length specified will not exceed 16 bytes. Data length 0 means 16 bytes. The status byte structure is defined in Section 6.2.3 “Error Status Byte” on page 23.

If bilateral authentication (BA) was selected in Bit 2 (CM) of EEPROM located at address 0x815, the transponder will not return the requested data until a successful authentication routine has occurred.

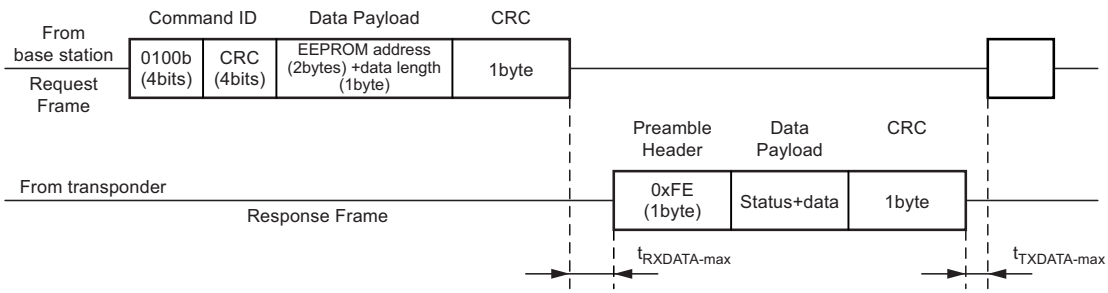
Table 6-17. Read User Memory (Request Frame)

Field	Size	Values	Description
Command ID	4 + 4bits	0100b + 1100b CRC	Read user memory
Data payload	2bytes + 1byte	Address + data (≤$0F)	EEPROM address + data length
CRC	1byte	Calculate	

Table 6-18. Read User Memory (Response Frame)

Field	Size	Values	Description
Preamble header	1byte	0xFE	Synchronization
Data payload	1byte + (≤ 16bytes)	Status + data	Status + EEPROM data
CRC	1byte	Calculate	

Figure 6-14. Read User Memory Sequence



6.3.9 Write User Memory

This command enables a write operation to the user memory (EEPROM). The request frame data block provides the beginning address of the EEPROM followed by the number of bytes and the data to be written. During normal operation the number of EEPROM data bytes to be written shall be at maximum 4bytes. During enhanced mode the number of EEPROM data bytes to be written shall be at maximum 16bytes or 128bits. The EEPROM data will always be sent as complete bytes.

The status byte structure is defined in Section 6.2.3 “Error Status Byte” on page 23.

Write commands that involve transponder EEPROM addresses in the AP1, AP2, and AP3 sections will first check the saved lock state. If the section is locked, the command shall be aborted and transponder will respond with an error message.

If bilateral authentication (BA) was selected in Bit 2 (CM) of EEPROM located at address 0x0815, the transponder will not write the requested data until a successful authentication routine has occurred.

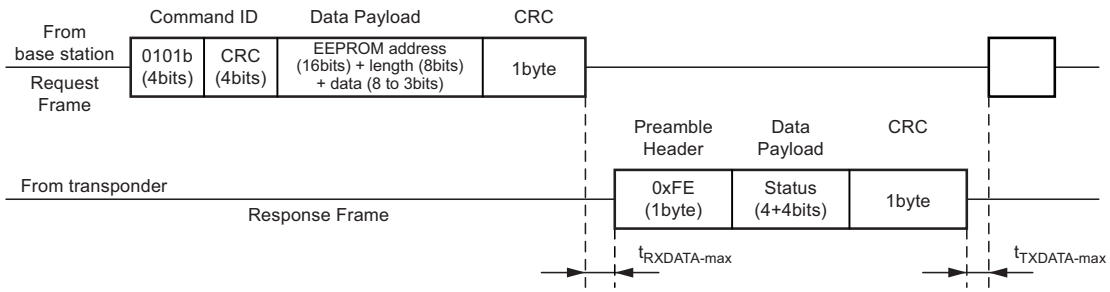
Table 6-19. Write User Memory (Request Frame)

Field	Size	Values	Description
Command ID	4 + 4bits	0101b + 1111b CRC	Write user memory
Data payload in normal mode	2bytes + 1byte + 1byte	Address + data length (≤$04) + data value	EEPROM address + data length + data
Data payload in enhanced mode	2bytes + 1byte + 1byte	Address + data length (≤$0F) + data value	EEPROM address + data length + data
CRC	1byte	Calculate	

Table 6-20. Write User Memory (Response Frame)

Field	Size	Values	Description
Preamble header	1byte	0xFE	Synchronization
Data payload	1byte + (≤16bytes)	Status	Status byte
CRC	1byte	Calculate	

Figure 6-15. Write User Memory Sequence



6.3.10 Write Memory Access Protection

This command will protect sections AP1, AP2, AP3 from being overwritten through transponder memory access commands (LF field commands). Once protection has been applied, it can not be removed.

