Chipsmall Limited consists of a professional team with an average of over 10 year of expertise in the distribution of electronic components. Based in Hongkong, we have already established firm and mutual-benefit business relationships with customers from,Europe,America and south Asia,supplying obsolete and hard-to-find components to meet their specific needs.

With the principle of “Quality Parts,Customers Priority,Honest Operation,and Considerate Service”,our business mainly focus on the distribution of electronic components. Line cards we deal with include Microchip,ALPS,ROHM,Xilinx,Pulse,ON,Everlight and Freescale. Main products comprise IC,Modules,Potentiometer,IC Socket,Relay,Connector.Our parts cover such applications as commercial,industrial, and automotives areas.

We are looking forward to setting up business relationship with you and hope to provide you with the best service and solution. Let us make a better world for our industry!

Contact us

Tel: +86-755-8981 8866 Fax: +86-755-8427 6832
Email & Skype: info@chipsmall.com Web: www.chipsmall.com
Address: A1208, Overseas Decoration Building, #122 Zhenhua RD., Futian, Shenzhen, China

ATA5580

125kHz Transponder with Open Immobilizer Software Stack and AES-128 Encryption

DATASHEET

Features

- AES-128 crypto transponder in plastic brick package
  - Includes coil and capacitor for tuned circuit antenna
- Radio frequency f_{RF} = 125kHz
- Contactless power supply
- Contactless bidirectional data communication interface
- High-performance AES-128 encryption hardware unit
- Atmel® open immobilizer stack
- 2K EEPROM for secret key storage, field user data and configuration data
- Error correction code support for NVM
- 32-bit unique ID
- Multiple configuration registers
- Modulation/coding: Biphase, Manchester, QPLM
- Configurable baud rate
- –40°C to +85°C operation temperature
- LGA-like brick package

1. Description

The Atmel® ATA5580 is a smart transponder module with an AES-128 encryption unit, customer EEPROM, a 125kHz LF front end and an LF ferrite antenna for wireless power supply and communication. All components are built up in a single pinless transponder package. The IC contains the highly configurable Atmel open immobilizer software stack.

1.1 Module Schematic

The Atmel ATA5580 transponder contains an ultra-low-power transponder IC with an AES-128 engine, an LF Antenna resonant circuit and a buffer capacitor.

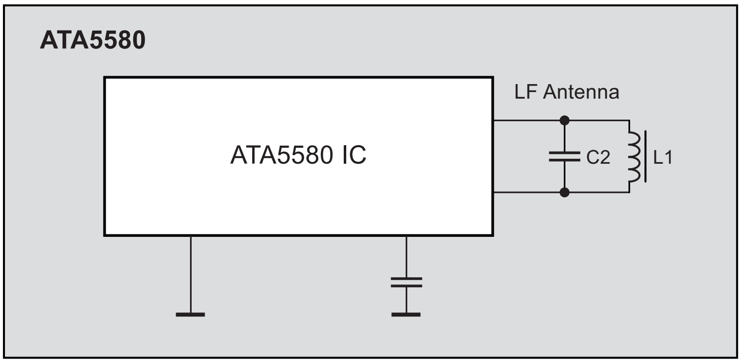
1.2 Functional Description

Atmel ATA5580 is designed for automotive immobilization applications in remote keyless entry (RKE) keys. The Atmel ATA5580 micro module consists of an ultra-low-power IC with AES-128 encryption engine and immobilizer front end, an LF ferrite antenna and capacitors for the antenna and as supply buffer.

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

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

Figure 1-1. Block Diagram



2. Atmel Open Immobilizer Protocol Description

2.1 Overview

2.1.1 Protocol Flexibility

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

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

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

2.1.2 Open Software Stack

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

In addition to selecting an open source and public AES-128 crypto function, the firmware includes user-configurable options that enable the engineer to "build" an authentication protocol that meets user requirements. The complete documentation of the protocol configuration options are made publicly available. The encryption and configuration of the authentication protocol are open source and freely available to customers free of charge.

