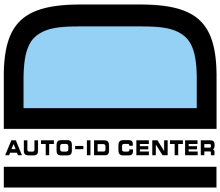
Published June 1, 2003. Early Release July 2003



TECHNICAL REPORT

860MHz–930MHz Class 0 Radio Frequency Identification Tag Protocol Specification Candidate Recommendation, Version 1.0.0

Auto-ID Center

AUTO-ID CENTER MASSACHUSETTS INSTITUTE OF TECHNOLOGY, 400 TECHNOLOGY SQ, BUILDING NE46, 6TH FLOOR, CAMBRIDGE, MA 02139-4307, USA

ABSTRACT

This document specifies the radio frequency communication interface, Reader commanded functionality requirements, anti-collision protocol for an Auto-ID Center Class 0 radio frequency identification (RFID) Tag operating in the frequency range of 860MHz-930Mhz. A Class 0 tag is designed to communicate only its unique identifier and other information required to obtain the unique identifier during the communication process.

WHITE PAPER

860MHz–930MHz Class 0 Radio Frequency Identification Tag Protocol Specification Candidate Recommendation, Version 1.0.0

Contents

WHITE PAPER

860MHz–930MHz Class 0 Radio Frequency Identification Tag Protocol Specification Candidate Recommendation, Version 1.0.0

Contents

A. BACKGROUND

1. OBJECTIVES

This document provides both mandatory and optional specifications for a low cost item identification tag operating in the ultra high frequency band in accordance with accepted and evolving worldwide standards. The tag contains an Electronic Product Code™ (EPC™) used for item identification, a cyclic redundancy check, and a destruct code. The identification of tags is performed using the first two elements, and tag destruction is performed using all three.

2. DOCUMENT STRUCTURE

Although the sections are simply serially numbered, the document is divided into three major parts. Part A provides general background, and gives a description of the context within which the standard is intended to operate. Part B provides a brief synopsis of the operating characteristics. Part C provides a definition of the **air interface** and **command set**; it covers signaling waveforms and extends to a description of detailed command structure and operation.

3. STATUS

This document defines the Auto-ID Center's Class 0 Candidate Recommendation Version 1.0.0 for Tags operating in the 860MHz–930MHz frequency range.

4. TERMINOLOGY

4.1. General

We take the opportunity here to clarify the terminology of this document.

We will not use the terms uplink, downlink, or forward link. Directions of communication will be described as **reader-to-tag**, or **tag-to-reader**.

Numbers, when they are written, have the most significant digit on the left and the least significant digit on the right.

In serial transmission, we will make no assumption as to whether the most significant bits or the least significant bits are transmitted first. We will make an **explicit statement** in every case.

We take the term **air interface** to mean the waveforms of the different symbols used in both the reader to tag signaling and tag to reader signaling, and the rules for building commands, but it does not include the commands themselves. It does include the coding of the tag replies.

The term **command set** is taken to mean the set of tag commands by means of which the tag population may be explored or modified by the reader.

The term operating **procedure** refers to how we should use the command set to identify or modify tags.

The term **protocol** is intended to refer collectively to the elements air interface, command set and **operating** procedure.

The term **Reset** is a long (~800 us) period of un-modulated, and sufficient, reader power that instructs all tags to jump to a **calibration state**. The Reader may issue a reset anywhere in the state diagram. Reset does not affect the tag internal flag that indicates that the tag has previously been read.

The term **contention** is defined as an event where multiple tags simultaneously reply to a stimulus from the reader. Contention is not necessarily a destructive process, but does need resolution.

The term **collision** is defined as an event where multiple tags simultaneously reply to a stimulus from a reader, where information is lost in the process. A classic example of a collision would be a packet collision.

The term **negotiation** is defined as the process where the tags send to the reader data bits in the tag to reader link, and the reader acknowledges the data in the reader to tag data link, and thus defines a path through the tag population binary tree.

The term **clock** refers to a time reference within the tag against which various operations within the tag are referenced.

The term **clock start** refers to the start of reader modulation to the lower power RF emissions. This point in time is the start of a bit period.

The term **singulation** refers to a process of negotiation, which culminates in a single tag being selected by the interrogator for further processing via interrogator commands.

The term **singulation string** refers to a string of digits used by the tag in the process of singulation of a tag. The singulation string may be the tag identity contained in its memory, a separate string contained in memory for the purpose of singulation, or may be randomly generated within the tag by a range of processes.

The term **spoofing** refers to an attack on the privacy or integrity of an information system in which the attacker creates a misleading context in order to trick the victim into making an inappropriate security-relevant decision. A spoofing attack is like a con game: the attacker sets up a false but convincing world around the victim. The victim does something that would be appropriate if the false world were real. Unfortunately, activities that seem reasonable in the false world may have disastrous effects in the real world.

4.2. Binary Trees

Since this document discusses the interpretation of tag EPC™ codes in terms of binary trees we will take the opportunity to clarify in Section 6.5 some tree concepts. This will be done, however, after the EPC™ concepts themselves are defined in Section 4.3.

4.3. Electronic Product Code™ (EPC™)

4.3.1. Introduction

This section describes the seven varieties of Electronic Product Code so far defined by the Auto-ID Center, and acknowledges that further varieties may also be defined.

The tag and reading system specification provided in this document is intended to apply to all varieties so far defined.

4.3.2. EPC™ Structure

The EPC™ is a representation of an Electronic Product Code. In the EPCs™, there are four fields, which are, in order: a **version number**, defining the variety of EPC™ among a number of possible structures; a **domain manager number** which is effectively a manufacturer number; an **object class** which is equivalent to a product number; and a **serial number**. An Electronic Product Code may be representable with multiple versions of EPCs™ and may not be representable with some versions of EPC™.

The below table gives, for the seven varieties of EPC™ so far defined, the size, in bits, of each field. The table also indicates, for each variety, the leading bits, i.e. the version number.

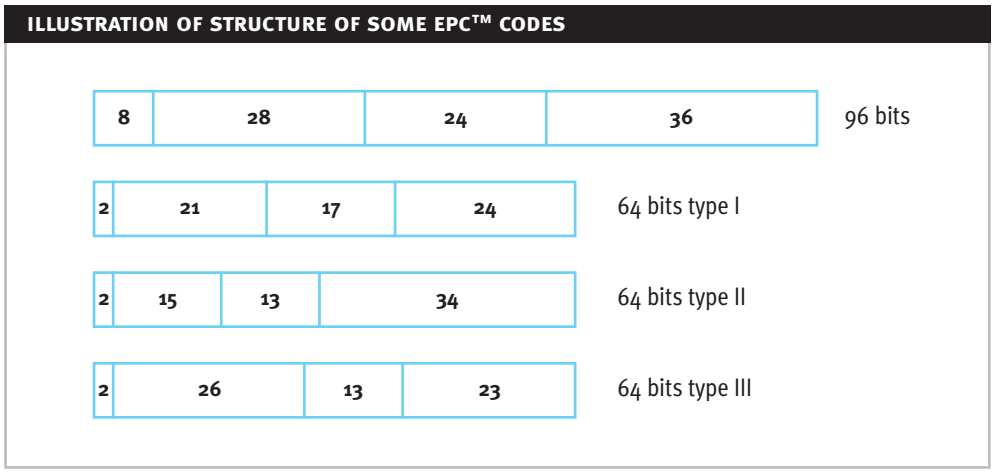
Table 1

EPC™ TYPE	VERSION SIZE	VERSION NUMBER	DOMAIN MANAGER	OBJECT CLASS	SERIAL NUMBER	TOTAL
EPC-64 TYPE I	2	01	21	17	24	64
EPC-64 TYPE II	2	10	15	13	34	64
EPC-64 TYPE III	2	11	26	13	23	64
EPC-96 TYPE I	8	0010 0001	28	24	36	96
EPC-256 TYPE I	8	0000 1001	32	56	192	256
EPC256 TYPE II	8	0000 1010	64	56	128	256
EPC256 TYPE III	8	0000 1011	128	56	64	256

4.3.3. Illustration

Figure 1 below provides to scale an illustration of some of the varieties the Electronic Product Code™ just defined.

Figure 1: The code structure is Version-Domain manager-Object class-Serial number



4.3.4. Additional Information

The definitions of section 4.3.2 do not preclude the definition of future versions of EPC™ code or designing tags for them.

4.3.5. Cyclic Redundancy Check (CRC)

The reader-to-tag link uses a 16-bit CRC (defined below), and is stored in the tag.

Figure 2

CRC DEFINITION				
CRC TYPE	**LENGTH**	**POLYNOMIAL**	**PRESET**	**RESIDUE**
ISO/IEC 3309	16 bits	$x^{16} + x^{12} + x^5 + 1 = 0x8408$	0xFFFF	0xF0B8

The CRC is calculated on all N bits of the EPC™ starting with the MSB thereof.

A further transformation on the calculated CRC is made. The value stored in the tag, and which is attached to the message for transmission, is the one's complement of the CRC calculated as in Table 2.

For ease of checking of received messages, the two CRC bytes are often also included in the re-calculation. In this case, the expected value for the residue of the CRC generated in the receiver is 0xF0B8.

5. OPERATIONAL CONTEXTS

5.1. Class 0 Tags

We call factory programmed read-only tags encoding an EPC™ and CRC as described above and conforming to the requirements of this specification Class 0 tags.

5.2. Higher Class Tags

In the design of the EPC™ system, issues such as security and privacy, sensor networks and ad hoc networks have been considered. To this end, higher Class tags that contain more functionality than that contained within the Class 0 tags are contemplated and planned for.

6. MULTIPLE TAG READING

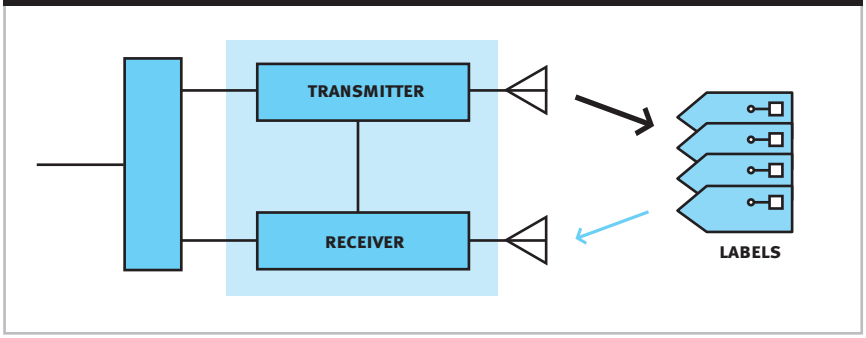
6.1. Illustration

Figure 3 below illustrates an example of a multiple electronic tag reading system. It is assumed the tags are **passive**, i.e. they contain no internal energy source.

In the figure, a group of tags is interrogated by a **reader** containing a **transmitter** for generation of an interrogation signal that supplies power and information to the tags. The reader also contains a **receiver** for reception of a reply signal from the tags and for decoding that signal. The reader operates under control of a **controller** that supplies the decoded signal to external apparatus, and manages the interrogation process.

Figure 3

ILLUSTRATION OF A MULTIPLE TAG READING SYSTEM



6.2. Functions Required of the Tag

The Class 0 tags must have the functions of:

- Being factory programmed with EPC™, 24 bit kill code, and CRC,
- Being read by the reader,
- Being selected as part of a related group of tags,
- Being individually destroyed, and
- Not containing memory readable and writable by a reader.

6.3. Performance Factors

The performance of an UHF EPC™ tagging system is influenced by the following factors:

- **Electromagnetic compatibility regulations.** Such regulations are considered in detail in Section 7. Their principal impact is on the choice of viable anti-collision algorithms that may be employed at UHF, and on the operating range achievable in simple standardized field creation systems.

- **Human exposure regulations for electromagnetic fields.** Such regulations are considered in detail in Section 8. They will have an impact on licensing considerations.

- **Tag antenna size.** The principal issue to consider is that electrically small tags require tuning with high quality factor to be efficient. In some circumstances, this may be an advantage, but environmental mistuning may become a problem. An effort should be made to minimize tag operating power and to maximize backscatter performance.

- **Communication parameters of the air interface.** The proposal below incorporates, for each direction of communication between reader and tag, appropriately compact communication with a suitable level of security for the EPC™ reading context.

- **Anti-collision algorithms for multiple tag reading.** The principal impact is on the number of practicable tag reads per second. This proposal will be based on a version of a binary tree-scanning algorithm that has been optimized, for performance and for robustness, when many readers occupy a single environment.

In the face of the complexity and inter-dependence of all of the above issues, we will propose what is believed, based on experience, sensible and achievable tag and system parameters.