The request frame data payload shall consist of one byte that conforms to the following format: [7:6] = 00b, [5:4] = 11b locks AP3, [3:2] = 11b locks AP2, [1:0] = 11b locks AP1. To leave individual sections unlocked, substitute 00b in the corresponding portion of the data frame payload. For example, data frame payload = 00110011b locks AP3 and AP1 and leaves section AP2 unlocked. The use of two bits for each memory section reduces the likelihood of inadvertent locking caused by one bit corruption.

The status byte structure is defined in Section 6.2.3 “Error Status Byte” on page 23.

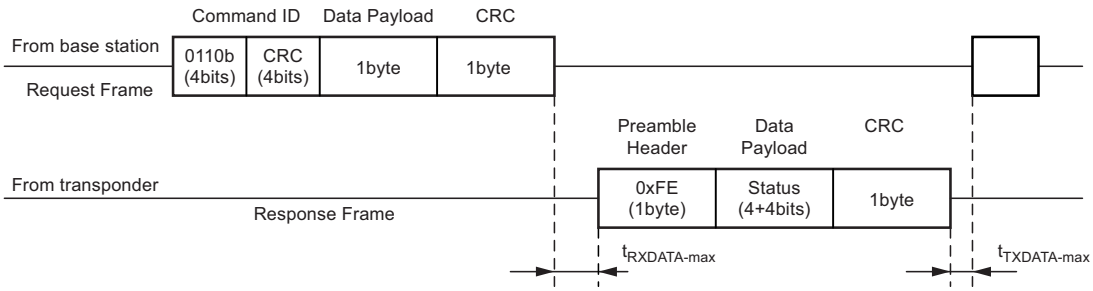
Table 6-21. Write Memory Access Protection (Request Frame)

Field	Size	Values	Description
Command ID	4 + 4bits	0110b + 1010b CRC	Write memory access protection
Data payload	1byte	[7:6]=00b, [5:4]=11b locks AP3, [3:2]=11b locks AP2, [1:0]=11b locks AP1	Protection scheme
CRC	1byte	Calculate	

Table 6-22. Write Memory Access Protection (Response Frame)

Field	Size	Values	Description
Preamble header	1byte	0xFE	Synchronization
Data payload	1byte	Status	Status byte
CRC	1byte	Calculate	

Figure 6-16. Write Memory Access Protection Sequence



6.3.11 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, thus preventing from sending a response.

Table 6-23. Leave Enhanced Mode (Request Frame)

Field	Size	Values	Description
Command ID	4 + 4bits	1010b + 1101b CRC	Leave enhanced mode
Data payload	N/A		
CRC	N/A		

Table 6-24. Leave Enhanced Mode (Response Frame)

Field	Size	Values	Description
Preamble header	1byte	0xFE	Synchronization
Data payload	1byte	Due to the description above, there will be no response.	[7:4] previous command [3:0] encoded error info
CRC	1byte	Calculate	

6.4 Communication Integrity and Error Mitigation

The commands are protected from transmission channel corruption by the use of a CRC nibble. This will prevent accidental processing of an unintended command due to bit corruption. The data can be protected through a second CRC byte. This is true of communication in both uplink and downlink directions. The use of this fast detection of bit level corruption enables the implementation of a very efficient error recovery strategy. When this is combined with the “Repeat Last Response” command, uplink errors can be quickly and automatically mitigated.

The following is suggested as a 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 removing of the LF field

7. Immobilizer Functionality

This describes the steps needed implement the immobilizer system functionality. This functionality would be achieved in the base station and vehicle controller through the use of features and commands that Atmel® has provided. The following sections provide a recommendation of how this can be achieved.

7.1 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.

7.1.1 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.

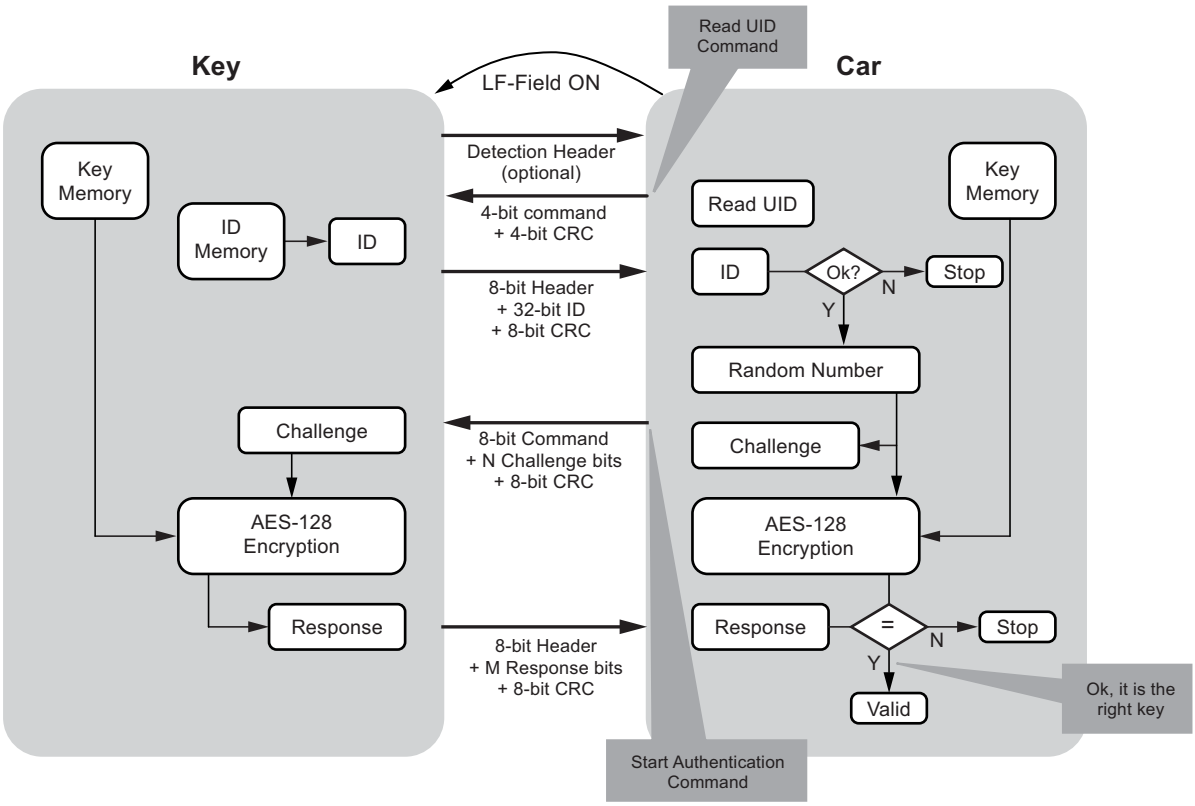
Unilateral authentication should be initiated by the base station and conform to the following sequence.

