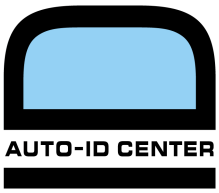
Published February 1, 2003. Distribution restricted to Sponsors until May 1, 2003.



TECHNICAL REPORT

The Use of the Electronic Product Code™

Daniel Engels

AUTO-ID CENTER MASSACHUSETTS INSTITUTE OF TECHNOLOGY, 400 TECHNOLOGY SQUARE, SIXTH FLOOR, CAMBRIDGE, MA 02139-4307, USA

ABSTRACT

An Electronic Product Code™ is a unique object identifier, and its bit-wise representation is referred to as an EPC™. An EPC™ representation encodes the three unique numbers that constitute the Electronic Product Code (Domain Manager number, Object Code number, and Serial Number) plus a Version Number that is used to differentiate between the various EPC™ versions. The structure of seven versions of EPC™ have been defined. This structure specifies the total number of bits available to encode each of the four numbers represented. However, specific operational details for the Electronic Product Code™ and the EPC™, such as the specific version number used to identify a particular EPC™ version, have not been defined or reserved in these structural definitions. The result has been an incomplete definition of EPCs™ that prevents their practical use. This paper defines the EPC's™ operational details to enable their practical use. Specific Version Numbers are assigned for each EPC™ version including guidelines for the assignment of future Version Numbers. Furthermore, practical use guidelines and special numbers are identified for the three numbers comprising the Electronic Product Code™.

TECHNICAL REPORT

The Use of the Electronic Product Code™

Biography

Daniel W. Engels
Associate Director

Daniel W. Engels received his B.S. from the University of Buffalo, his M.S. from the University of California, Berkeley, and his Ph.D. from the Massachusetts Institute of Technology all in Electrical Engineering and Computer Science. His master's thesis is in the area of computer-aided design for electronic systems, and his doctoral thesis is in the field of theoretical computer science. Dr. Engels joined the Auto-ID Center after obtaining his doctoral degree where he leads the day-to-day research activities of the Center. Dr. Engels' research interests include scheduling theory and applications, real-time system design, distributed and mobile computing, and computer-aided design for embedded systems.

TECHNICAL REPORT

The Use of the Electronic Product Code™

Contents

1. INTRODUCTION

The Electronic Product Code™ is a naming and identification scheme designed to enable the unique identification of all physical and virtual objects, assemblies and groupings of objects, and non-objects such as services. An EPC™ is a specific representation, or encoding, of an Electronic Product Code™ as a sequence of bits. An EPC™ is simply a sequence of bits that encodes an Electronic Product Code™ in a well-defined manner.

An Electronic Product Code™ is comprised of three distinct numbers: Domain Manager number, Object Class number, and Serial Number. An EPC™ encodes each of these three numbers separately with each in its own partition of the EPC™ bit sequence. An EPC™ also encodes a Version Number in its own partition. The Version Number enables the definition of multiple types, or versions, of EPCs™ that differ in total bit length and number of bits in each of the four partitions. Therefore, all EPCs™ contain four distinct partitions that encode four distinct numbers (in order from most significant bit to least significant bit in the EPC™ bit sequence): Version Number, Domain Manager number, Object Class number, and Serial Number. Each of these numbers is encoded as a binary number within its partition. A partition is a predefined set of bits in the EPC™ bit sequence. EPC™ versions differ only in their structure, that is, the number of bits available to encode each of the four numbers.

The structure of seven EPC™ versions in three bit sequence lengths, 64 bits, 96 bits, and 256 bits, have been defined [1, 2, 3]. However, operational details for the practical usage of these versions, such as the allocation of version numbers and definition of any reserved or preallocated numbers in these partitions, has not been defined.

This paper defines the operational details for the practical usage of Electronic Product Codes™ and EPCs™. Section 2 reviews the structure of the defined EPC™ versions and the design of the Electronic Product Code™. Section 3 defines the operational constraints on the Version Numbers. Section 4, Section 5, and Section 6 define the operational constraints on the Domain Manager number, the Object Class number, and the Serial Number respectively. Section 7 summarizes the operational details for the EPC™.