6.4. Design Objectives

The design objectives pursued in producing the specification of this document are as follows.

- The design must allow production of very low cost tags.
- The signaling and tag operations should support in at least some of its realizations selection of groups of tags by a combination of EPC™ version, domain manager and object class.
- The signaling and system operation should allow high throughput in terms of tag reads per second.
- The design must allow for a good tag operating range.
- The design must allow for tolerance of nearby similar tag reading systems.

6.5. Approaches

This specification describes a binary tree scanning anti-collision protocol that is an implementation of a "reader talks first" methodology. By this, we mean that no tag transmits any information prior to a specific request by the reader. Collision-free refers to the fact that the simultaneous replies from multiple tags represents a contention for reader attention yet need not represent a loss of information. This protocol is a contention-resolving and collision-free method for negotiating data from multiple tags.

Reader-to-tag communication is accomplished through an amplitude-modulated (AM) carrier. Tag-to-reader communication is accomplished through the passive backscatter of the tag-to-reader carrier to produce widely separated sub-carrier tones.

A population of tags to be read by the reader can be represented as a binary tree. Our diagrammatic representation of a tree will descend from the **root**, at the top (not considered to be part of the tree), with **branches** leading downwards to more **nodes**. In the sense that scanning the tree from root to leaf fully defines an EPC™, and hence a particular item, terminating nodes at the bottom represent **leaves** at which products may be present.

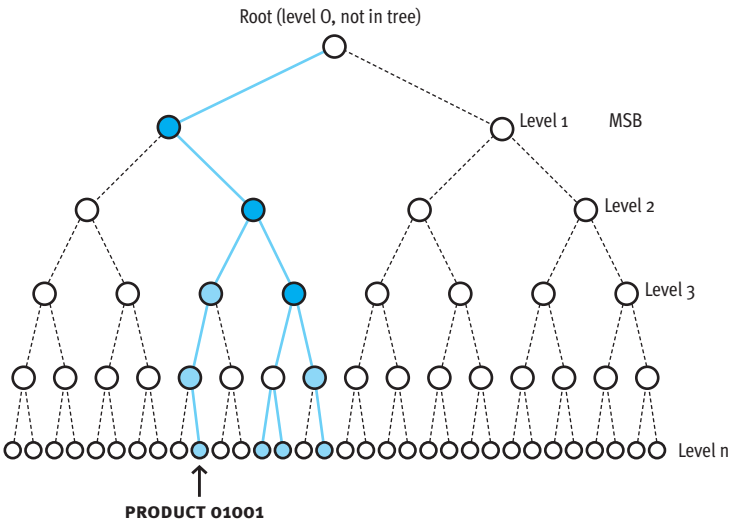
Such a tree representation is shown in Figure 4. In this case, we have placed the MSB of the EPC™ adjacent to the root of the tree. The LSB is considered by default to be at the leaf end of the tree.

We can see in this figure a **unique path** through the tree defined by the EPC™ of a particular product.

In some tree interpretations, the tree may extended downwards to lower levels by including the CRC, but in any case of extension from a product (as defined previously), the path in the extension will not involve any further branching.

Figure 4

REPRESENTATION OF AN EPC™ AS A TREE



We regard a node as **populated** if there are branches descending from it to a product or it is a bottom node corresponding to a product, otherwise we describe the node as **unpopulated**. We can classify a populated node as **singly populated** if only a single branch leading to a product descends from it, or **multiply populated** if more than one branch leading to a product descends from it. We have, in the tree illustrated in Figure 4, shown unpopulated nodes with a white interior, singly populated nodes with a lightly shaded interior, and multiply populated nodes with a more heavily shaded interior.

Our approach to negotiating the tags represented in the population binary tree assumes that there will be one reader in communication with all RFID tags of the population. Any contention is then defined as occurring in the communication channel from tag to reader. Interference between tags, or **masking**, is possible but extremely unlikely to occur. The masking of weak tag-to-reader signals by strong tag-to-reader signals is avoided completely in the subcarrier tone encoding method employed, but destructive interference between tags of equal strength can occur, but only with very low probability. Although destructive interference may happen in the tag-to-reader data link, it can only be intermittent as changes in the reader-to-tag carrier frequency, and drift in the internal tag subcarrier tones, can always be counted on to eliminate destructive interference masking. The operation of multiple readers may jam both the reader-to-tag and tag-to-reader data links, but proper reader design and the speed of the protocol can mitigate these inherent problems.

The protocol performs tag singulation on a bit-by-bit basis as information is progressively received. Each tag response is defined by two sub-carrier frequencies, one for a binary 0, and the other for a binary 1. In such a manner, with high reliability, many tags can communicate without collisions. It is not important that a receiver cannot differentiate one data 0 from multiple data 0s (or a single 1 from multiple 1s for that matter), just that there exists a data 0. Because 0s and 1s are communicated as distinct tones, the reader can simultaneously receive both. Thus on a single bit backscattered from the population data is not lost because of contention. This explains how to communicate one bit of information without collision.

After each collision-less tag-to-reader bit communication, the reader, by choosing one of the two possible binary tree branches, directs tags to either **remain active,** or go **temporarily inactive**. In particular, tags

that receive a bit that matches the last bit backscattered remain active; those that do not see such a match will go temporarily inactive and wait to participate in the next tree traversal. The negotiation continues for all bits of the singulation string, and results in a **tag singulation**. Once the tag has been singulated, the reader may send commands to this tag and/or put the tag to sleep (**dormant state**). This method is applied repeatedly for each tag in the population. Provisions are made for entering into a global command state before a tag is singulated, whereby multiple tags can be addressed and manipulated simultaneously.

6.6. Use of Identification Numbers for Tag Singulation

A typical approach in RFID is to use the unique but low entropy identification number (EPC™), in its negotiations to singulate a tag. Several disadvantages are associated with this method, such as the potential lack of efficiency and security. Given the reality of a typical application and the physics of RF power transmission, a reader will never address more than a few thousand tags at a time. A few thousand tags could be uniquely represented by a 12-bit number, so it is inefficient to require the transmission of the complete identity when 12 bits or slightly more could do the job. A small high entropy number could be used as a proxy for the EPC™ data encoded in the tag thereby speeding tag negotiations in some circumstances while preventing the widespread broadcast of the user data (i.e., EPC™ data.)

This specification will detail the use of three ID numbers for the process of tag singulation, ID0, ID1, and ID2. ID2 is intended to store EPC™ data and its associated CRC and can be used for both tag singulation and just transmission from the tag to the reader. ID1 is a static pseudo-random number that is contained on chip, and is used in tag singulation and in a method for recalling an already established tag identity stored in ID2. ID0 is a fully randomized number that is generated on chip as needed, and will be re-randomized at each address by the reader of the full population of tags. ID0 may be used in tag singulation, but must always follow with the reading the EPC™ data in ID2 for establishing a tag identity. Under interrogator command, any one of ID2, ID1, or ID0 may be used for singulation.

While the EPC™ (ID2) is the default number to be negotiated and allows easy product selection, it does exhibit a low level of security. Security is compromised when the EPC™ (stored in ID2) is broadcast via the high power reader emissions during the process of singulation. Singulation on a dynamically generated purely random number (ID0) will provide a highest level of security since it contains no EPC™ information whatsoever, but sacrifices some of the communication robustness and repetitive tag read speed for that greater security. Tag speed is reduced because it is still necessary to extract the EPC™ information from the tag even though the contents will be contained only in extremely low power tag emissions. However, with an appropriately complex reader algorithm, repeated singulation of a tag population based on static pseudo-random numbers (ID1) gives the greatest read performance while maintaining a moderate level of security. The security level still remains high as the EPC™ information is only contained in tag emissions, but repetitive addresses of a same tag population need only singulate on this static ID1 number and would not be required to repetitively extract the EPC™ information from the tag.

6.7. Secure Transmission Methodology

The communication channel that is most susceptible to eavesdropping by a distant receiver is the reader transmit channel, because it is at such a high power level. In some environments, it may be of concern that the reader broadcasts every bit of the ID during negotiations.

The use of the data page ID0 or ID1, which contains data for singulation only, and does not contain any application data whatsoever, would prevent eavesdropping on application information contained in the reader-to-tag link during singulation. Singulation on ID0 or ID1 would not alone provide the necessary EPC™ data to the reader. Once a tag has been singulated with ID0 or ID1, the communication of the pertinent user data can be more securely transmitted on only the tag-to-reader link with the additional command READ described in detail in section 13.4.7. The tag-to-reader link is very low power and as a result is considered a very short range, very localized signal that is not easily compromised to eavesdropping.

ID1 is a pseudo-random number that can be derived from a portion of the unique user data (just the CRC), or a random generated number stored in the memory of the tag at manufacture. ID0 is a dynamic random number generated anew every time the tag is singulated. Neither ID0 nor ID1 contains information that would allow disclosure of the EPC™ information.

For a pseudo-random number derived from a portion of the unique user data, even though there is no evident user data, an eavesdropper could obtain information on the quantity of tags and possibly track tag movements. For a dynamically derived random number, the tag movements are completely obscured from an eavesdropper but at a slight degradation in noise rejection capability of a complex reader implementation in the communication channels.

6.8. Performance of Protocol in RF Noisy Environments

There are many factors to be concerned with when evaluating a protocol performance within an RF noisy environment. Typically, one would expect to evaluate reader-to-tag channel performance and impact separate from tag-to-reader channel performance and noise impact. However, this particular protocol has benefit from evaluating multiple channels and the impact of a noise source into all channels simultaneously. Expanding somewhat on text in section 6.5, the protocol effectively utilizes a bit for bit acknowledgement. In some cases, this is a reader acknowledging a tag data bit, and in other cases, the tag acknowledges the reader data bit. It is important to note that for each bit, data occupies a reader-to-tag channel, and one of two tag-to-reader channels. A data error is very likely to be detected at each bit as noise would be required to impact simultaneously two separate frequencies. Additionally, either a CRC or parity information is used at the end of data to further aid in the detection of a data error. The combination of the above two forms of error detection allow this protocol an extremely high robustness against noisy RF environment impact.

The early and rapid detection of data transmission errors accommodates faster recovery and less time wasted in re-transmission of data. For simplicity of reader design, this protocol does not insist upon the most extravagant reader designs for noise detection and recovery, but does insist upon support for flexibility. Many alternate reader designs with different levels of performance of data transmission error detection and recovery will suit a variety of applications with different equipment budgets. The following characteristics in general are available to change as reader designs and application requirements may dictate:

- Reader symbol definitions (elongated for less impact by noise)
- Tag return symbols occur both above and below the carrier frequency, allowing for reader selection of least noise impact channel.
- Tag return symbol length is also selectable to enable longer, more robust tag transmission detections.
- Reader controlled negotiations allow for simple and extremely complex algorithms (with added cost) to aid in the best recovery from transmission errors.

Two widely differing approaches to noise robustness are accommodated in the protocol of this specification. One method of design would be to sacrifice tag read rate in exchange for robustness from large time periods separating data definitions. This is accomplished easily by lengthening the reader to tag training pulses that define data symbols. The other method is to simply and rapidly detect and recover from errors in a data channel. The high tag read rate can be extremely compromised by noise while still providing an application tag read rate equivalent to or higher than other systems implemented without noise considered. Additionally, with a high tag read rate, the probability of random noise impacting a particular tag read is less, simply because the time to read that tag is small. A final note on implementation is that reader designs should take into account end unit cost goals as well as expected noise environment (high background noise vs. random noise patterns) for a best fit for a particular market and application.

7. HUMAN EXPOSURE REGULATIONS

7.1. Various National Standards

American National Standard
IDE standard for safety levels with respect to human exposure to radio frequency electromagnetic fields, 3 kHz to 300 GHz, IEEE C95.1 – 1991, April 1992.

Europe
E. U. Council recommendation of 12 Jul 1999 on the limitation of exposure of the general public to electromagnetic fields (0 Hz to 300 GHz).

EN 50357 Evaluation of human exposure to electromagnetic fields from devices used in electronic article surveillance (EAS), radio frequency identification (RFID) and similar applications.

EN 50364 Limitation of human exposure to electromagnetic fields from devices operating in the frequency range 0 Hz to10 GHz, used in electronic article surveillance (EAS), radio frequency identification (RF I D) and similar applications.

The above two documents appear in the Cenelec index

Australia
Interim Australian/New Zealand standard radio frequency field Part 1: Maximum exposure levels 3 kHz to 300 GHz; AS/NZS 2772.1 (int): 1998.

B. OPERATING CHARACTERISTICS

8. INTRODUCTION

This part B and the following part C of the specification proposes an air interface and anti-collision method utilizing a binary search tree algorithm.