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 128bits.

A graphical example is shown in Figure 7-1 on page 35.

Figure 7-1. Unilateral Authentication Protocol



7.1.1.1 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.

7.1.1.2 Start Authentication

The encrypted authentication is initiated with the Start Authentication request that provides the challenge data. Atmel® recommends choosing 104 or 128bits for the challenge length. The encrypted response should be chosen as 56 or 80bits 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 192 and 240 respectively. This works out to a bit security level of 52 respectively 64bits 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.

7.1.2 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.

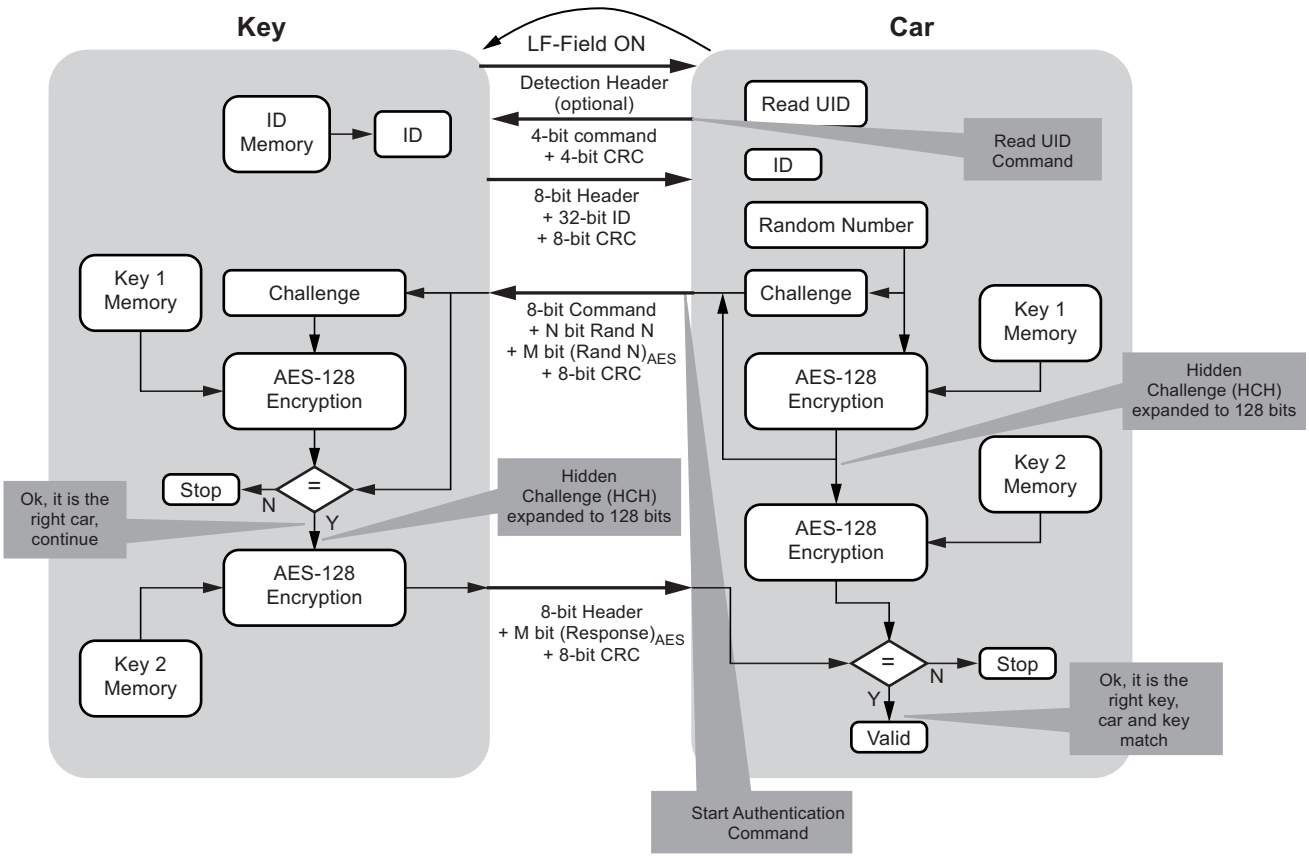
1. The base station shall send the LF command “Read UID”.
2. The transponder shall respond by providing the 32-bit UID in its “Response Frame”.
3. The base station shall send the LF command “Start Authentication” which includes a random number “Challenge” followed by an AES encrypted version of the “Challenge” 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 below.