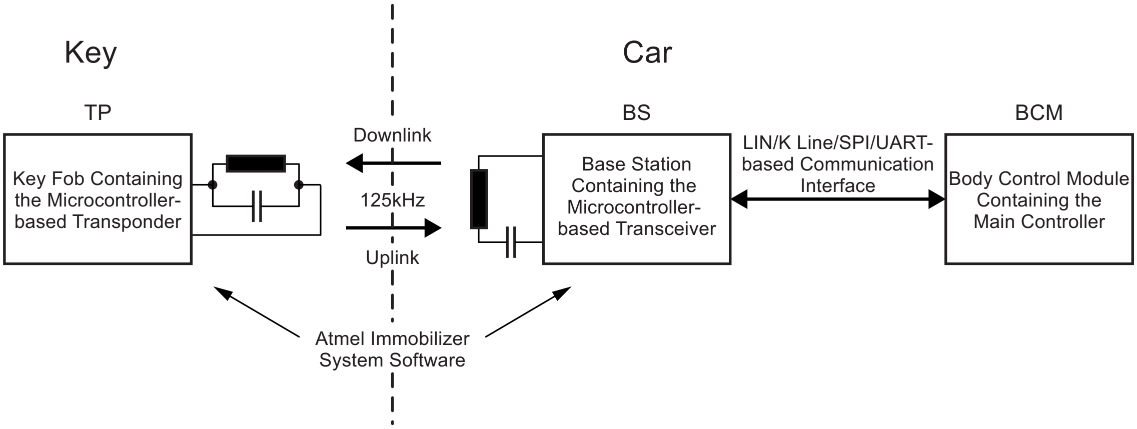
2.1.3 Production-Ready Software Implementation

Besides defining an open immobilizer protocol stack, Atmel chose to implement it in all car access devices with an embedded LF front end. This implementation complies with automotive grade development standards (CMMI - Automotive Spice) and is production-ready.

2.2 System Overview

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

Figure 2-1. System Overview


Key
Car
TP
BS
BCM
Key Fob Containing the Microcontroller-based Transponder
Downlink
125kHz
Uplink
Base Station Containing the Microcontroller-based Transceiver
LIN/K Line/SPI/UART-based Communication Interface
Body Control Module Containing the Main Controller
Atmel Immobilizer System Software

2.3 Device Support

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

- Atmel ATA5580: stand-alone transponder
- Atmel ATA5790: passive entry/go microcontroller with 3D LF receiver and transponder interface
- Atmel ATA5794: RKE microcontroller with transponder interface
- Atmel ATA5795: RKE microcontroller with transponder interface and Frac-N RF transmitter.

The Atmel base-station device ATA5272 includes a matched firmware library for implementing the entire system.

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

2.4 Firmware Features

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

2.5 Memory Partitioning

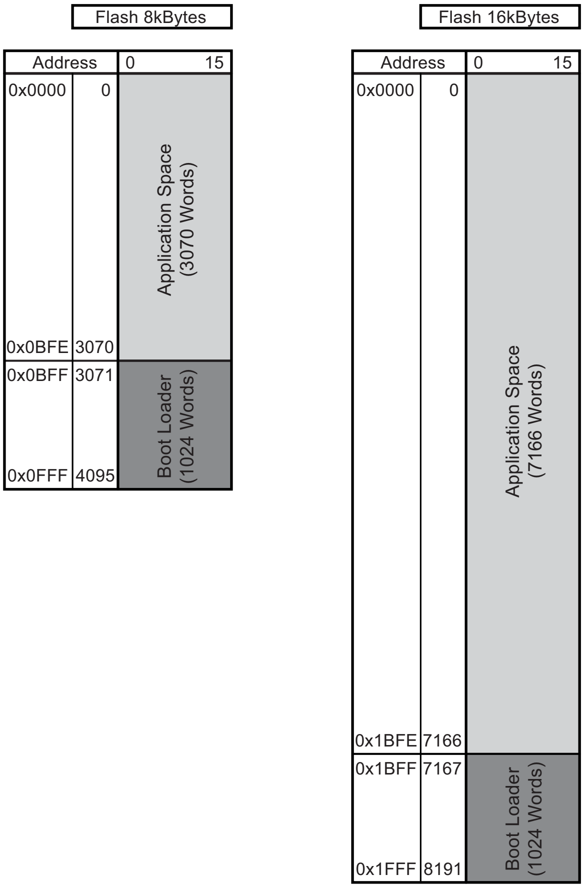
Except for the Atmel ATA5580, there are two types of memory on the Atmel devices used by both the immobilizer and the application. These memories need to be partitioned and some guidelines established to ensure reliable operation. Program code stored in flash memory is typically used as read-only memory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