An objective of the specification is to keep the complexity and size of the required tag circuitry as low as possible.

9. SOLUTION FEATURES

The solution produced has the following features.

- It satisfies the design objectives identified in Section 6.2 and Section 6.4.
- It is compatible with mixed tag populations containing any of the so far defined varieties of EPC™, and expected future versions.
- It employs, in a tree walking technology, Context Dependent Protocols (CDP) allowing high throughput in the EPC™ context, in a range of physical applications, and will provide a reading speed not limited by increasing tag numbers.
- It incorporates tag selection for any foreseeable distance along the EPC™ code.
- Tags may be destroyed, i.e. rendered unreadable, on a secure reader command.
- It is adaptable for operation under current US, current European, and expected European regulations.
- It supports the design and manufacture of low cost readers.

C. AIR INTERFACE

10. COMMUNICATION

Communication between the reader and the tag is conducted via an **air interface** described in Sections 12 and 13. The command set and the detailed operation of the anti-collision **operating procedure** is described in Section 13.

11. READER-TO-TAG COMMUNICATION

11.1. Operating Frequency

The tag receives its energizing power, and the instructions that regulate its behavior, from an UHF (860MHz – 930MHz) electromagnetic field produced by a reader.

11.2. Frequency Hopping

Carrier frequency hopping in most jurisdictions is expected to occur, but no assumption that a tag will remain powered after a frequency hop has occurred will be made. If the tag has remained powered through a frequency hop, it will retain its record of whether it has been read. If the tag has not remained powered through a frequency hop, it is allowed but not required to retain its record of whether it has been read. Products that implement a complex structure to perform a memory without power will have a slight advantage over those products that do not, but only in those applications that will exhibit temporary tag power loss through frequency hops. The disadvantage will be evident when adjusting to a new frequency and re-negotiating those tags that were temporarily without power and already negotiated in previous frequencies.

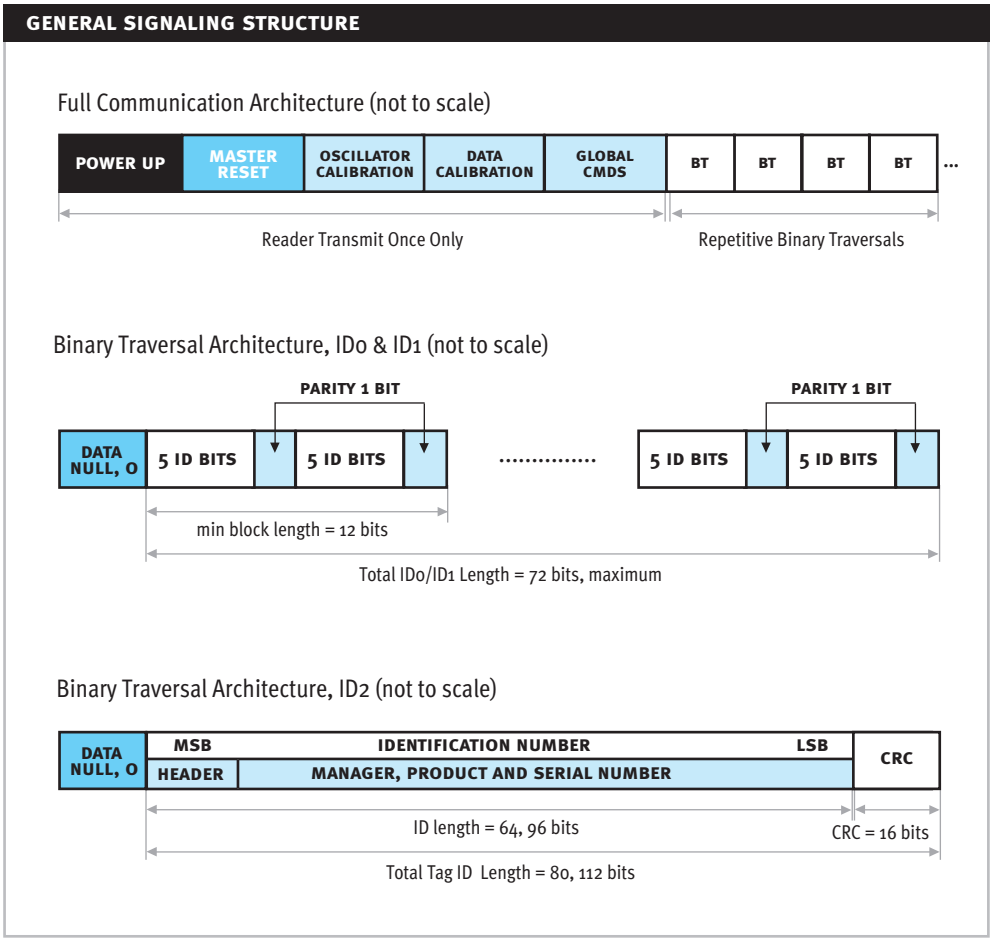
Many jurisdictions require that frequency hops be uncoordinated between different readers, both in respect of the pattern of hopping and the times at which hops occur. Despite these restrictions, it appears to be legal that readers can be shut down for externally controlled and coordinated periods.

11.3. Direct Sequence Spread Spectrum

In jurisdictions that allow broadband direct spreading techniques, the reader and tag may take advantage of this alternate spread spectrum technology. Broadband direct spreading techniques can almost eliminate problems of tags power loss because of multi-path nulls. These readers are not required to frequency hop, thereby eliminating temporary tag power loss issues caused by shifting frequencies. The temporary power loss issue remains for those tags physically moving in space, but the effect may be greatly reduced by a spread spectrum implementation that more evenly distributes RF power in space.

11.4. General Protocol Structure

Figure 5: General Protocol Structure



Proper RFID system design suggests that a reader would be commanded by a host (or timed internally) to address a population of tags, for either a read of all tag Ids or a confirmation read of specific tags. Before and after this polling process, the reader is not emitting RF energy. This allows other readers and other 900 MHz ISM band devices to operate. The negotiation between the reader and tags can be divided into three categories: start up signals, tree traversal negotiations, and command communication.

- **Start up signals** are sent at the beginning of the addressing of the population of tags, and after a frequency hop. During this process, the reader will emit signals to power the tags, calibrate the tag oscillator, and train the tag to interpret the three reader-to-tag data symbols. After the setup, the reader and tags will communicate digitally, the reader with three symbols, and the tags with two symbols.

- **Tree traversal negotiation** is the process by which tags backscatter their singulation bits, which are then acknowledged by the reader, thereby mapping a path through the population binary tree to singulate one product code (tag.)

- **Command Communication** is divided into **global commands** that can be acted upon in the global command state, and singulated commands that can be acted upon in the **singulated command** state. The global commands are a subset of the singulated commands. The global command state is used

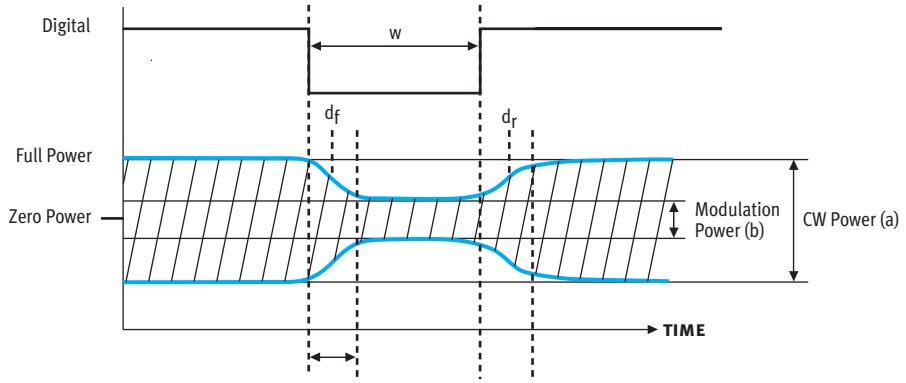
to configure operational parameters, tag states, and manipulate data for sub-sets of or the entire tag population. Global commands are accessible before or during a tree-traversal. Singulated commands are accessible only after an entire tag ID or singulation string has been negotiated, and thus singulated. The differences between the global and singulated command states are in how a tag enters and exits these states, and that some commands, such as the Kill command, may not be accessible from the global command state. Otherwise, the global and singulated commands use the same 8-bit command set.

11.5. Reader-to-tag Data Encoding

Defined in the sections below are the basic parameters for encoding information into the Reader to tag data link, and a description of the elements of a typical reader to tag data stream in the chronological order in which they are likely to occur.

Figure 6

READER TO TAG MODULATION



The reader-to-tag link is defined as the communication channel from the RFID reader to one or many RFID tags. The reader-to-tag link is accomplished through AM pulse-width modulation of the reader transmitter carrier. A typical band-limited AM signal for the reader-to-tag link is presented above in Figure 6. Information is conveyed from the reader in the form of AM pulses. Pulses are defined by the period between a falling and a rising edge in the carrier amplitude. The rise and fall times of the pulse, the pulse width, and the pulse depth, determine the bandwidth of the reader-to-tag link and thus are limited by local regulations. The timing of the reader-to-tag link is defined by:

- The rise and fall (edge transition) timing:
  Falling edge = $y_f - x_f$; Rising edge = $y_r - x_r$ (defined in the chart below)

- Pulse width (W) encodes the data for the reader-to-tag communication:
  $W = x_r - x_f$ (defined in the chart below)

- Modulation dip depth (D):
  $D = (a\text{-}b)/a$ (defined in the chart above.)

Table 2

PARAMETER	MINIMUM	MAXIMUM	UNITS
Edge timing (falling and rising)	.3	10	µs
Modulation dip depth (D)	.3	1	(D)
Pulse width (W); max dip depth	3	15	µs
Pulse width (W); min dip depth	3	60	µs

The wide range between timing minima and maxima reflects the need of the tag circuits to accommodate reader modulations that are compliant with a wide range of world standards. Please also note that longer periods of modulated signal (w in the chart above) would provide extended periods of much lesser power for tag operations under a full (100%) modulation scheme. Therefore, any periods (w) greater than 15 µs will require minimal modulation depth (30%) in order to provide the best powering scenario to the tag.

11.5.1. RF Waveform Examples

The following charts depict actual signal waveforms that should be expected from a reader in multiple situations. These depictions outline the minimum and maximum parameters of modulation depth and data rate in the reader-to-tag link. Each set of figures depicts first the baseband signal and then the RF signal for a bit '0' followed by a bit '1' from the reader. The RF signal waveforms are as follows in order of appearance:

- Fast data rate baseband signal. This should be expected in regions of operation similar in regulation to the United States, with little RF background noise. The bit period is 12.5 ms.
- 100% modulation, fast data rate RF signal. This should be expected in a fast mode reader implementation in the United States and similar regions. This gives maximum data signal into the tag. 100% modulation is useful at short range with inexpensive readers.
- 20% modulation, fast data rate RF signal. The smaller modulation depth provides more average power to the RFID tags. 20% modulation is useful at long range, and in some bandwidth limited regions of operation.
- Slow data rate baseband signal. The bit period is 62.5 ms.
- 20% modulation, slow data rate RF signal. This characteristic is to be expected in regions of operation such as in Europe. Minimal power and minimal bandwidth are available in regulations in this area of the world.