Figure 7-2. Authentication BA



7.1.3 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.

7.1.4 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 104bits and encrypted challenge of 56bits. 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 56bits.

This response can be evaluated by the vehicle to determine authenticity of the transponder. The total number of bits transferred is 248 and provides a bit security level of 52. 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.

7.1.5 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 56bit “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 56bits could be captured, this leaves 72bits 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.

7.2 Memory Access

General purpose memory is a very important part of an immobilizer system. Included in the Atmel solution is a large section of EEPROM and a very efficient LF commands set for access. The block size for access is flexible and allows the end system designer to build structures that are optimized to fit the data format. The only areas that are not accessible through the memory commands are the AP0 section (used for secret keys) and EEPROM page 2 (used for default secret key storage). All other memory can be accessed. This offers the engineer the option to allow the vehicle to interact with the transponder immobilizer or RKE application. For example, the vehicle could resynchronize with the RKE rolling code counter, read out user specific information, or read/write diagnostic trouble codes.

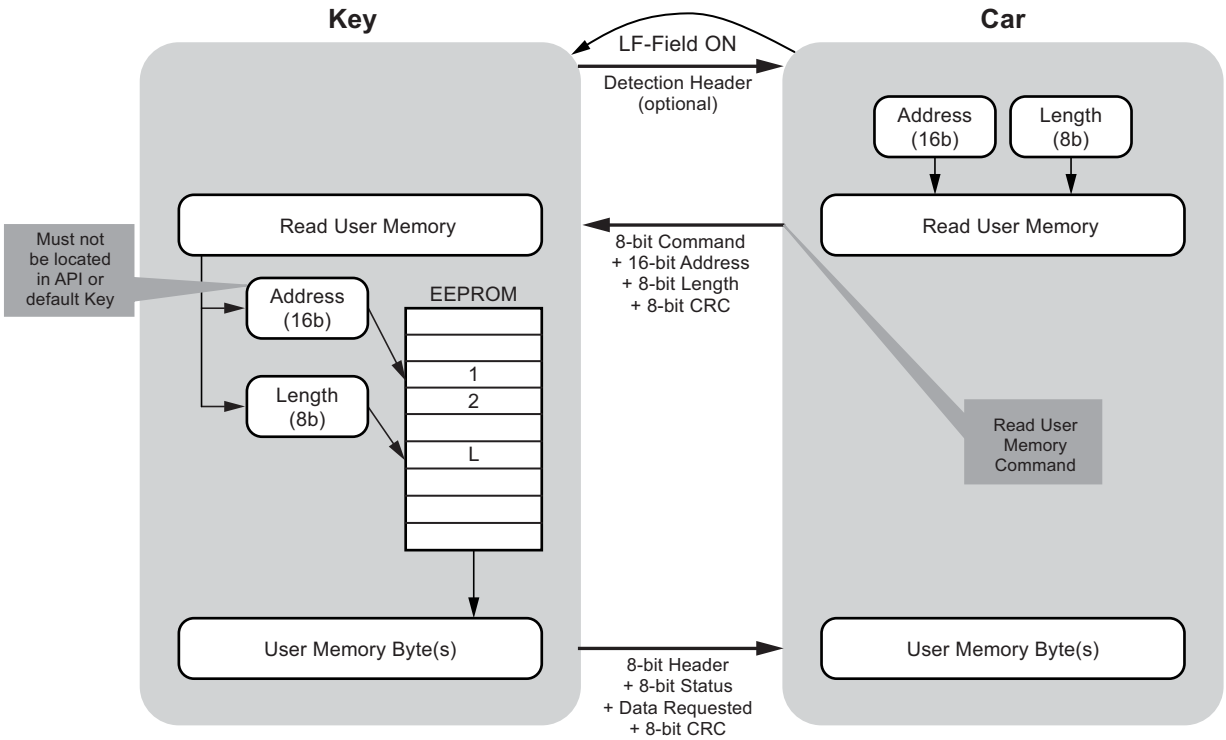
For higher security, if the transponder is configured to use bilateral authentication, an authentication session must be successfully accomplished before any memory access command will be possible!