2.5.1 Flash Memory

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

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

Figure 2-2. The Flash Memory Partition

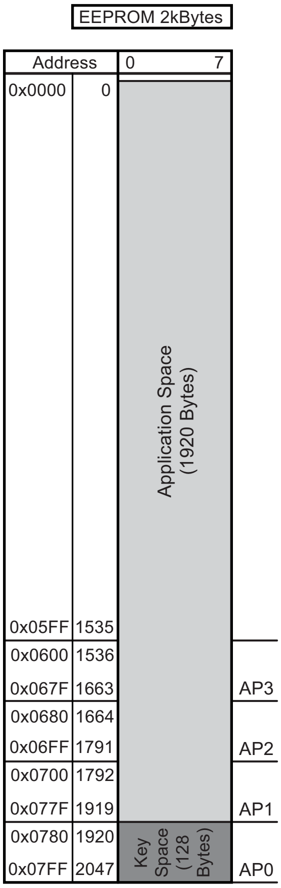


2.5.2 Non-Volatile Memory

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

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

Figure 2-3. EEPROM Page 1



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

2.5.2.1 Secret Key Storage

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

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

The size of the secret key is 16 bytes.

The secret keys for the immobilizer and the application must be stored based on the configuration stored in page 2.

Both secret key1 and secret key2 must be stored with two copies in their respective locations.

Figure 2-4 on page 7 represents the allocation of the secret key in the EEPROM memory.

Figure 2-4. The AP0 Memory Map

	128 Bit																
Secret Key	**Data 1**	**Data 2**	**Data 3**	**Data 4**	**Data 5**	**Data 6**	**Data 7**	**Data 8**	**Data 9**	**Data 10**	**Data 11**	**Data 12**	**Data 13**	**Data 14**	**Data 15**	**Data 16**	**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

AP0 128 Bytes

128 Bytes of Secret Key Memory

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

2.5.2.2 Configuration Memory Options

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

Data Check Disable

EEPROM address 0x0815 bit 0 allows the CRC data to be disabled for both the request frame and the response frame.

Data check disable (DCD): 0 = CRC enabled, 1 = CRC disabled

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

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

Authentication Format

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

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

This configuration bit is checked when the start authentication and memory access commands are executed. Details of this interaction are provided in the LF command set section.

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

Challenge and Response Length

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

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

Uplink Coding and Data Rate

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

Uplink modulation (MOD): 0 = Manchester, 1 = Biphase

The baud rate setting (0x0817) sets the threshold for the Manchester/Biphase encoder. This works in combination with the T2 prescaler (0x0818) to provide a very accurate and flexible transmission of data from the transponder to the vehicle. A typical value is recommended as 0x07 and 0x00 respectively to provide approximately 3.906kb/s.

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

Downlink Coding and Data Rate

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

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

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

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

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

Secret Key Selection and Transfer

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

Key select (KS): 0 = Secret key one, 1 = Secret key two

Secure key transfer (SKT): 0 = OFF, 1 = ON

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

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

Fob Power-Up

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

Detection header (TDH): 0 = OFF, 1 = ON

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

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

Default Secret Key

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

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

2.5.2.3 Fixed Identification

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

Unique ID

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

The customer ID stored at address 0x0804 may optionally be added to the unique ID.

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

Atmel Traceability

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

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

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

Software Revision

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

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

2.6 Device Initialization

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

2.6.1 Power-up Scenarios

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

2.6.2 LF Field Detection

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

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

2.6.3 Enhanced Mode Detection

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

2.6.4 Transponder Initialization

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

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

2.6.5 Reliable Communication Channel Indication

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

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 it has been achieved.

For the uplink to be successful, the transponder must modulate the carrier field with sufficient coupling and modulation depth for the base station to be able to 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 the header is visible on the demodulated output.

2.7 LF Physical Layer

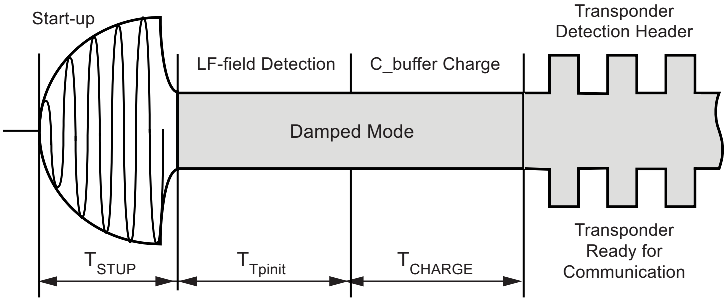
All communication between the base station and the transponder occurs using the LF field as the signal carrier. The LF communication link is established when the transponder transmits the LF channel detection header consisting of a Manchester coded sequence of “1010…” as a 125kHz signal which continues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