Figure 7

READER BASEBAND SIGNAL, FAST DATA RATE

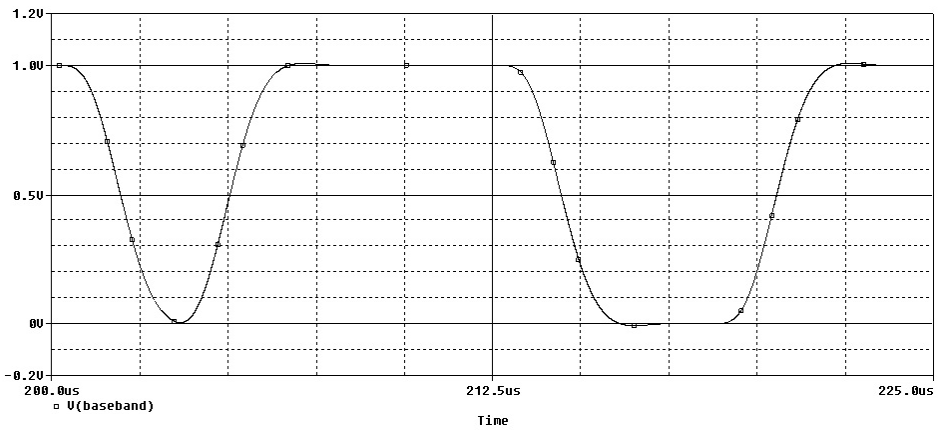


Figure 8

READER RF SIGNAL, 100% MODULATION, FAST DATA RATE

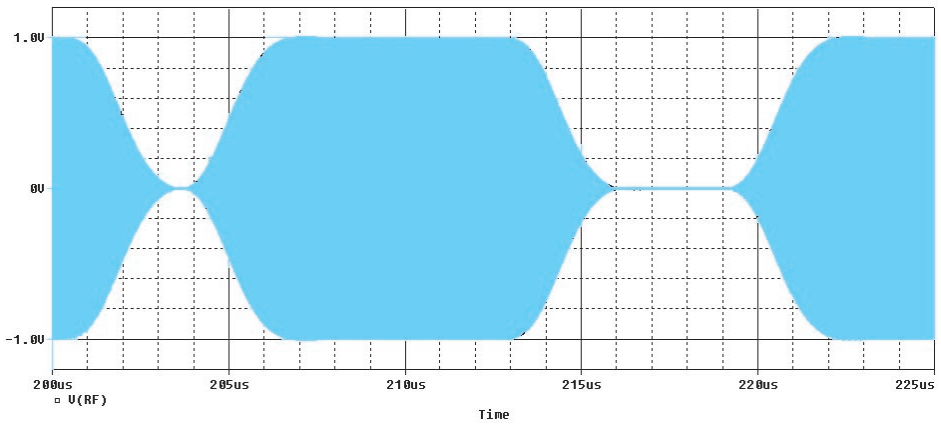


Figure 9

READER RF SIGNAL, 20% MODULATION, FAST DATA RATE

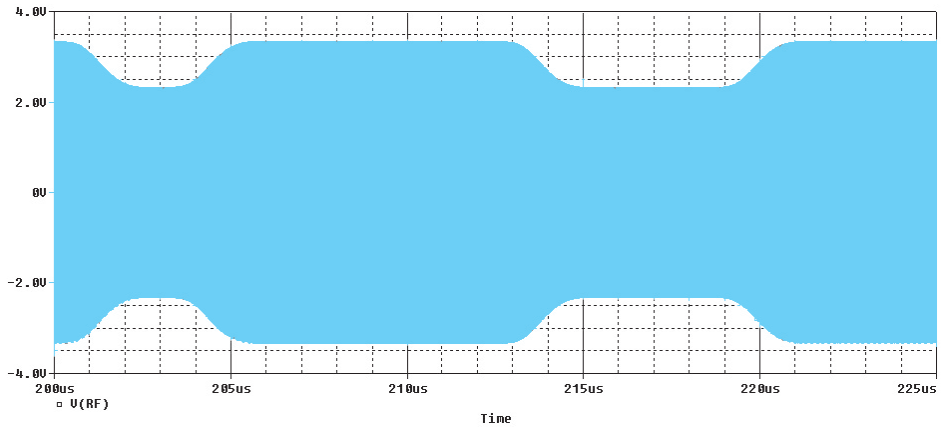


Figure 10

READER BASEBAND SIGNAL, SLOW DATA RATE

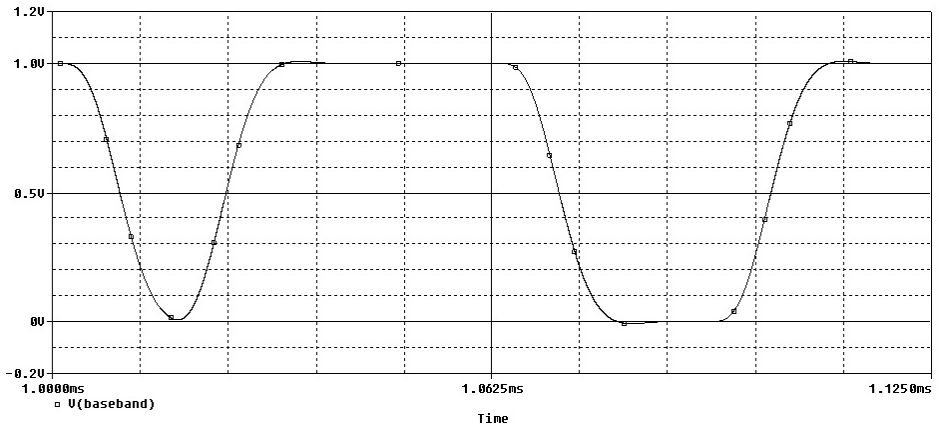
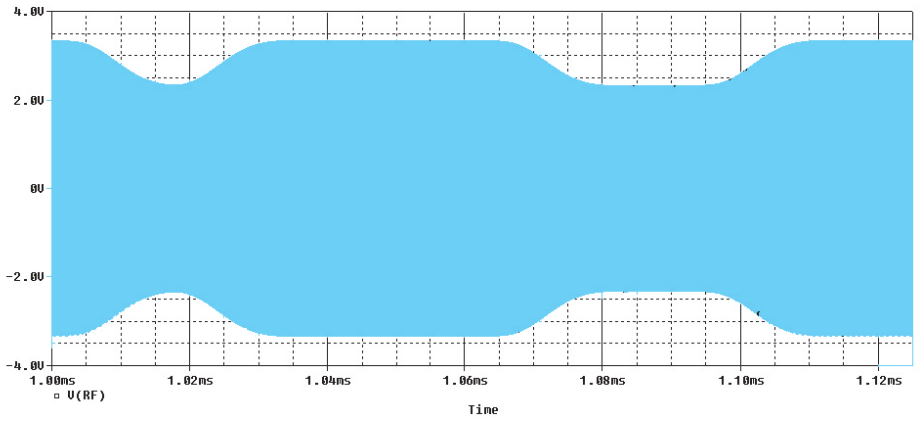


Figure 11

READER RF SIGNAL, 20% MODULATION, SLOW DATA RATE

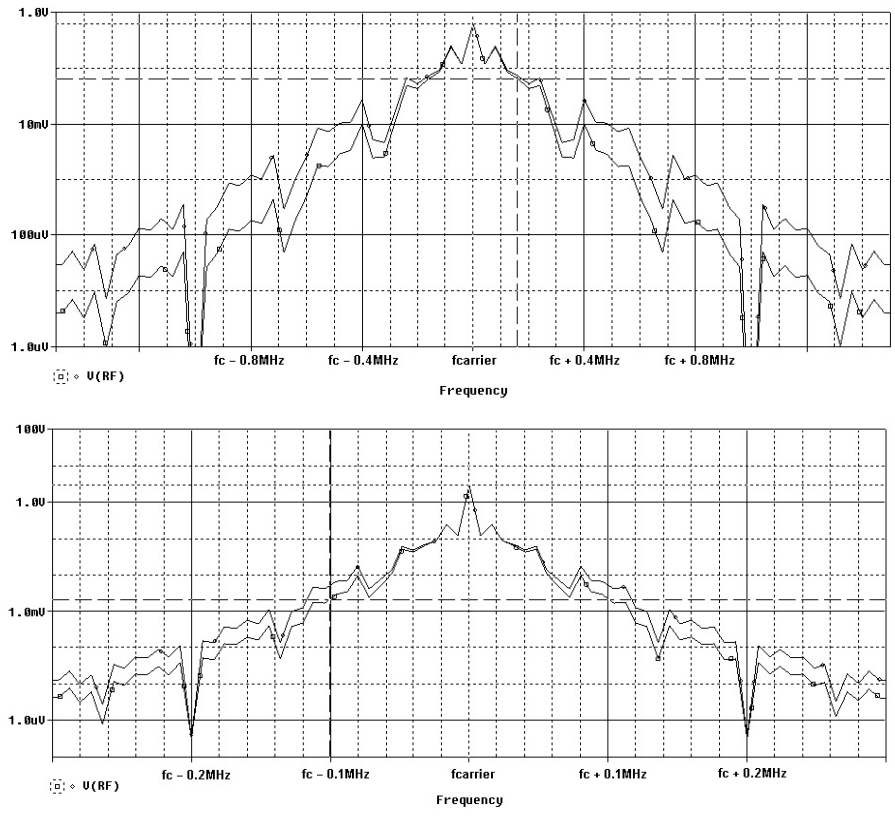


11.5.2. RF Emissions Examples

The following charts are only examples of expected emissions from traditional style readers for the previous sections waveforms, specifically corresponding to Figure 8, and Figure 11. Figure 12 provides an example of a simple US reader design, and Figure13 provides an example of a simple European reader design. In each chart, a horizontal dotted line indicates the level defined by the regulatory body that must be within the channel also defined by the regulatory body.

Figure 12, Figure 13

READER EMISSIONS LIKELY CORRESPONDING TO FIGURE 8 WAVEFORM



11.5.3. RF Power Up

RFID reader designs may vary widely in the timing and methods to power-up an amplifier and get ready for data communications. To help the tag electronics to induce an internal power-on-reset, the RF power should ramp up as quickly as possible while still obeying local bandwidth regulation.

11.5.4. Reset

A reset signal is defined as a signal that appears to a calibrated chip as a 1024 cycles (timer overflow) equating to 465msec of uninterrupted (no data modulation) RF carrier from the reader. Several important considerations need to be accounted for in a reader transmitted reset signal so as to guarantee a reset detection within all chips, especially in the original, un-calibrated state. In the following discussion, we must distinguish between the separate concepts of reset and power on reset

- Power on reset for the chip is another condition that is to be considered when the reader emits CW for a reset signal. The power on reset time allotted will be a maximum of 200msec.

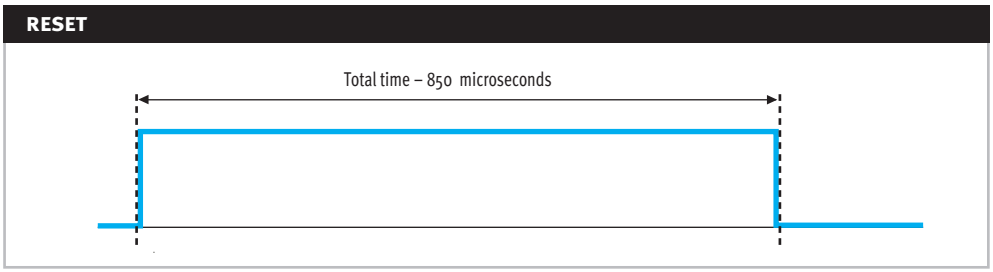
- A frequency hop or a change for some other reason in RF field power absorbed by the tag may cause a general loss of the internal chip assessment of the data signal level. The chip will be allowed to attempt to restore its assessment of the data signal level and will assume the current signal level is at a 100% (CW).

- The total allotted period of time for the above two operations to occur is a maximum of 200msec. It is important to note that a specific tag may either enter the power on reset condition (first bullet) or the data re-assessment period (second bullet), but not both simultaneously.

- Finally, an un-calibrated oscillator on chip requires an accommodation of a ± 30% tolerance of the timing of the reset signal. The range that tags may detect a reset signal will then be from 350msec to 650msec in time. Thus, in addition to the considerations mentioned above, the reader must emit a CW signal for at least 650msec to ensure all tags properly decode the signal to a reset condition.

The reader will therefore be required to emit an 850msec CW signal after reaching full power output to ensure all tags will decode the reset signal properly. The chip shall properly detect this minimum time as a reset. Any signal that may be detected as longer is still to be considered a reset signal by the chip. In this case, the chip will resume processing the first piece of information whenever the next data edge (falling) is received.

Figure 14



11.5.5. Oscillator Calibration Signals

Oscillator calibration accomplishes the transfer of the precise reader time base to the tag. The oscillator calibration is a series of eight pulses that the tag Successive Approximation Register (SAR) uses to adjust the trim value of the oscillator. At the end of each of the eight SAR calibration pulses is a separation pulse that allows the tag to adjust its oscillator frequency and ready itself for the next SAR calibration pulse. The SAR directly controls the 2.2MHz system clock by means of eight binary weighted switched elements

that collectively tune the operating frequency over a range of ±50% to a final theoretical accuracy of ±.391% (±50%/2^7), which does not factor in the likely noise in the RF communication channel. For the purpose of this discussion, let us assume that the SAR starts out with the MSB set (10000000), and this puts the oscillator at the middle of its tuning range. Setting a SAR bit increases the oscillator frequency. The tag measures the first SAR calibration pulse against a counter being clocked by the tag oscillator. The counter value is latched on the falling edge of the SAR pulse cycle. If the latched counter value is 256 or greater then the oscillator frequency is high and bit #7 (0 to 7) of the SAR is cleared, else bit 7 remains set. The calibration pulse period was chosen such that the SAR evaluation of "›255 or not" can be made by simply testing the eighth bit of the latched counter value. The tag then sets bit #6 of the SAR and waits for the next SAR pulse falling edge to signify the start of the next calibration pulse. This process is repeated 8 times, each time making the choice to set or reset the next bit in the SAR, thus each time halving the oscillator tolerance. The main source of error in this process is jitter in the calibration pulses caused by noise in the environment. Including harsh environment noise estimates and the impact of jitter in the SAR pulses, the tag frequency tolerance after calibration will be no worse than ±1.25 %.