7.2.1 Read Memory

The reading of user memory only requires that the starting address and the requested number of bytes be provided. This allows block sizes from one to sixteen bytes to be accessed from the transponder's non volatile memory. The memory will be accessed and the data returned starting at the first address and incrementing sequentially until all requested address locations are read.

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

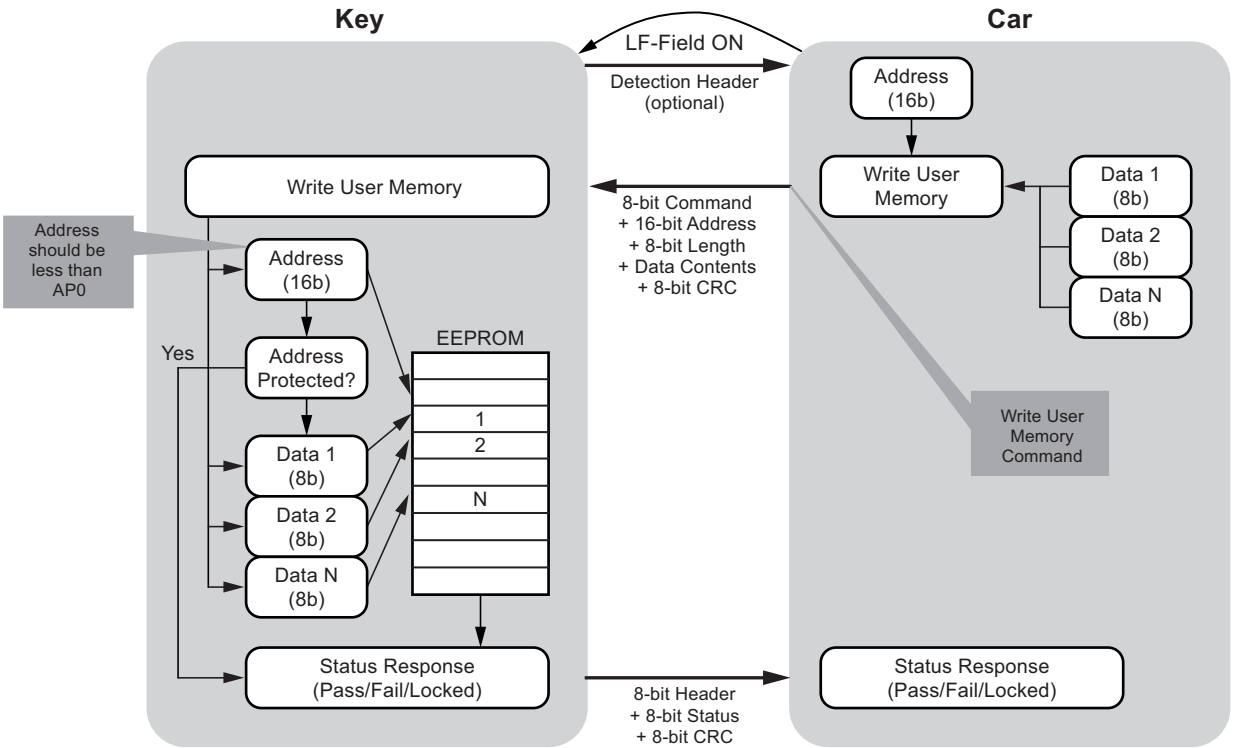
Figure 7-3. Read Memory



7.2.2 Write Memory

Writing data into the memory requires the starting address and the number of data bytes to be stored. The length of this block is limited to four bytes (128bytes in enhanced mode) and must always be sent as complete bytes (partial bytes are not allowed). Before the memory location is written, the firmware will check to see if access protection applies and determine if this request is allowed. Only if these checks are successful does the data get written into EEPROM.