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

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

Figure 2-5. LF Physical Layer



2.7.1 Downlink

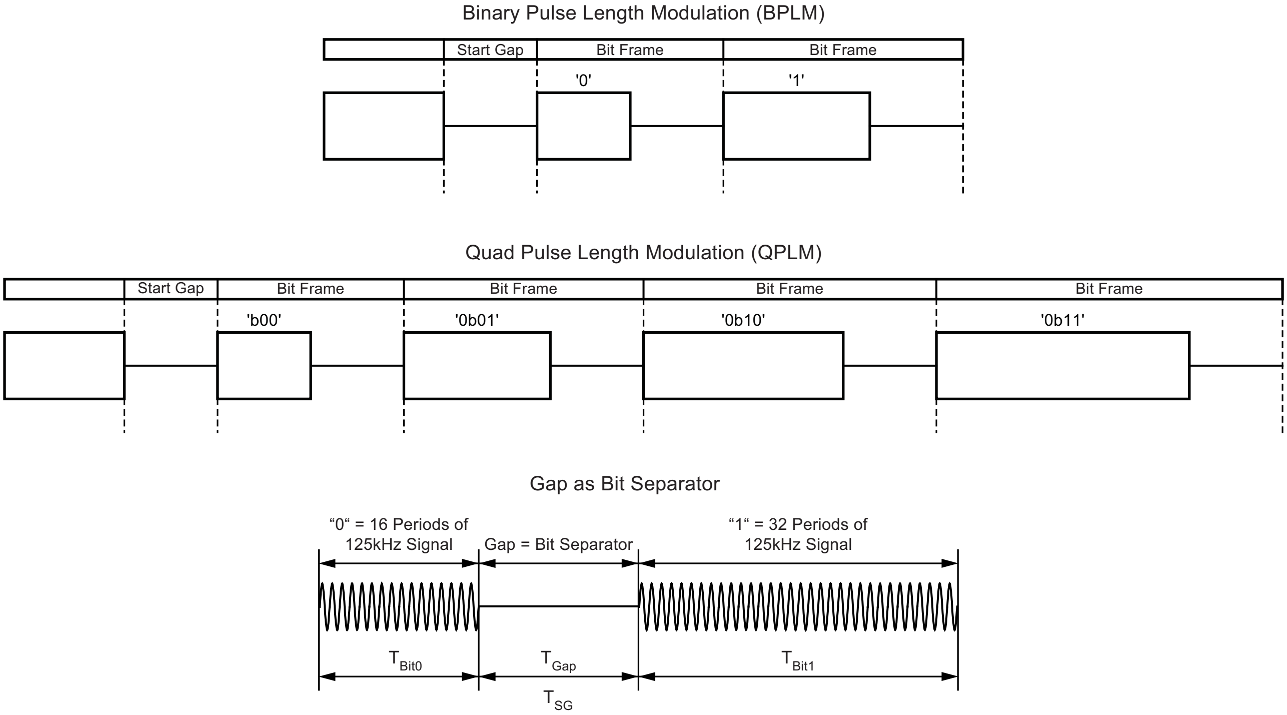
A downlink channel is established when the data is being transmitted from the base station to the transponder. The downlink communication uses amplitude modulation (AM) in the form of ON/OFF keying (OOK). To encode data pulse length coding is used. The pulses and LF bursts are separated by gaps. Data can be encoded in the following ways:

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

Quad pulse length modulation (QPLM): also known as 1-of-4 encoding. In this case a single pulse length is decoded into dual binary logic state (2-bit value).

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

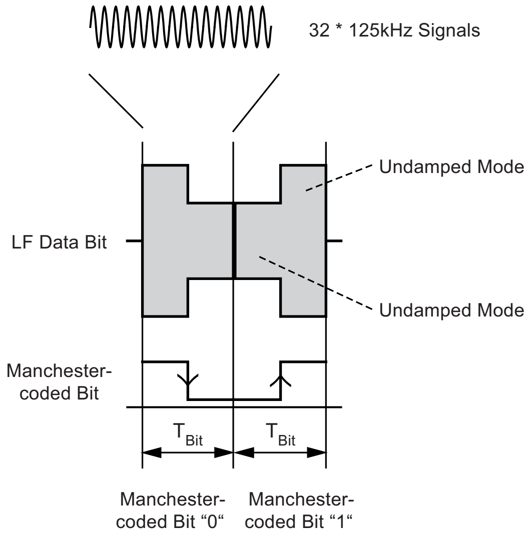
Figure 2-6. Downlink



2.7.2 Uplink

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

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



2.8 LF Communication

The protocol developed by Atmel® relies on two frame structures for the bidirectional communication. The downlink path from the base station to the transponder consists of a request frame. The uplink path uses the response frame defined below.