The diagram of Figure 15 below illustrates the waveform of the eight SAR calibration pulses, each of 6 µs low, and 116 µs total period, and also illustrates the separation pulses of 6 µs low and 6 µs high period.

Figure 15

OSCILLATOR CALIBRATION SIGNALS

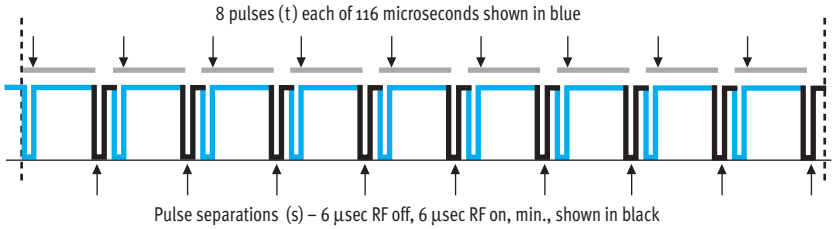


Table 3

PARAMETER	MINIMUM	TYPICAL	MAXIMUM	UNITS
Calibration Pulse (t)	115	116	117	µs
Separation Widths (s)	6	---	15	µs

After a reset, the oscillator calibration pulses must always be supplied.

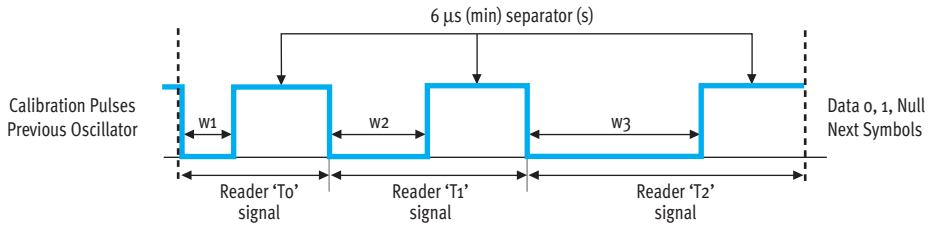
11.5.6. Data Symbol Calibration Signals

Data Symbol Calibration signals are a sequence of three pulse cycles that inform the tag how to interpret the reader-to-tag link symbols 0, 1, and Null, as well as the time point where backscatter should stop. Like the oscillator calibration signals, the data calibration signals must be given following each reset and subsequent SAR pulses. There is no condition after a reset in which the data calibration signals do not follow the oscillator calibration pulses.

Timing is defined for each of the three data symbol calibration pulses by the interval between the falling and rising edge, i.e., the low periods w1, w2, w3 shown in Figure 16. Each of the three data sync pulses are separated by a high going pulse of period (s) to give the tag time to latch the data and prepare for the next low going data sync pulse. Each of pulse intervals w1, w2, and w3 is required to be in increasing widths. Each of these, as well as the separator periods (s), is constrained by local regulatory requirements. However, these all have operational minima and maxima defined in the table further below.

Figure 16

DATA SYMBOL CALIBRATION SIGNALS



We show the Data Symbol Calibration Signals again in Figure 17 along with representative values of the data transmission signals that will be interpreted against the intervals w1, w2 and w3 defined in the data calibration process. It is important to note that w1 conveys timing of a decision point midway between the intended data 0 and data 1 symbols. Similarly, w2 conveys timing of a decision point midway between the intended data 1 and data null symbols. These timing signals (w1, w2) shall never equal any intended data symbol length, as these are not sample data symbols.

Figure 17

DATA CALIBRATION AND DATA TRANSMISSION SIGNALS

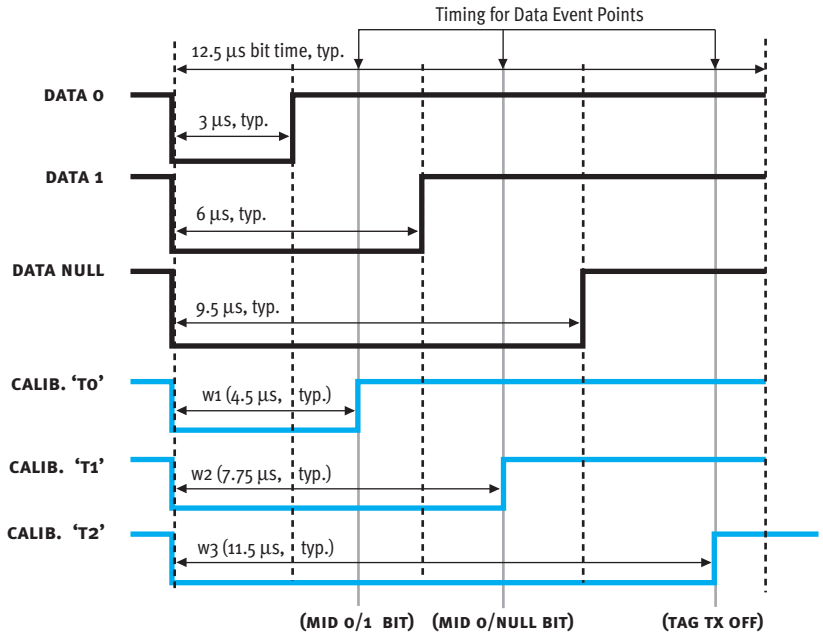


Table 4: Note: Minimum dip depth required for implementations of w3 > 15 µs

PARAMETER	MINIMUM	OTHER CONSTRAINT	MAXIMUM	UNITS
w1	4.5		30	µs
w2	6.5	> w1	40	µs
w3	9.5	> w2	55	µs
s (separation)	6	---	30	µs

11.6. Reader-to-tag Data Symbols

For clarity, we have provided in Figure 17 an example of how the data calibration signals are used to interpret the data from the reader.

– Data ‘0/1’ timing w1 is intended to be the mid point between the rising edges of a data ‘0’ symbol and a data ‘1’ symbol. On the falling edge of a reader data symbol, the tag counter is cleared and starts counting. When the timer equals the latched w1 value, a single bit flag, called the To flag, is set. If the data line goes high before the To flag is set then the data is interpreted as a data ‘0’ symbol, and the tag responds appropriately. If the reader pulse has not gone high after the w1 interval, then the symbol may be a ‘1’ or a ‘null’. That determination will be made at or before the w2 interval. The w1 interval for North American operation is typically 4.5 µs.

– Data ‘1/null’ timing w2 is intended to be the mid point between the rising edges of a data ‘1’ symbol and a data ‘null’ symbol. On the falling edge of a reader data symbol, the tag counter is cleared and starts counting. When the timer equals the latched w2 value, a single bit flag, called the T1 flag, is set. If the data line goes high and the To flag is set (previous paragraph) but the T1 flag is not set then the data is interpreted as a data ‘1’ symbol, and the tag responds appropriately. If the data line goes high and both the To and T1 flags are set then the data is interpreted as a data ‘null’ symbol. The w2 interval for North American operation is typically 7.75 µs. It is important to note that a data null symbol is defined as a period longer than the w2 interval. Thus, a data null symbol may also be longer than the w3 period as well. The w3 period (next) defines an end of backscatter that is not applicable to a data ‘null’ symbol.

– Tag backscatter end point w3 defines the point at which the tag must stop sending data back to the reader, and prepare for the falling edge of the next reader data symbol. The w3 interval is typically 11.5 µs for North American operation.

11.6.1. Data Symbol ‘0’

Figure 18

READER DATA SYMBOL 0

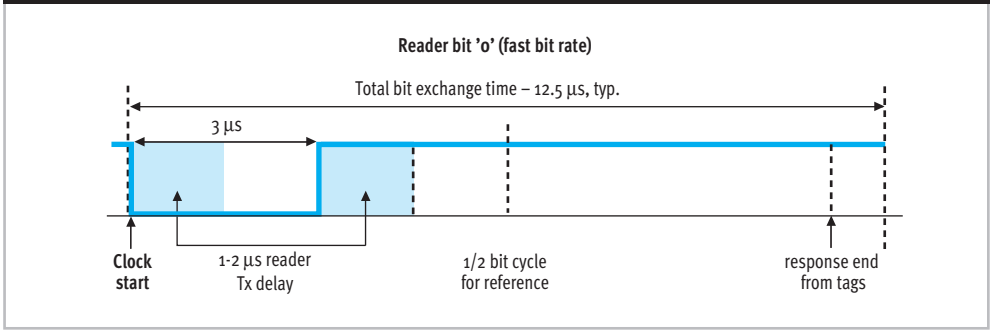


Table 5: Note: Minimum dip depth required for implementations of modulation > 15 µs

PARAMETER	MINIMUM	RELATIVE	MAXIMUM	UNITS
Bit ‘0’ modulation	3.5	< w1 – 1	25	µs
Bit ‘0’ total width	12.5	> w3 + 1	65	µs

11.6.2. Data Symbol '1'

Figure 19

READER DATA SYMBOL 1

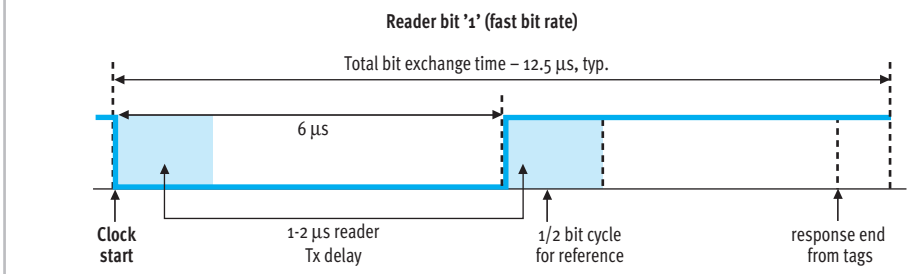


Table 6: Note: Minimum dip depth required for implementations of modulation > 15 µs

PARAMETER	MINIMUM	RELATIVE	MAXIMUM	UNITS
Bit '1' modulation	6.0	> W1 + 1 < W2 + 1	35	µs
Bit '1' total width	12.5	> W3 + 1	65	µs

11.6.3. Data Symbol 'Null'

Figure 20

READER DATA SYMBOL NULL

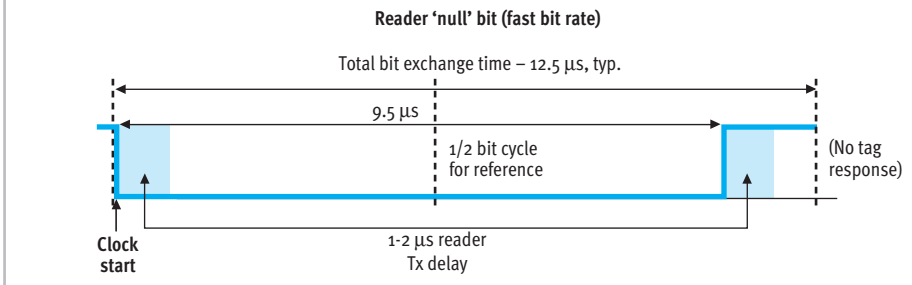


Table 7: Note: Minimum dip depth required for implementations of modulation > 15 µs

PARAMETER	MINIMUM	RELATIVE	MAXIMUM	UNITS
Bit 'null' modulation	7.75	> W2 +1	45	µs
Bit 'null' total width	12.5		65	µs

12. TAG-TO-READER COMMUNICATION

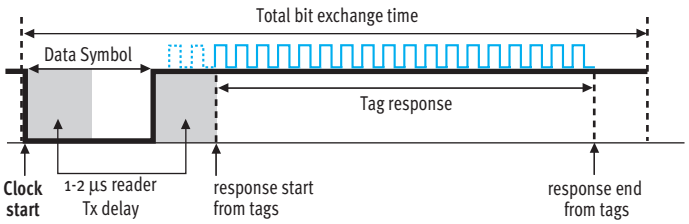
12.1. Generation of Reply

The tag replies are generated by the mechanism of RF backscatter, using two sub-carrier tones. Two operational modes select different frequencies for sub-carrier tones which is selectable by the reader utilizing the command SetRegionOfOperation defined in section 13.4.4.

12.2. Return Link (Backscatter) Data Encoding

Figure 21

TAG BACKSCATTER ON READER DATA SYMBOL



Data is sent back to the reader through modulated backscatter. Modulated backscatter requires no transmission power and is accomplished by alternately changing the chip port impedance, and thus the reflectivity of the antenna. The backscatter modulation is characterized by the induced voltage drop as seen at the chip port. The antenna modulation shall maximally be 30% drop in RF port voltage over the full input power range to allow the chip to maintain power for operation.