Figure 7-4. Write Memory



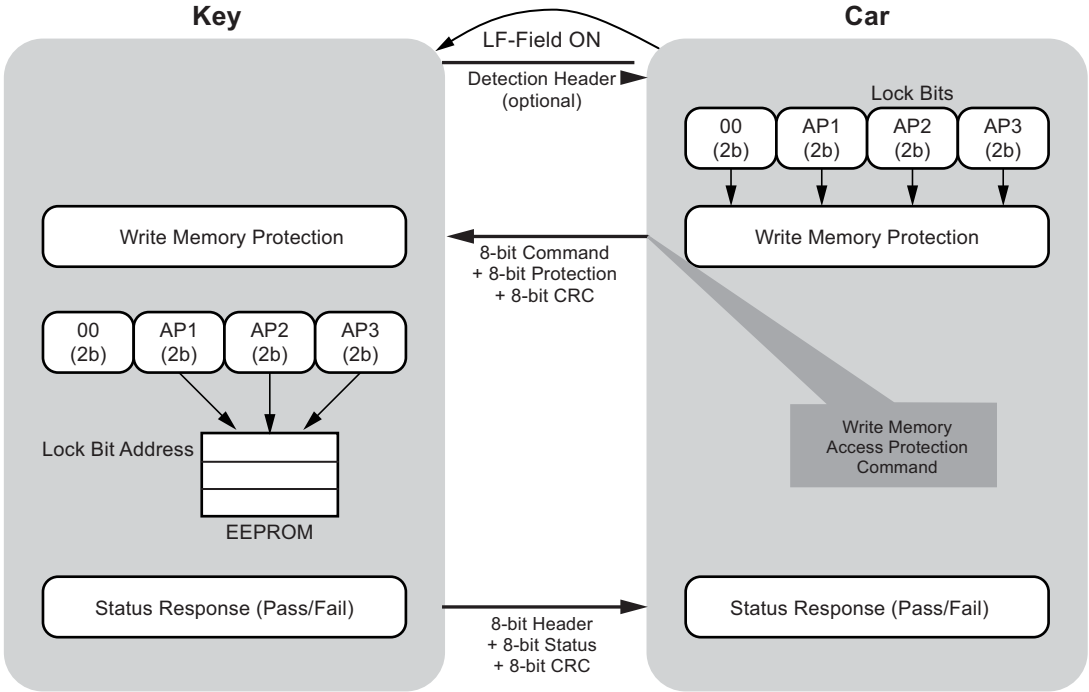
7.2.3 Memory Protection

The memory protection request provides a means to prevent EEPROM data from being modified. Once this protection is applied, it can not be removed by any LF commands. The protection applies to a complete section of EEPROM. There are three such sections that can be selected for special protection. They are defined as the AP1, AP2, and AP3 portions of EEPROM and contain 128bytes in each section. One example of this could be a block of manufacturing process information may be programmed into AP1 and locked so that this can not be modified through LF commands. This would allow a returned device to be traced back through the exact manufacture chain.

The memory protection feature is implemented completely in firmware. The protection causes the transponder to ignore LF commands from the base station that attempt to modify contents of the protected sections in EEPROM.

The write memory access protection command consists of one byte that defines which of sections AP1, AP2, or AP3 will be protected. Two bits are used for each memory section to add extra protection from false locking scenarios. Both bits must be set to logical one for the protection to be applied. All unaffected memory sections should have their corresponding protection bits set to logic zero. In both cases, execution of the Memory Protection command will not change current values in memory.

Figure 7-5. Write Memory Protection



7.2.4 Memory Encryption

The LF Command Set does not support memory encryption. If required the application software should use the hardware encryption block to encrypt (decrypt) the data prior to writing (reading) it to (from) non-volatile memory. For example, a rolling code counter could be encrypted and stored in memory. Each time this is needed, the application software would read it, decrypt it, use the counter, increment it, encrypt the new counter value, and then store this back in memory.

7.3 Identification

One of the primary goals of the immobilizer is to quickly establish and verify the identity of the user. The firmware provided by Atmel® can accommodate many options for identification that allow the system designer to optimize trade-offs between security and speed. The following sections provide examples of how the fixed identification characteristics (native to the Atmel device) can be used to create a custom identification scheme by using the memory access commands and custom block sizes.

7.3.1 Serial Number

The unique device serial number is a fixed value programmed and locked by the IC manufacturer. This value is a 32bit 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 in order to provide a fast preliminary screen for acceptable users.

7.4 Personalization

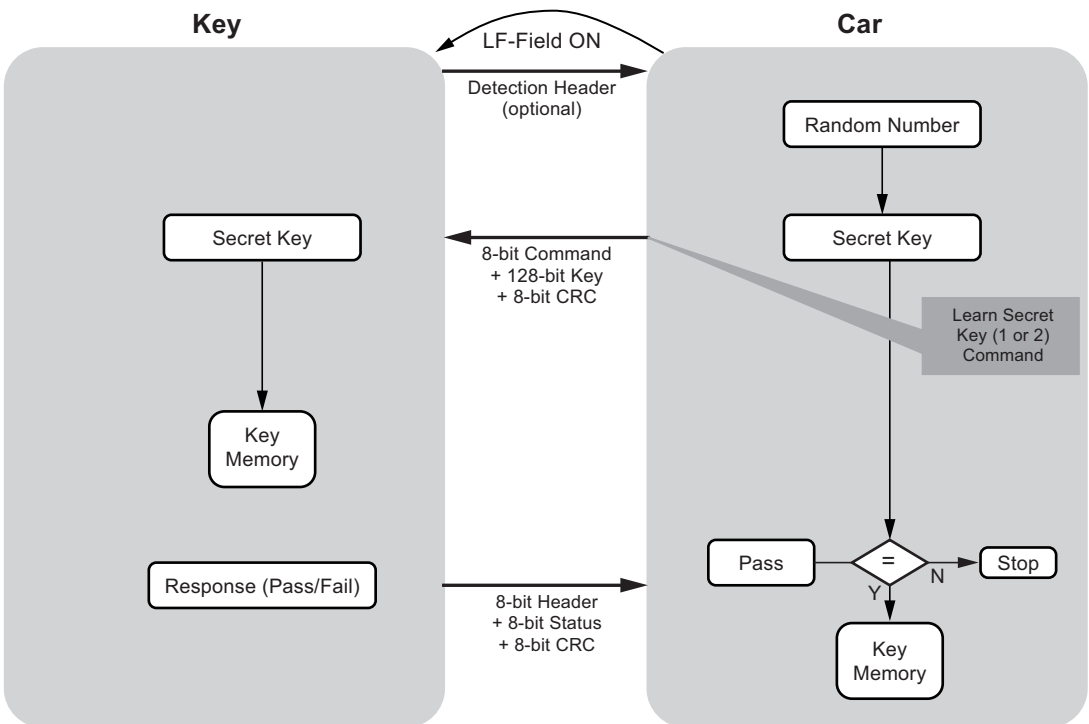
Personalization refers to the process of setting or resetting the initial parameters of the device. In the case of the immobilizer this takes the form of pairing the transponder to the vehicle. The most common pairing method is the transfer of the secret key(s) from the vehicle to the transponder. Other 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.