Communication sessions consist of a base station request, a 2ms delay, and a transponder response. All communication follows this process and creates functionality by executing a series of communication sessions. The base station request contains the means to utilize the command set provided by the Atmel firmware. All commands have a defined response that is returned from the transponder. The command set indicates that the response only occurs if communication is successful. Any errors that occur cause the transponder to signal the base station in a unique manner by sending a fixed 1kHz waveform. This allows very rapid detection of a problem. The exact cause of the error is stored and can be accessed by a dedicated command.

2.8.1 Request Frame Definition

All transactions are initiated by the base station sending the following:

Command field = 4-bit command + 4-bit command CRC

Data field = variable bit length payload (optional based on command)

CRC field = payload CRC8 (optional based on presence of payload data)

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

2.8.2 Response Frame Definition

All responses the transponder makes to the base station include sending the following:

Header field = recognizable pattern fixed at 0xFE

Data field = variable bit length payload (optional based on command)

CRC field = payload CRC8 (optional based on presence of payload data)

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

2.9 LF Command Set

2.9.1 Read UID

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

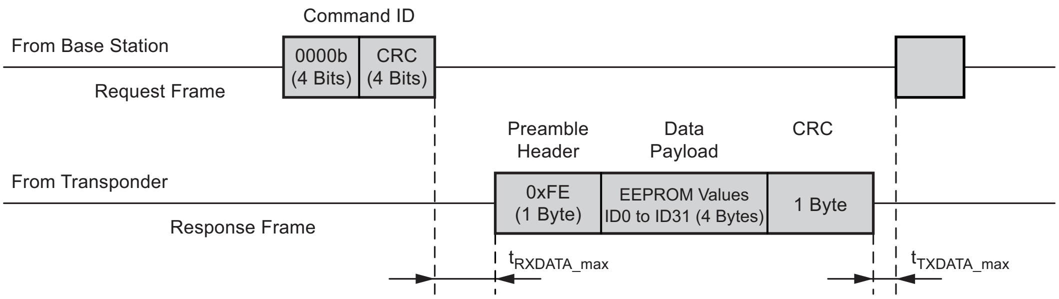
Table 2-1. The Read UID (Request Frame)

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

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

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

Figure 2-8. The Read UID Sequence



2.9.2 Transponder Error Status

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

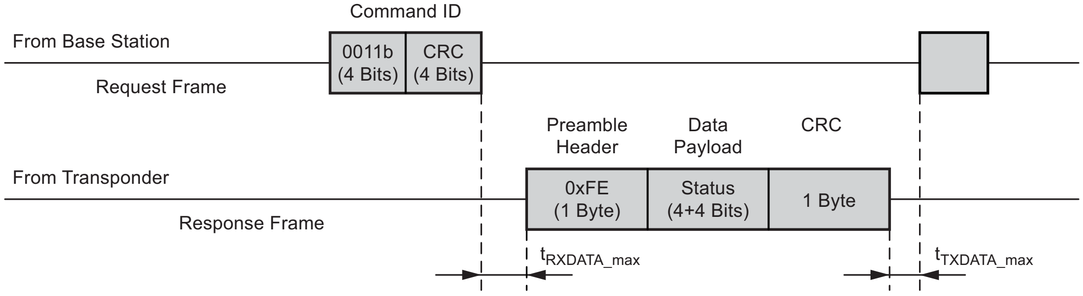
Table 2-3. The Transponder Error Status (Request Frame)

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

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

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

Figure 2-9. The Transponder Error Status Sequence



2.9.3 Start Authentication

The immobilizer authentication protocol is to be based on challenge-response topology. This can be the unilateral authentication (UA) method or bilateral authentication (BA).

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

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

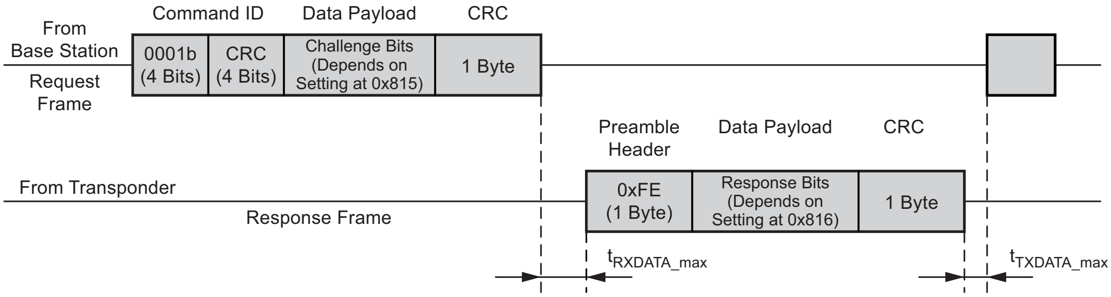
Table 2-5. Start Authentication (Request Frame)