Backscatter starts at the rising edge of a bit period on the CW portion of a reader-to-tag bit 0, and bit 1 symbol. No backscatter occurs on a 'null' symbol. The backscatter ends at the point defined by the training interval w3 (see Figures 15 and 16). The interval w3 may be adjusted to accommodate different data rates and reader architectures.

Table 8

PARAMETER	MINIMUM	RELATIVE	MAXIMUM	UNITS
Tag backscatter loading	10%		30%	
Bit '0' frequency, Region 2		2.2		MHz
Bit '1' frequency, Region 2		3.3		MHz
Bit '0' frequency, Region 1		1.1		MHz
Bit '1' frequency, Region 1		1.65		MHz
Frequency Tolerance	–2.5%	---	+2.5%	%

12.2.1. Bi-Directional Communication

After the SAR oscillator calibration and data calibration pulses are sent, both the reader-to-tag and tag-to-reader communication channels are fully defined and all subsequent communications are sent using the three reader-to-tag symbols and the two tag-to-reader tones. With this basic alphabet, the reader

may communicate global commands, singulate tags, or issue singulated commands. The illustrations below show the full relationships between the reader-to-tag data signals and the backscatter from the tag, in all combinations during digital data exchange. This illustration is based on a typical implementation running at the quickest possible timing rate as described in previous sections. An example depicting the slowest possible timing rate is provided as the last diagram below. It is key to note that the timing is virtually analog and can be adjusted by the reader at any point by issuing a reset and calibration sequence.

The delays shown are the result of a combination of edge detection error and filter delays in the reader conforming to U.S. regulations for a frequency hopping reader design. Delays are expected to be more severe in other operational regions. It is the responsibility of the reader to foresee and compensate for these delays in the calculation of the T2 data calibration signal. The T2 data calibration signal will instruct the tag as to the length of backscatter. Many factors in reader design, including receiver filter delays, may play a role in determining the best-case data timing.

Shown below are the four bit combinations possible – note that there is no backscatter on the data symbol 'null', or any other unrecognized symbol. Combinations are:

– Reader 0, Tag 0
– Reader 0, Tag 1
– Reader 1, Tag 0
– Reader 1, Tag 1

Figure 22

TAG 0 ON READER 0, FAST BIT RATE EXAMPLE

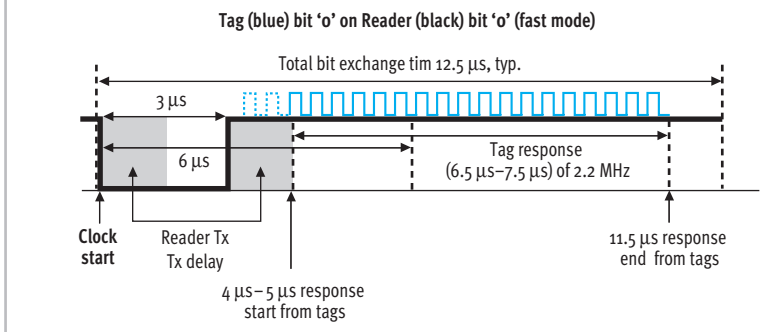


Figure 23

TAG 1 ON READER 0, FAST BIT RATE EXAMPLE

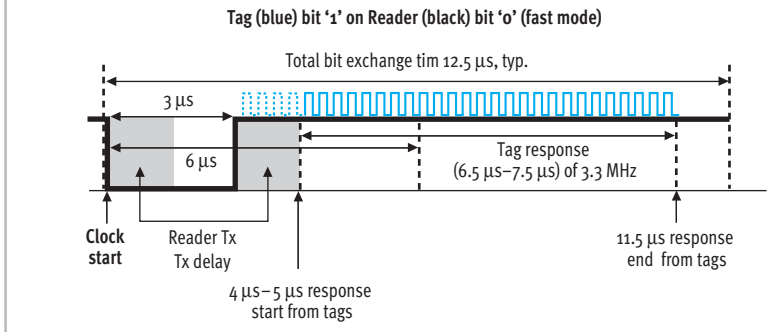


Figure 24

TAG 0 ON READER 1, FAST BIT RATE EXAMPLE

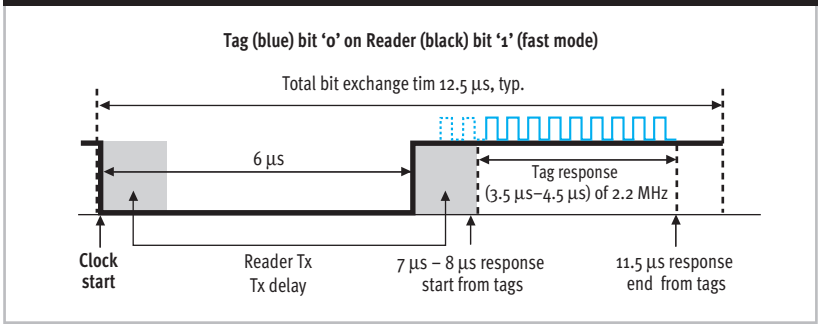


Figure 25

TAG 1 ON READER 1, FAST BIT RATE EXAMPLE

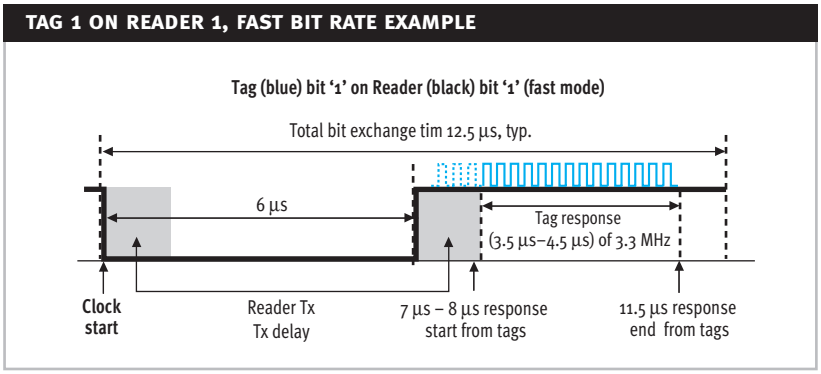
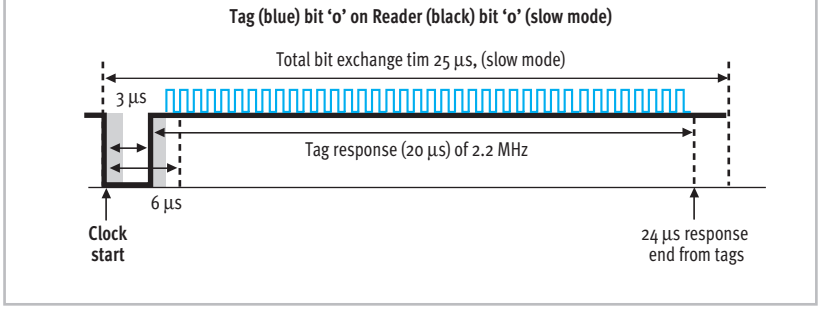


Figure 26

TAG 0 ON READER 0, SLOW BIT RATE EXAMPLE

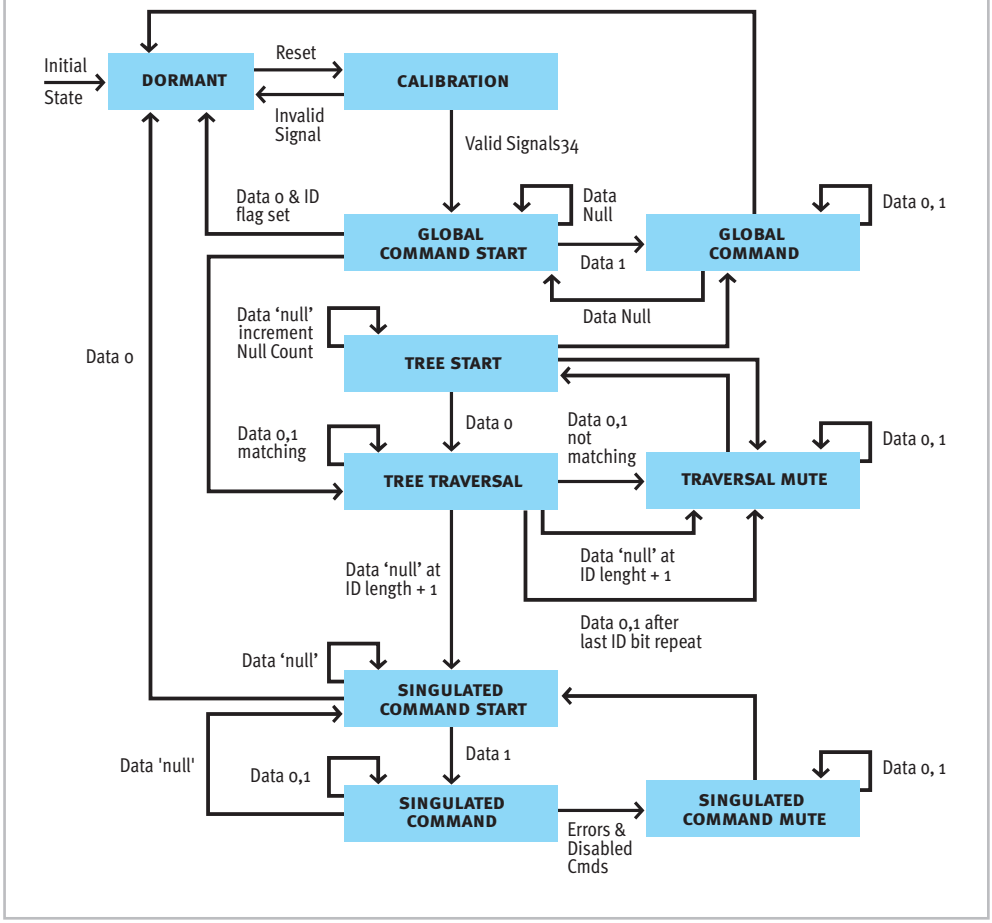


13. TAG STATE MACHINE ANTI-COLLISION AND COMMAND PROTOCOL

13.1. Tag State Machine

Figure 27

TAG PROTOCOL STATE DIAGRAM



The easiest way to detail the tag operation is to describe the hierarchy of system states and events, and the data that are passed between the tag and reader. A state flow chart shown in Figure 26 describes the general state evolution. Detailed descriptions are presented in the following paragraphs.

13.2. Tag State Definition

Tag states are all mutually exclusive, thus if any one state is true, all other states are false.

– **Dormant State.** While the tag is in the Dormant State, the tag is largely inactive, and is waiting for a Reset from the reader. Any low power condition, in which the logic, oscillator or associated timings may be compromised, shall force the circuit into the Dormant State. This is to eliminate any misinterpretation of reader data and/or any spurious backscatter from malfunctioning tags. The Dormant State is the default state at tag power-up. Note that the Dormant State is also used as the holding state for keeping quiet tags that have already been read.

– **Calibration State.** A Reset always initiates calibration. Calibration encompasses the eight oscillator calibration pulses, and three data calibration pulses that are detailed in Sections 12.4.4. and 12.4.5. Upon completion of calibration, the tag enters the Global Command Start State. If at any time an out of-range calibration pulse length or signal is detected, the tag is to put itself into the Dormant State automatically.

– **Global Command Start State**. This state is immediately entered upon the receipt of the last data calibration pulse. The Global Command Start State is similar to the Singulated Command Start State, except that the entry and exit conditions are different. From the Global Command Start State, if a data '1' is received, the tag will enter the Global Command State. A null symbol is ignored. If a data '0' is received this will place the tag into one of two states depending upon the identified (ID) flag status. If the ID flag is set (this tag has previously been read), the '0' bit will place the tag back into the Dormant State. If the ID flag is cleared, the tag will proceed into the Tree Traversal State. Note that on power-up, the ID flag is always cleared, unless a persistent memory for the ID flag has been implemented. For this case, the ResetIDFlag command will be useful. Note that how the ID flag gets set is detailed under the singulated command start state.