7.4.1 Open Key Learn

If the security of the key transfer can be insured 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. The following section details how this sequence would occur.

- The base station shall send the 128-bit secret key using the learn secret key command.
- The transponder will store the key in the AP0 section of EEPROM at the corresponding key location (1 or 2 depending on command).

Figure 7-6. Open Key Learn 1/2

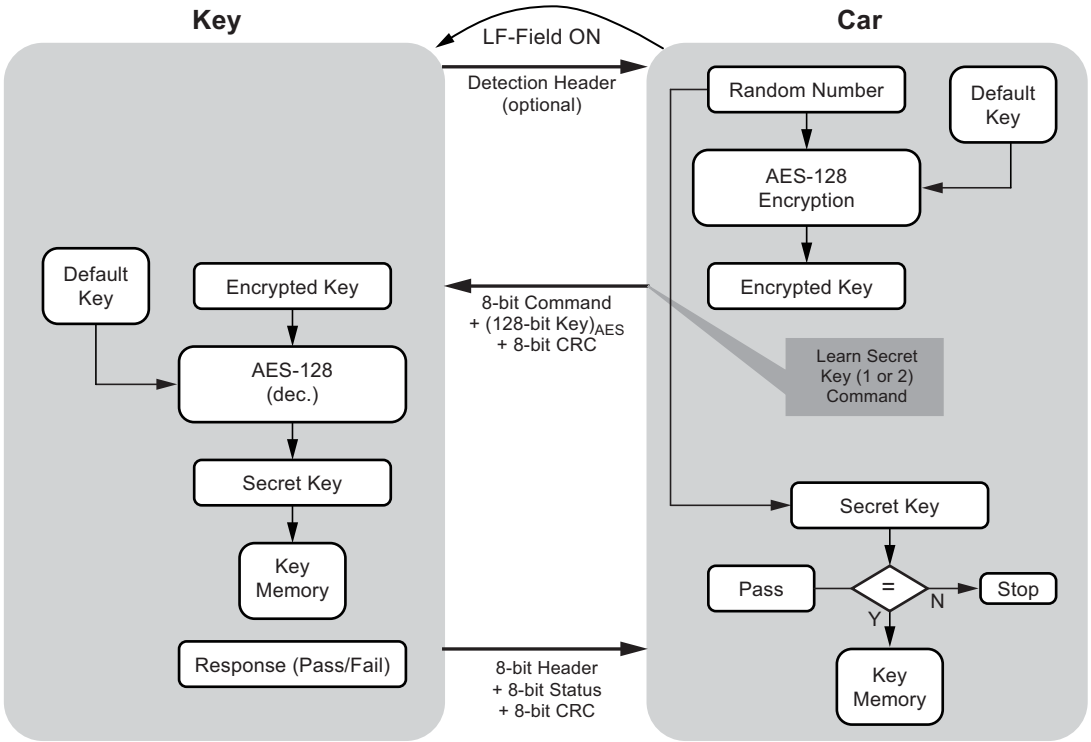


7.4.2 Secure Key Learn

Because the integrity of the authentication process depends on knowledge of the secret keys, it is important to minimize their exposure in order to avoid potential theft. To address this, Atmel® has provided a means to transfer the secret key in an encrypted fashion. This involves the use of the default secret key stored in EEPROM page 2. This protects against eavesdropping by an attacker during key transfer. This allows secure implementation of user initiated personalization in situations where physical security can not be insured. The following section details how this sequence would occur.

- The base station sends the 128-bit secret key that has been encrypted using the default key stored in EEPROM page 2.
- The transponder decrypts this to reveal the secret key.
- The transponder will store the key in the AP0 section of EEPROM in key position 1 or 2.

Figure 7-7. Secure Key Learn



7.5 Enhanced Mode

Enhanced mode provides a means to execute an LF communication session using battery power instead of LF field power. The reason to do this can be anything from writing larger EEPROM blocks at a time, allowing battery charging features, or any other function that runs in the application space (customer provided) that would need an LF communication interface. The process to remain in battery power during an LF event requires very specific handling. There are multiple steps that must take place to insure the proper use this feature.

They are:

- Initiation of enhanced mode
- Disable of power switching
- Ensure that this is only a one-time event
- Acknowledging LF field presence
- Execution of enhanced features

7.5.1 Initiation of Enhanced Mode

To enable enhanced mode, an LF command (init enhanced mode) is sent to the transponder. This sets a flag in EEPROM to be processed after the next reset event. The reason for this is that during normal operation, when the LF command is first sent, the transponder operates on LF field power not battery power. The device does not have knowledge of the state of the battery and can not make the determination to proceed to battery power. However, this determination can be made if the enhanced mode flag is set prior to removal of the LF field.