Field	Size	Values	Description
Command ID	4 + 4 bits	0001b + 0011 CRC	Start authentication
Data payload	Varies (104 or 128 bits recommended)	Challenge bits	Depends on EEPROM page 2 setting
CRC	1 byte	Calculate	

Table 2-6. Start Authentication (Response Frame)

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

Figure 2-10. The Start Authentication Sequence



2.9.4 Learn Secret Key1

This command starts the learn secret key1 process for the first secret key. Depending on the configuration setting stored in EEPROM page 2 at address 0x811(bit 6) it is either open transfer or secure transfer. If the bit (SKT- secure key transfer bit) is 0, the transfer is open mode and if the bit is 1, the transfer is secure mode. The request frame carries a128-bit secret key data payload (may be encrypted during secure transfer). The 128-bit key transferred through this command is stored in AP0 key position 1 (0x7C0) along with two copies. The response frame consists of a status byte at the data payload. The status byte is stored in RAM and updated with each communication session. The status byte consists of the last command received (MS 4 bits) and an error flag (LS 4 bits). Status byte [7:4]: Four MSBs of the field contain an echo of the command received in the last request frame. Status byte [3:0]: Four LSBs of the field contain status information in encoded form.

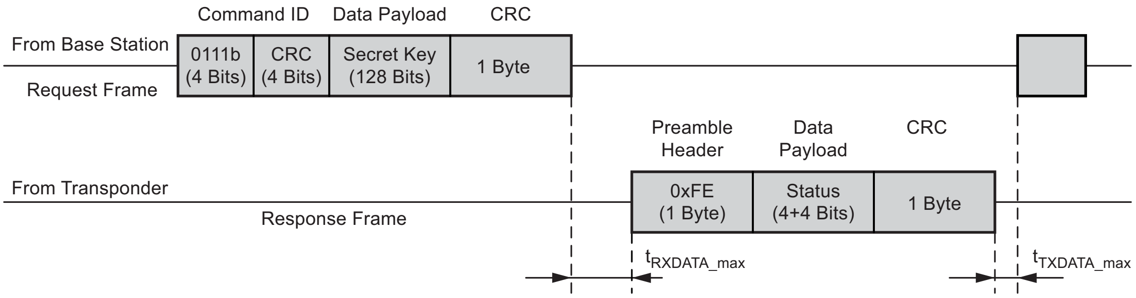
Table 2-7. The Learn Secret Key1 (Request Frame)

Field	Size	Values	Description
Command ID	4 + 4 bits	0111b + 1001 CRC	Learn secret key1
Data payload	128 bits		AES-128 (possibly encrypted) secret key
CRC	1 byte	Calculate	

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

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

Figure 2-11. The Learn Secret Key1 Sequence



2.9.5 Learn Secret Key2

This command starts the secret key2 learning process. Depending on the configuration stored in EEPROM at address 0x811(bit 6) it is either open or secure transfer. If the bit (SKT - secure key transfer bit) is 0, the transfer is open mode and if the bit is 1, the transfer is in secure mode. The request frame carries a 128-bit secret key data payload (may be encrypted during secure transfer). The 128-bit key transferred through this command is stored in the AP1 key position 2 (0x780) along with two copies. The response frame consists of a status byte at the data payload. The status byte is stored in RAM and updated with each communication session. The status byte consists of the last command received (MS 4 bits) and an error flag (LS 4 bits).

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

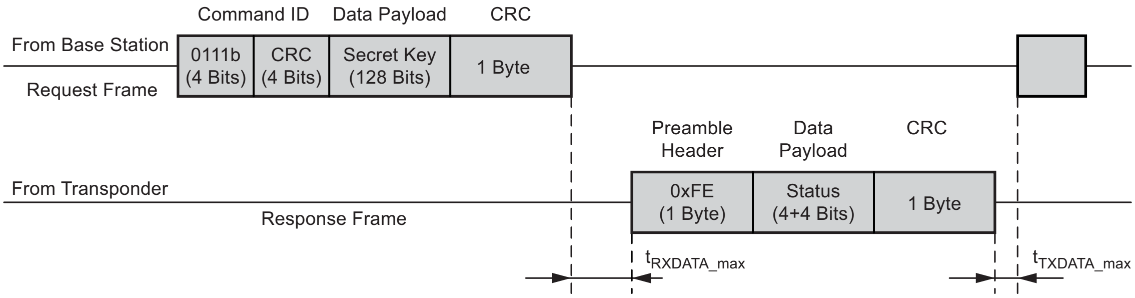
Table 2-9. The Learn Secret Key2 (Request Frame)