2. EPC™ STRUCTURAL OVERVIEW

An EPC™ is a specific representation of an Electronic Product Code™ as a sequence of bits. As such, an EPC™ encodes the three numbers comprising an Electronic Product Code™ plus a Version Number that identifies the structure of that particular EPC™. Thus, all EPCs™ encode four distinct numbers (Version Number, Domain Manager number, Object Class number, and Serial Number) with each number encoded in its distinct partition of the bit sequence. The structure of the defined EPCs™ is shown in Table 1. The only structural difference between EPC™ versions is the number of bits within a specific partition. The 96-bit EPC™ versions, EPC-96, are defined in the white paper MIT-AUTOID-WH-002 [1]. The 64-bit EPC™ versions, EPC-64, are defined in the white paper MIT-AUTOID-WH-008 [2]. And, the 256-bit EPC™ versions, EPC-256, are defined in the white paper MIT-AUTOID-TR-010 [3].

The Version Number indicates the total number of bits in that particular EPC™ as well as the number of bits that are available to encode numbers in each of the remaining three partitions. The Domain Manager number, the Object Class number, and the Serial Number comprise the Electronic Product Code™, and their combination uniquely identifies an object. The owner of a Domain Manager number controls the allocation of all Object Class numbers and Serial Numbers for that Domain Manager number.

Table 1: The structure (bit allocations) for each of the seven defined EPC™ versions.

		VERSION NUMBER	DOMAIN MANAGER	OBJECT CLASS	SERIAL NUMBER
EPC-64	**TYPE I**	2	21	17	24
	TYPE II	2	15	13	34
	TYPE III	2	26	13	23
EPC-96	**TYPE I**	8	28	24	36
EPC-256	**TYPE I**	8	32	56	192
	TYPE II	8	64	56	128
	TYPE III	8	128	56	64

The structure of the EPCs™ are designed such that all Electronic Product Codes™ that may be represented by an EPC™ of smaller bit length may be represented also by all EPC™ versions of larger total bit length. The effect of this design decision is that the structures of future EPC™ versions are constrained by the maximum bit lengths for a partition in shorter EPC™ versions and the minimum bit lengths for a partition in longer EPC™ versions.

3. VERSION NUMBER

The Version Number encoded within an EPC™ identifies the structure of that EPC™. Seven versions of EPCs™ have been defined thus far as shown in Table 1. The three 64-bit versions have 2-bit Version Number partitions, and the one 96-bit and three 256-bit versions have 8-bit Version Number partitions.

An EPC™ is designed to be used as a unique identifier that is communicated to some sensor device, such as an RFID reader. As such, an EPC™ may be obtained in a bit-wise fashion, that is, one bit at a time without prior knowledge on the total length of that particular EPC™. In these scenarios, it is important for the Version Number to be obtained first and for the Version Number to indicate the total number of bits within that particular EPC™. Knowledge of the total length of an EPC™ without knowledge of its Version Number is unreliable in such situations; therefore, it cannot be relied upon to be known in advance.

The potential to obtain an EPC™ one bit at a time has led us to allocate Version Numbers such that they are unique (between EPCs™ of different lengths) in a most significant bit to least significant bit fashion. Therefore, the position of the most significant bit with a value of 1 indicates the total length of the EPC™. The three 64-bit EPCs™ have only two bits in their Version Number partition. These EPC™ versions are assigned Version Numbers 01, 10, and 11. Consequently, a one in either of the two most significant bit positions of the Version Number indicates a 64-bit EPC™. To be distinguishable from 64-bit EPCs™, all EPCs™ greater than 64 bits in length must have the two most significant bits of their Version Number be equal to 00. Accordingly, we have defined that all 96-bit EPC™ Version Numbers begin with the bit sequence 001. Similarly, all EPCs™ greater than 96 bits in length must have the first three significant bits of their Version Number equal to 000; therefore, we have defined that all 256-bit EPC™ Version Numbers begin with the bit sequence 0000 1. The specific Version Number assigned to each defined EPC™ version is shown in Table 2.

Table 2: Version Number assignments for the defined EPC™ versions.

EPC VERSION		VALUE (BIN)	VALUE (HEX)
EPC-64	**TYPE I**	01	1
	TYPE II	10	2
	TYPE III	11	3
	EXPANSION	NA	NA
EPC-96	**TYPE I**	0010 0001	21
	EXPANSION	0010 0000	20
EPC-256	**TYPE I**	0000 1001	09
	TYPE II	0000 1010	0A
	TYPE III	0000 1011	0B
	EXPANSION	0000 1000	08
RESERVED		0000 0000	00

The 2-bit Version Number for all 64-bit EPCs™ limits the total number of distinct 64-bit EPC™ versions to three (3). To prevent such limitations in larger bit-length EPCs™, we reserve specific Version Numbers for use with longer bitlength Version Numbers. Specifically, the Version Numbers 0010 0000 (20 hex) and 0000 1000 (08 hex) are reserved to enable Version Numbers with bit lengths greater than 8 in the 96-bit and 256-bit EPCs™ respectively. Thus, a 96-bit EPC™ with a 16 bit Version Number, for example, will have a Version Number that begins with the bit sequence 0010 0000.