7.5.2 Disable of Power Switching and Ensuring Single Operation

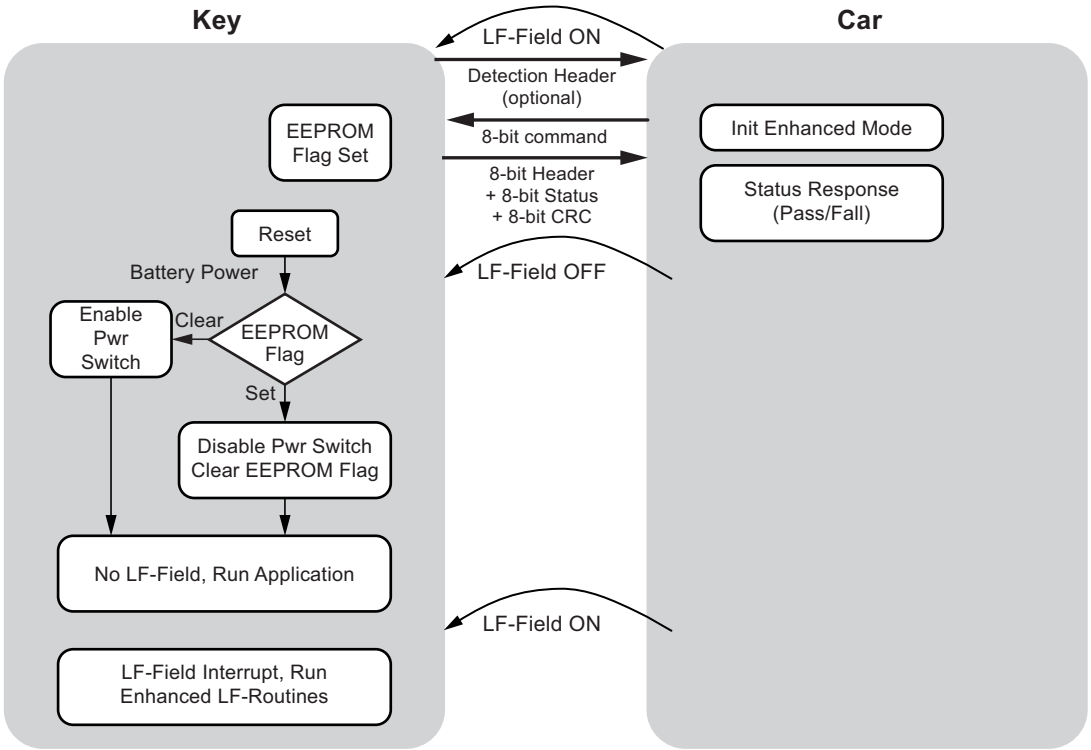
Once the LF field is removed, if battery power is present and contains sufficient voltage, a RESET is triggered internal to the device. When this occurs, the device will first check the enhanced mode flag in EEPROM as part of the RESET recovery routine. If the flag is set, the device will derive its power from the battery supply, and mask RESETS related to the presence/absence of the LF field. Second, the device will clear the enhanced mode flag to insure that this feature is only active until the next POR event. At the conclusion of this RESET recovery routine, program execution is vectored to the application code section.

7.5.3 Acknowledgement of LF Field and Execution of Enhanced Mode Features

After the enhanced mode has been configured, the LF field carrier presence/absence RESETS will be masked and will be unable to push the device into execution of the immobilizer firmware. Therefore, LF event handling routines must reside in the application code. This means that the end system designer must provide this capability. While LF field RESET events are masked, they will still generate an interrupt that will allow the application to acknowledge and respond. Some recommended features include battery recharge and custom commands. It is also possible to make use of the Atmel® immobilizer firmware by simply setting a jump to the immobilizer firmware reset address. This would allow the immobilizer routines to be run using battery supply.

To leave the enhanced mode the command “Leave Enhanced Mode” should be sent. This will enable again the internal power switch and generate a power on reset.

Figure 7-8. Init Enhanced Mode



7.6 Battery Recharge

Battery recharging using the LF field to provide the charging voltage is a special subset of the enhanced mode. There are a few conditions that apply to this feature.

The first is that it must be done while some charge is left in the battery. Without sufficient residual power in the battery, it is not possible for the device to be configured into the enhanced mode. Additionally, general purpose I/O pins used to control the external charging circuitry are disabled when the battery power is below the minimum operating voltage. This dictates that the application software includes provisions to prevent complete battery discharge.

Second, the source code that implements this feature must reside in the application space of the device. The immobilizer firmware fully occupies its dedicated program space.

8. Terms and Abbreviation Definition

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

Atmel Corporation 1600 Technology Drive, San Jose, CA 95110 USA **T:** (+1)(408) 441.0311 **F:** (+1)(408) 436.4200 | **www.atmel.com**