Field	Size	Values	Description
Command ID	4 + 4 bits	1000b + 1011 CRC	Learn secret key2
Data payload	128 bits		AES-128 (possibly encrypted) secret key
CRC	1 byte	Calculate	

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

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

Figure 2-12. The Learn Secret Key2 Sequence



2.9.6 Initiate Enhanced Mode

This command initializes the enhanced mode command structure and switches the transponder into enhanced mode when it enters the VFLD the next time by setting the enhanced mode flag in EEPROM. In addition, this command begins a sequence to place the transponder into the enhanced mode where the battery supply is used during transponder communication. An EEPROM flag having a TBD value is stored at the TBD address. The address is checked at each POR to determine if the power switch should be disabled. Once the flag is set by this LF command, the NEXT power cycle causes the following LF session to be operated using battery power. It occurs only once each time this LF command is received.

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

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

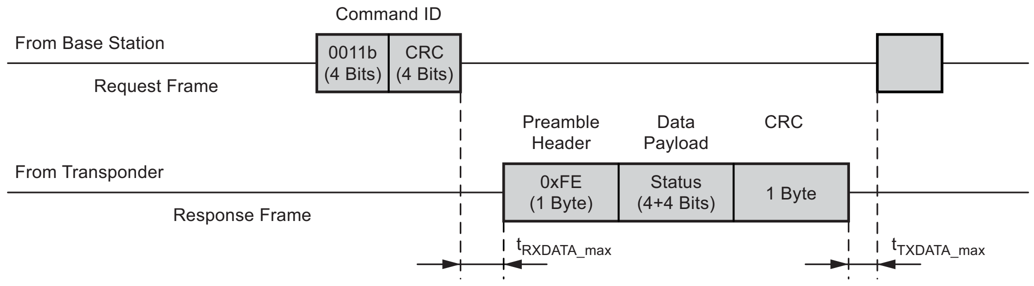
Table 2-11. The Initiate Enhanced Mode (Request Frame)

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

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

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

Figure 2-13. The Initiate Enhanced Mode Sequence



2.9.7 Repeat Last Response

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

The response frame matches the response from the previous command.

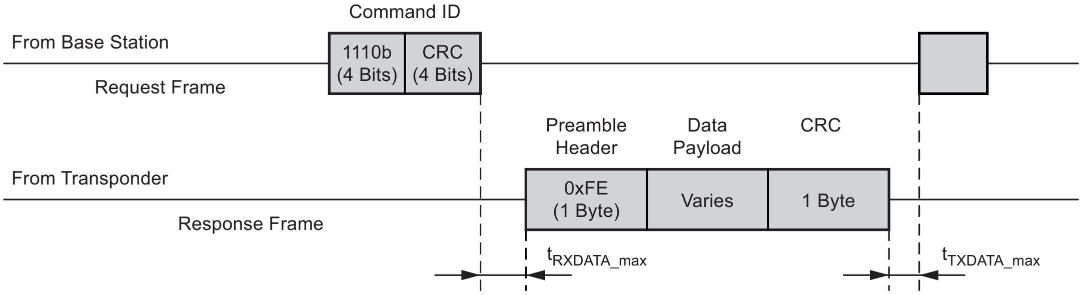
Table 2-13. Repeat Last Response (Request Frame)

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

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

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

Figure 2-14. The Repeat Last Response Sequence



2.9.8 Read User Memory

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

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

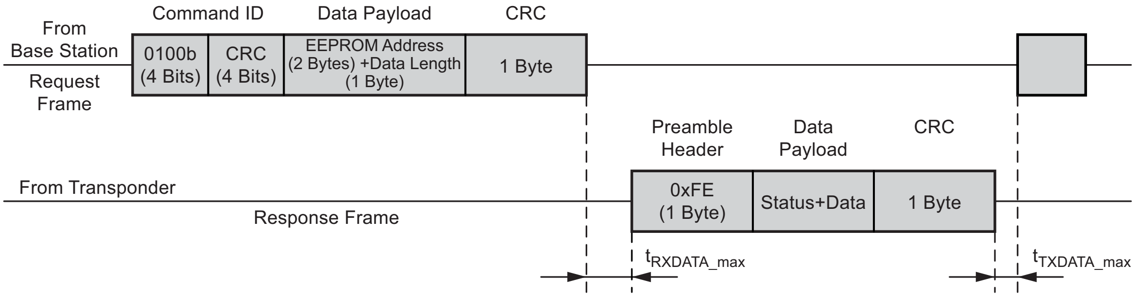
Table 2-15. Read User Memory (Request Frame)