– **Global Command State.** This command state implements features and functions on the tag before any tags have been singulated via a full tree traversal. This state accepts and processes 8-bit commands from the reader. The Global Command State may be entered either from the Global Command Start State at the receipt of a data '1', or from the Tree Start State at the receipt of exactly 2 data nulls in succession followed immediately by a data '1'. The latter method allows a subset of the active population of tags to be addressed by global commands. Global and singulated commands are discussed in detail in Section 13.3 and following. Some commands accessible in the Singulated Command State, such as the Kill command, will be disabled in the Global Command State. If a command is received that is unimplemented, disabled, or if a communication error is detected, the tag will revert to the Dormant State. This is done to remove tags from the tree traversal process that cannot support a particular command, such as an optional command. This action, however, does not set or clear the ID flag, and subsequent Reset signals can recover the tags that were put into the Dormant State. A data null will terminate a command and return the tags back into the Global Command Start State.

– **Tree Start State.** This mode is entered from the Traversal Mute State after the receipt of a data 'null'. On entering the Tree Start State, the count of consecutive data nulls received while in this state is reset to zero. Since a data '0' and data '1' will exit this state, a simple counter of any data null will suffice as a successive null counter. This is the only function performed on receipt of a data 'null'. Otherwise, a one-bit header is then expected from the reader. A '0' puts the tag into the Tree Traversal State. A '1' puts the tag into either typically the Traversal Mute State, or specifically after the null counter equals 2 (not more, not less), a data '1' will put the tag into a Global Command Mode. The null counter comparison allows a reader to address a subset of an active population of tags; the subset defined by matching bits of a partial singulation string. For clarification, a reader may send the bits of a desired partial singulation string to the tag population. The active population will, after receipt of the partial singulation string, either remain in the Tree Traversal State, or if any bits are not matching, will transition into the Traversal Mute State. A reader, at this point, may issue 4 consecutive nulls. For those tags not matching the partial singulation string, the null counter will equal 3. For those tags that do match the partial singulation string, the null counter will equal 2. The reader then has the option at this specific instance of issuing a data '1'. A data '1' will place all tags matching the partial singulation string into a Global Command State, and all tags not matching into the Traversal Mute State. In a similar fashion, a reader can optionally issue only 3 consecutive nulls followed by a data '1'. In this case, all tags that do NOT match a partial singulation string go into the Global Command State, while tags that do match will transition into the Traversal Mute State. Tags that transition into the Traversal Mute State are not intended to be addressed by Global Commands, and these tags will harmlessly transition to and from Tree Start and Traversal Mute States.

– **Tree Traversal State.** This state is entered from the Tree Start State with a '0' bit. The tree traversal mode allows the reader to simultaneously negotiate with many tags to singulate a tag with one of the ID pages. Upon entering the Tree Traversal State, the tag shifts out its MSB to the backscatter modulator. The tag then receives the next data bit from the reader. If the reader sends a bit that matches the previous bit backscattered by the tag, then the tag shifts out the next bit in its ID to the backscatter modulator. If the tag receives a bit from the reader that does not match the last bit shifted out, the tag goes into the Traversal Mute State where it waits for a null. On the last bit of the ID, which is a CRC bit for ID2, or on a boundary bit, which is every 12th bit for ID0 and ID1 (see Figure 27 & 28 below for details), the tag backscatters this bit and waits for a confirmation from the reader. Upon this confirmation from the reader, the tag then backscatters out either a) the last data bit again in the case of ID2, or b) the next available bit in the ID sequence in the case of ID0 and ID1. After the shift out of this next bit past the boundary bit, if the tag receives an additional '0' or '1' from the reader, the tag in ID2 traversal only is put into the Traversal Mute State. If the tag receives a 'null' after this next bit past the boundary bit, then the tag has shifted out and has had confirmed all its necessary ID bits, and is put into the Singulated Command Start State. The reader will determine the appropriate point to send a data null by decoding the either the first bits received from the tag as in the ID2 case, or by count as in the ID0 & ID1 case. The tag shall not accept any other occurrence of a data null for transition into the Singulated Command Start State. If the tag receives such a misplaced data null, it will immediately transition into the Traversal Mute State. Note that in a sensible protocol, after all ID bits have been negotiated in the Tree Traversal State, the tag ID parity or CRC is checked by the reader. If there was a parity or CRC error, an additional dummy '0' or '1'can be sent to the tag while still in the Tree Traversal State to force it back into the Traversal Mute State to be read again. The reader may also proceed through the Singulated Command Start State into the Singulated Command State where the tag can be commanded to re-send out its ID or singulation string as a confirmation. From the Singulated Command State, the tag can be put into the Dormant State via the Singulated Command Start State using a data null followed by a data zero, or the tag can be put into the Traversal Mute State by the ForceMute command, the latter path not being shown in the state diagram.

– **Traversal Mute State.** In the Traversal Mute State, the tag receives data from the reader but gives no response until a data 'null' puts the tag into the Tree Start State. In the state transition from Traversal Mute State into Tree Start State, the tag shall reset a count of continuous data 'nulls'. Refer to the Tree Start State for further detail of the null counter use.

– **Singulated Command Start State.** This state is entered from the Tree Traversal State after receiving a 'null' following the confirmation of the last ID bit. From the Singulated Command Start State, if a '0' is received, the tag has been confirmed read, and will go into the Dormant State. During this process, the tag will set the identified (ID) flag as a memory of a confirmed read. A 'null' symbol is ignored. Finally, if a '1' is received, the tag will enter the Singulated Command State.

– **Singulated Command State.** This state implements features and functions on the tag, after a tag has been singulated, by processing 8-bit commands from the reader. These commands are discussed in detail in Section 13.3 and following. If a command is received that is unimplemented, disabled, or if a communication error is detected, the tag will be put into the Singulated Command Mute State.

– **Singulated Command Mute State.** Similar in function to the Traversal Mute State, the tag, while in this state, will not backscatter, but will look for a 'null' to exit this state into the Singulated Command Start State. This state is to be entered when a tag cannot perform a command. Several conditions may force this state: an unimplemented command, a request for a disabled command, or a detected parity error.

13.3. Tag Command Implementation

Tag commands, global and singulated, implement operations where an 8-bit command with a parity bit and an N-bit argument, with error checking of the argument as defined below, initiate a potentially complex operation or computation. Command use is detailed below:

– For singulated commands, the tag is first singulated by the tree traversal, including the repeated backscatter of the last ID bit. On completion of a full tag singulation, the reader issues the sequence 'null, 1'. During the CW period of the '1', the tag will not backscatter. This enables the reader to determine if any tags have been left in the Tree Traversal State.

– The reader then issues an 8-bit command plus a parity bit, most significant bit first. On each bit from the reader, the tag acknowledges by backscattering the same bit. For example, if the reader sends a data '0', the tag should backscatter a data '0'.

– For singulated commands, if the tag detects a parity error in the 8-bit command, or the command received is not supported, then the tag goes into the Singulated Command Mute State, where it waits for a 'null, 1' to start a new command. For global commands, if the tag detects a parity error in the 8 bit command, or the command received is not supported, then the tag goes into the Dormant State.

– After the 8-bits of command plus one bit parity, the reader will follow with an undefined number of data bits that are a parameter to the command just sent. Backscatter on these data bits is also REQUIRED for each bit, as either confirmation or as the method of getting specific tag data to the reader. The commands, their arguments, and returned data are listed in Section 13.4 in this document.

– If the reader does not see the correct bit acknowledged by the tag (a reader-to-tag or tag-to-reader bit error), the command can be aborted and started again with the sequence 'null, 1'.

13.4. Command Definitions

13.4.1. Command Summary

Commands are 8-bit numbers, followed by a one-bit parity check. The 8-bit command space is broken into categories of commands. The range of commands is as follows in decimal format:

– 163: Mandatory Global commands
– 64-127: Mandatory Singulated only commands
– 128-191: Optional commands
– 192-254: Test commands

Mandatory commands, listed below, are required by both readers and tags to conform to this specification. Installed applications may not necessarily make use of a particular command for a number of reasons. For instance, in Region 2 operations, it is not necessary to use the SetRegionOfOperation command, even though it is a requirement for readers and tags to implement and make available that command.

No optional commands have been defined in this specification, but as the specification evolves, some might be. Optional commands, if defined, may be implemented by a manufacturer. If a manufacturer wishes to implement such a function, it must be implemented exactly as specified in the appropriate document.

Test commands are reserved commands for manufacturers to implement test features and functions. These commands enable a manufacturer to provide specific test functionality for their tags. All test

commands must be disabled prior to the use of a tag. No tag in use will respond to a test command. In no case is it allowed to have a test command replace a mandatory or optional command. The mandatory or optional functions are to be implemented exactly as indicated in this specification.

Global and Singulated Mandatory Commands (with parity bit):

ResetIDFlag	{00 000001 1} (reset all identified flags to not read)
SetNegotiationPage	{00 000010 1} {argument} (choose string for negotiation)
SetRegionOfOperation	{00 000011 0} {argument} (set backscatter parameters)
ForceDormant	{00 000100 1} (immediately enter Dormant State)
ForceMute	{00 000101 0} (immediately enter Mute State)
Read	{00 000110 0} {argument} (read new data from an indexed page)

Singulated Only Mandatory Commands (with parity bit):

Kill	{01 111111 1} {argument} (Permanently disable tag)

13.4.2. ResetIDFlag (binary '00 000001 1')

The ResetIDFlag command has no argument and returns nothing. In a sensible protocol the command set is used to ensure that the ID flag is set only after a successful tag singulation and CRC check. Setting this flag signifies that this tag has already been read, and need not participate in future negotiations. How the flag is set has been described in the discussion of the Singulated Command Start State. The ResetIDFlag command clears the ID flag, thus allowing tags that were previously read to be put into dormant state and read again.

13.4.3. SetNegotiationPage (binary '00 000010 1')

The SetNegotiationPage has a 4-bit page argument, and no return data. This command sets which data page shall be used to perform the binary tree traversal. Additional pages are currently undefined, but a maximum of 16 are allowed for in the page argument. The default page for tree negotiation is page '0010'(ID2). Currently implemented pages are:

- Page '0000' Dynamically random derived within tag (ID0)
- Page '0001' Repeatable pseudo random within tag (ID1)
- Page '0010' User data ID2 (EPC™)

13.4.4. SetRegionOfOperation (binary '00 000011 0')

The SetRegionOfOperation command has a 4-bit region argument, and no returned data. The 4-bit region arguments are defined as:

- '0001' : Region 1 current operations mode
- '0010' : Region 2 current operations mode (default)
- The remaining numbers are reserved for future regions of operation.

13.4.5. ForceDormant (binary '00 000100 1')

The ForceDormant command has no arguments. The chip is to immediately enter into the Dormant State following the receipt of the 9 bits of command and parity.

13.4.6. ForceMute (binary '00 000101 0')

The ForceMute command has no arguments. The chip is to immediately enter into the Traversal Mute state following the receipt of the 9 bits of command and parity.

13.4.7. Read (binary '00 000110 0')

The Read command is essential in secure tag negotiations. In secure modes of operation, readers will singulate tags based upon ID0 or ID1 registers, which do not contain useful application information. The readers will be required to use the Read command to extract the EPC™ application information in the ID2 register. The Read command has a 4-bit page argument and subsequently returns the content of that page. The only valid argument is 0010 referring to ID2 containing the EPC™. All data (4 bits of page data plus up to 112 bits of ID) will be sent via backscatter to the reader. The reader first sends 4 bits of page information that shall be echoed back and decoded by the chip. Following these 4 bits, the reader sends out dummy bit symbols (1 or 0), and the tag backscatters the page data on the CW portion of each symbol, regardless as to the value of the bit 0 or 1. A data null will be recognized as a signal to terminate this command. In this way, the tag and reader maintain synchronization, while no useable tag ID information is re-broadcast by the reader. Note that 0000 and 0001 (referring to ID0, ID1) are not valid arguments.

13.4.8. Kill (binary '01 111111 1')

This function will permanently disable this tag from any backscatter whatsoever. This command is intended to ensure the tag can no longer be detected by either conforming data bits or any other non-conforming communications in the form of backscatter modulation. The kill command carries as an argument a tag-specific authorization code of 24 bits that has been programmed into tag memory at time of manufacture. The tag shall backscatter each bit received from the reader for all 24 bits, regardless of whether these bits match the authorization code. Following the issuance of these 24 bits, the reader shall issue an additional data bit '1'. The tag shall reply to this bit with a either a data bit '0' indicating no match to the authorization code, or a data bit '1' indicating a proper authorization code match. If there is no match, the tag shall immediately enter the Dormant State. The reader, upon receipt of a tag data bit 0 (no match), shall issue a data null to terminate this attempt and carry on appropriately. However, upon receipt of a tag data bit 1 (authorization code match), the reader shall issue up to 500 elongated data '0's (standard data 0 modulation, but an overall bit period of 300 msec)to allow for implementation of the kill feature on the tag. During this series of pulses, the tag is afforded time and power to implement the kill function. When completed, the tag shall reply with a backscatter defined as a data '1' for successful implementation or a data '0' for an unsuccessful implementation possibly arising from to little power available. When the kill function is complete, the backscatter reply is to occur on each elongated data '0' as if the pulse is a standard data '0' from the reader until a data null is received. When the reader receives this backscatter indicating completion of processing, the reader will immediately issue a data null to terminate the command. After the Kill command has been issued (with a matching kill passcode) and correctly executed by a tag, the tag will never again modulate its backscatter.