For more generic expansion, the Version Number 0000 0000 (00 hex) is reserved for the expansion of the length of the Version Number in future (as yet undefined) EPC™ versions. The 00 (hex) Version Number has been used as an experimental Version Number by the Auto-ID Center for EPCs™ used within its Field Trial and other internal trials.

4. DOMAIN MANAGER

The Domain Manager partition is used to encode the Domain Manager number of an Electronic Product Code™. This encoding is simply a zero padded binary representation of the value of the Domain Manager number. For example, the decimal number 1206466 is represented as a 21-bit binary number as

100100110100011000010

and a 26-bit binary number as

00000100100110100011000010.

The variability in the lengths of the Domain Manager partitions in the various EPC™ versions makes the smaller Domain Manager numbers the most valuable. The EPC-64 Type II has the smallest Domain Manager partition with only 15 bits. Therefore, only Domain Manager numbers smaller than $2^{15} = 32768$ may be represented by all EPC™ versions.

Two Domain Manager numbers have been reserved for special purposes: 0 and 167 842 659 (in decimal). The first reserved Domain Manager number, zero (0), is allocated to MIT. Thus, MIT controls the allocation of all Electronic Product Codes™ that contain the zero (0) Domain Manager number.

The second reserved Domain Manager number is allocated for private use. This Domain Manager number is 167 842 659 (in decimal) which may also be represented in hexidecimal as A011363 and in binary as

10100000000100010011011000011.

The need for a private use Domain Manager number stems from the anticipated use patterns of Electronic Product Codes™. Individuals and organizations that have private objects that they wish to identify using Electronic Product Codes™ will use any convenient Electronic Product Code™ without registering those numbers with the Global Object Name System. To minimize confusion and conflict with Electronic Product Codes™ owned by other individuals, particularly when private objects enter a space not controlled by their owner, all private objects should be allocated Electronic Product Codes™ either controlled by their owner or guaranteed to never be controlled by another entity. By reserving a Domain Manager number for private use, private objects allocated Electronic Product Codes™ with that Domain Manager number are guaranteed to never infringe upon the namespace owned by another entity.

5. OBJECT CLASS

The Object Class partition is used to encode the Object Class number of an Electronic Product Code™. This encoding is simply a zero padded binary representation of the value of the Object Class number.

There are no restrictions on the assignment of Object Class numbers by the owner of a particular Domain Manager number. However, there is one suggested use Object Class number, the zero (0) Object Class number. We suggest that the zero Object Class number not be used as part of an Electronic Product Code.

6. SERIAL NUMBER

The Serial Number partition is used to encode the Serial Number of an Electronic Product Code™. This encoding is simply a zero padded binary representation of the value of the Serial Number.

There are no restrictions on the assignment of Serial Numbers by the owner of a particular Domain Manager number. However, there is one suggested use Serial Number, the zero (0) Serial Number. We suggest that the zero Serial Number not be used as part of an Electronic Product Code™.

7. SUMMARY

This paper defines the operational details necessary for the practical use of Electronic Product Codes™ and EPCs™. In doing so, the design strategy for the relationship between versions of different length and the allocation of Version Numbers have been clarified. Specific Version Numbers are defined for each of the seven defined EPC™ versions, and a private use Domain Manager number has been defined to offer an authorized alternative to the unauthorized use of Domain Manager numbers. The zero (0) Domain Manager number is owned by the Auto-ID Center, and the zero (0) Object Class number and Serial Number are suggested not to be used within an allocated Electronic Product Code™.

8. REFERENCES

1. **D.Brock, "The Electronic Product Code™ – A Naming Scheme for Physical Objects".**
Technical Report MIT-AUTOID-WH-002, Auto-ID Center, January 2001.

2. **D. Brock, "The Compact Electronic Product Code™ – a 64-bit Representation of the Electronic Product Code™".**
Technical Report MIT-AUTOID-WH-008, Auto-ID Center, November 2001.

3. **D.W. Engels, "EPC-256: The 256-bit Electronic Product Code™ Representation".**
Technical Report MIT-AUTOID-TR-010, Auto-ID Center, February 2003.

Designed and produced by Foxner. www.foxner.com