Field	Size	Values	Description
Command ID	4 + 4 bits	0100b + 1100 CRC	Read user memory
Data payload	2 bytes + 1 byte		EEPROM address + data length
CRC	1 byte	Calculate	

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

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

Figure 2-15. The Read User Memory Sequence



2.9.9 Write User Memory

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

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

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

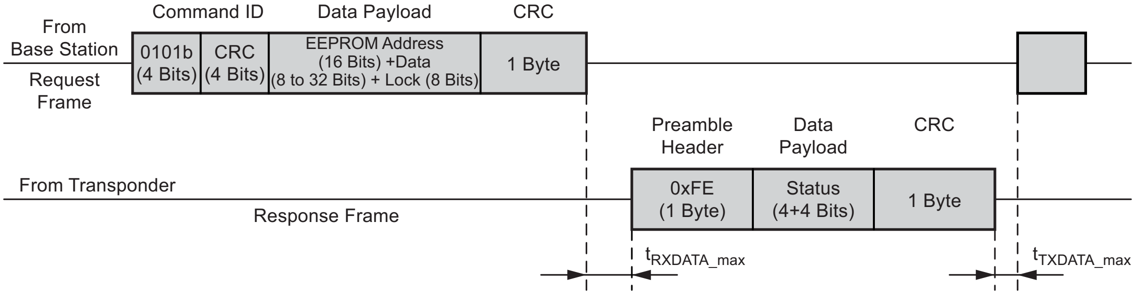
Table 2-17. Write User Memory (Request Frame)

Field	Size	Values	Description
Command ID	4 + 4 bits	0101b + 1111 CRC	Read user memory
Data payload	16 bits + 1 to 4 bytes + 8 bits		EEPROM address + data lock
CRC	1 byte	Calculate	

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

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

Figure 2-16. The Write User Memory Sequence



2.9.10 Write Memory Access Protection

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

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

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

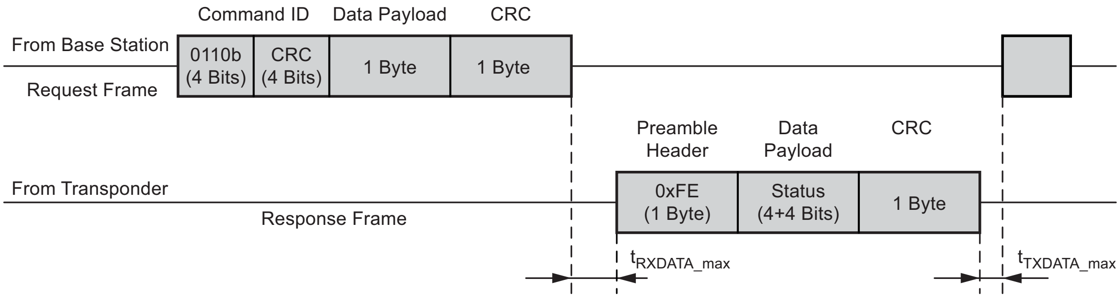
Table 2-19. Write Memory Access Protection (Request Frame)

Field	Size	Values	Description
Command ID	4 + 4 bits	0110b + 1010 CRC	Write memory access protection
Data payload	1 byte		Protection scheme
CRC	1 byte	Calculate	

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

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

Figure 2-17. The Write Memory Access Protection Sequence



2.9.11 Leave Enhanced Mode

This command clears the enhanced mode flag from EEPROM.

If the transponder receives the command “leave enhanced mode,” the internal power switch inside the transponder front end is enabled. If the LF field is active, the internal power management automatically switches to the field-supplied mode. This then generates a power-on reset and the immobilizer firmware is then executed.

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

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

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

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

2.10 Communication Integrity and Error Mitigation

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

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

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

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

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

3. Immobilizer Functionality

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

3.1 Authentication

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

3.1.1 Unilateral Authentication

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

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

1. The base station sends the "read UID" LF request.
2. The transponder responds by providing the 32-bit UID in its "response frame."
3. The base station then sends the "start authentication" request including a random number "challenge."
4. The transponder returns an "encrypted response" message to the base station.

Notes:
1. The "challenge" uses the 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" uses the bit length defined by configuration memory address 0x081A.
4. When necessary for encryption, the challenge is extended by first padding the upper bit positions with the 32-bit UID, then with "0"s as needed, and in this order, to reach 128 bits.

A graphical example is shown in Figure 3-1 on page 26.