IMPORTANT NOTE

Other than as described n the previous paragraph, the tag is specifically not allowed under any command, except test commands, to transmit this authorization code or any portion thereof. Violation of this note is not allowed under adherence to this specification.

13.5. Frequency Hopping Procedures

Due to FCC requirements, the reader will need to change frequency of operation on a periodic basis ranging from 50 to 400 milliseconds. During this period, the reader will follow this procedure:

- Place the reader in a CW state (no data.)
- Adjust frequency near to (within 1MHz) a new frequency over a 30-microsecond period. This may create rapid changes in RF energy as detected by tags that are forced into and out of RF nulls as a result of changing.

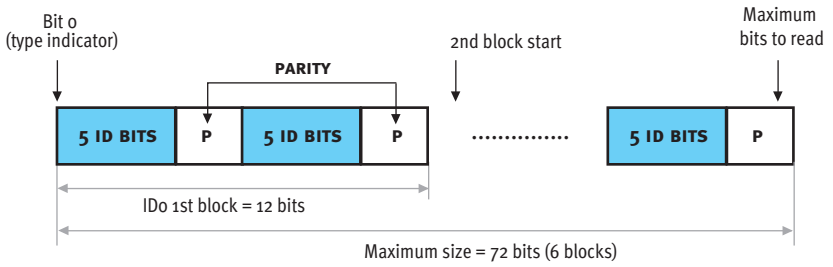
– Adjust to final precise target frequency over the next 800 microseconds (a reset signal.) The tag will use this period to internally adjust itself to a possible new power level, and assume that this signal is the new CW (100%) power level.
– Issue the appropriate calibration signals.
– Re-issue any previously issued or desired global commands.
– Continue reading tags until none are remaining.

The above procedure has been determined to provide the best possible success rate to continue tree traversal amidst a frequency change by the reader. It is also likely that a small percentage of the population will not be able to respond due to a power. These tags will require yet another frequency hop and calibration to be read.

13.6. Identification Number IDo and ID1

Figure 28

FULL TAG BITS LAYOUT CHART IDO AND ID1



The IDo and ID1 string is used only to singulate tags within a reader field by negotiating a minimum of random bits. The read command will be required to be used after singulation in order to extract the EPC™. For IDo & ID1 based singulation, twelve or twenty four bits will singulate a tag in most populations, but more bits may be needed in some cases. IDo and ID1 are broken into 12-bit blocks for both security modes. Each block consists of ten ID bits, and two parity bits as follows:

– 1st five bits of a static or generated ID
– 1st parity bit of the preceding
– 2nd five bits of a static or generated ID
– 2nd parity bit of the preceding

When reading small populations of tags, on average only one block may need to be read. In large tag populations, several blocks may need to be negotiated to singulate a tag. Because the number of bits to negotiate is variable, parity bits are interspaced within the ID bits. This guarantees that each block has some error detection. Parity generation is a simple enough procedure that it can be generated on the tag as the ID bits are generated. Each block is to be read in its entirety (all twelve bits.) Multiple blocks can be read, up to a maximum of six blocks, depending upon the requirements to negotiate a particular tag to uniqueness. The reader determines how many blocks will be used; however, the tag is to enter into a singulated command start state only after receiving a null at the end of a block, and not in the middle of the block. To be specific, after the tag backscatters every 12th (a parity) bit, the next bit from the reader will acknowledge this parity bit with same, then next bit (13th) from the tag is sent and finally the reader

must send a data 'null' in order to place the tag into a singulated command start state. Data 'null's in other bit positions (must be n+1, where n is multiples of 12) will transition the tag into a traversal mute state.

Parity will be determined as odd parity. An odd number of '1's in the first 5 bits will result in a parity value of '0'. An even number of 1's will result in a parity value of '1'. Bits 'ooooo' will result in a parity of '1', and '11111' will result in a parity bit '0'.

The first bit transmitted by the tag will also indicate the style of implementation of the contents of IDo or ID1. The style of implementation is directed by the reader when it issues a SetNegotiationPage command. Styles are defined as (1st bit 0) for IDo, AND (1st bit 1) for ID1. Both styles are required to be implemented and available in each chip design.

The ID1, pseudo random implementation is to be a pseudo-random number that can be generated from a seed based in the ID2 register's CRC. The generation will conform also to the interspersed parity bits as defined in the previous paragraph. The result is 60 bits of generated ID with 12 bits of interspersed parity bits for a total of 72 bits. The pseudo-random number should be generated in such a fashion to guarantee uniqueness over the full 72 bits even when two tags have the same CRC information. This number is required to be the same for every singulation of a particular chip regardless of environmental extremes (RF power, temperature, humidity, light, etc) and regardless of the specific reader that is addressing the tag. Alternatively, it is acceptable to store a randomly generated 72 bit number conforming to the parity rules above into a tag's memory for ID1. This may be less attractive for implementation in that it would require additional storage on the tag for these 72 bits.

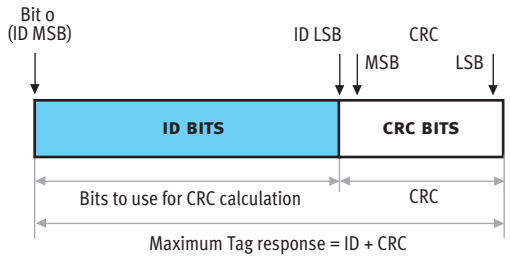
The IDo, truly random implementation is to be a randomly generated number that does change at least each time a Reset (tree traversal process) occurs. However, the suggested implementation is to randomly generate 1 bit at a time while in the traversal process for a true bit-by-bit coin flip randomization. The bit-by-bit method will not generate every 6th bit, but shall implement a parity device to calculate each 6th bit based on the previous 5 bits as outlined in the first paragraph of this section. Implementation of the bit-by-bit method implies a number that changes constantly (on each traversal, or at each data null). Dynamic random ID generation is used when the highest level of security is needed.

13.7. Identification Number ID2

The ID2 number is transmitted highest order bit first in transmission time. ID2 is the storage of the user EPC™ data, and is the default page used to singulate the tag.

Figure 29

FULL TAG BITS LAYOUT CHART ID2



13.8. Error Detection Code

The error detection code for the ID2 register is a standard CCITT 16-bit CRC (cyclic redundancy check) code. Several parameters are necessary to fully define the particular implementation in use by this protocol by default. These are:

Table 9

PARAMETER	MINIMUM
Polynomial	$X^{16} + X^{12} + X^5 + 1$
Preload Value	oXffff
Suggested Calculation	Table Driven (16 bits calculations)

14. ANNEXES

14.1. CRC-16 Example

The following is a section of a 'c' program that provides an extremely quick method of calculating a tag's CRC under the Class o protocol. This allows the reader to perform a bit to bit CRC calculation on the fly, so that little processing power is required for this procedure. This follows the standard CCITT CRC calculation method described in an earlier section of this specification.

```
Static const t_SHORTWORD crcTable[ 256]  =
{
 0x0000, 0x1189, 0x2312, 0x329b, 0x4624, 0x57ad, 0x6536, 0x74bf,
 0x8c48, 0x9dc1, 0xaf5a, 0xbed3, 0xca6c, 0xdbe5, 0xe97e, 0xf8f7,
 0x1081, 0x0108, 0x3393, 0x221a, 0x56a5, 0x472c, 0x75b7, 0x643e,
 0x9cc9, 0x8d40, 0xbfdb, 0xae52, 0xdaed, 0xcb64, 0xf9ff, 0xe876,
 0x2102, 0x308b, 0x0210, 0x1399, 0x6726, 0x76af, 0x4434, 0x55bd,
 0xad4a, 0xbcc3, 0x8e58, 0x9fd1, 0xeb6e, 0xfae7, 0xc87c, 0xd9f5,
 0x3183, 0x200a, 0x1291, 0x0318, 0x77a7, 0x662e, 0x54b5, 0x453c,
 0xbdcb, 0xac42, 0x9ed9, 0x8f50, 0xfbef, 0xea66, 0xd8fd, 0xc974,
 0x4204, 0x538d, 0x6116, 0x709f, 0x0420, 0x15a9, 0x2732, 0x36bb,
 0xce4c, 0xdfc5, 0xed5e, 0xfcd7, 0x8868, 0x99e1, 0xab7a, 0xbaf3,
 0x5285, 0x430c, 0x7197, 0x601e, 0x14a1, 0x0528, 0x37b3, 0x263a,
 0xdecd, 0xcf44, 0xfddf, 0xec56, 0x98e9, 0x8960, 0xbbfb, 0xaa72,
 0x6306, 0x728f, 0x4014, 0x519d, 0x2522, 0x34ab, 0x0630, 0x17b9,
 0xef4e, 0xfec7, 0xcc5c, 0xddd5, 0xa96a, 0xb8e3, 0x8a78, 0x9bf1,
 0x7387, 0x620e, 0x5095, 0x411c, 0x35a3, 0x242a, 0x16b1, 0x0738,
 0xffcf, 0xee46, 0xdcdd, 0xcd54, 0xb9eb, 0xa862, 0x9af9, 0x8b70,
 0x8408, 0x9581, 0xa71a, 0xb693, 0xc22c, 0xd3a5, 0xe13e, 0xf0b7,
 0x0840, 0x19c9, 0x2b52, 0x3adb, 0x4e64, 0x5fed, 0x6d76, 0x7cff,
 0x9489, 0x8500, 0xb79b, 0xa612, 0xd2ad, 0xc324, 0xf1bf, 0xe036,
 0x18c1, 0x0948, 0x3bd3, 0x2a5a, 0x5ee5, 0x4f6c, 0x7df7, 0x6c7e,
 0xa50a, 0xb483, 0x8618, 0x9791, 0xe32e, 0xf2a7, 0xc03c, 0xd1b5,
 0x2942, 0x38cb, 0x0a50, 0x1bd9, 0x6f66, 0x7eef, 0x4c74, 0x5dfd,
 0xb58b, 0xa402, 0x9699, 0x8710, 0xf3af, 0xe226, 0xd0bd, 0xc134,
```

```
 0x39c3, 0x284a, 0x1ad1, 0x0b58, 0x7fe7, 0x6e6e, 0x5cf5, 0x4d7c,
 0xc60c, 0xd785, 0xe51e, 0xf497, 0x8028, 0x91a1, 0xa33a, 0xb2b3,
 0x4a44, 0x5bcd, 0x6956, 0x78df, 0x0c60, 0x1de9, 0x2f72, 0x3efb,
 0xd68d, 0xc704, 0xf59f, 0xe416, 0x90a9, 0x8120, 0xb3bb, 0xa232,
 0x5ac5, 0x4b4c, 0x79d7, 0x685e, 0x1ce1, 0x0d68, 0x3ff3, 0x2e7a,
 0xe70e, 0xf687, 0xc41c, 0xd595, 0xa12a, 0xb0a3, 0x8238, 0x93b1,
 0x6b46, 0x7acf, 0x4854, 0x59dd, 0x2d62, 0x3ceb, 0x0e70, 0x1ff9,
 0xf78f, 0xe606, 0xd49d, 0xc514, 0xb1ab, 0xa022, 0x92b9, 0x8330,
 0x7bc7, 0x6a4e, 0x58d5, 0x495c, 0x3de3, 0x2c6a, 0x1ef1, 0x0f78
};



/*
 * uint16 calcBlockCRC(size_t count, uint16 crc, void *buffer)
 *
 * ARGUMENTS
 * size_t count : The number of bytes in the buffer
 * uint16 crc : The initial value of the CRC
 * void *buffer : The buffer whose CRC is to be calculated.
 *
 * DESCRIPTION
 * This routine is called to calculate the CCITT CRC value of a block
 * of data.
 *
 * RETURNS
 * CRC value.
 *
 */
t_SHORTWORD p_RS485_calcBlockCRC(size_t count, t_SHORTWORD crc, const void
*buffer)
{
 const t_BYTE *pBuf = (const t_BYTE *)buffer;
 while (count--)
 crc = (t_SHORTWORD)((crc >> 8) ^ crcTable[ (t_BYTE)(crc ^ *pBuf++)] );
 return (crc ^ ((t_SHORTWORD)0Xffff));
}
```

Designed by Foxner. www.foxner.com