A Technique for Label Text Compression Applied to RFID Passive Tag

Patcharaporn Choeysuwan[1], Manas Sangworasil[2], and Somsak Choomchuay[3]

Department of Electronic Engineering, Faculty of Engineering, King Mongkut's Institute of Technology Ladkrabang
Chalongkrung Rd., Ladkrabang Bangkok 10520, Thailand, Tel: 02-329-8344-5, Fax: 02-329-8346
[1]Email: patcharapornc@yahoo.com, [2]Email: ksmanas@kmitl.ac.th, [3]Email: kchsomsa@kmitl.ac.th

Abstract

The advantage of Radio frequency identification (RFID) technologies as an intelligent agent system supports real time decision as a solution for integration challenge. The RFID tags are the next generation of the 1D barcodes and 2D barcode. The enhancements of functionality in application are to store a unique identification. The tag memory is a very important element on RFID system base on cost and physical requirements as attributes and characteristic. The less bit memory is relative inexpensive cost and vise versa for more than memory are more expensive. In this paper, we present the framework of UHF RFID passive tag at frequency range 860 MHz to 960 MHz using the restrictive memory by implement with the compression text. The framework consists of UHF RFID tag; Methodology, Electronic Product Code (EPC), Lossless compression entropy and result show the reducing size of file after passed each compression text.

Keywords: RFID, Passive tag, Data compression, Text Compression, Logistics

1. Introduction

RFID is a collective term for near field communication devices and in reality refers to devices adhering to a number of different standards. In the HF band interfaces have been standardized for "proximity" (ISO 14443), "vicinity" (ISO 15693) and "near field" (NFCIP-1/ECMA340, ISO 18092) devices, with maximum operating ranges in the order of 10 cm to 1 m. An ISO18000 [1] defines possible communication interfaces for LF, HF, and UHF bands. The EPC Class-1 standard [2] is well known for UHF item management tags, which also operate at a much longer range. Our experiments were conducted in the HF band using tokens conforming to the ISO 14443 a standard, which is used by popular commercial products such as Philips Mifare (and is also one of the standards specified for e-passports [3]). It also corresponds closely to the "near field" standard.

RFID technology has been grown rapidly for automatic identification and widely considered to represent the next cycle of innovation beyond ubiquitous 1-D and 2-D barcode. One of the key emerging is the opportunities and challenges related to deployment of Electronic Product Code (EPC) and RFID as the unique situation of each manufacturer. The relative impact on EPC/RFID attractiveness will vary based on each product's unique characteristics and each company's specific situation.

One of the benefit potential is reduced the cost of a RFID deployment [4] can be reduce in three major key: hardware, software, and services. Tag costs are one of the key considerations in RFID deployment. Tags become in various of appearance shape and memory size base on applications which these factors affect tag pricing significantly.

2. Problem Identification

EPC/RFID tag as well as barcode system are generally used for the same purpose: a product ID. Now, let's consider a tiny product (or pallet) packed in a small box with ID either in a form of a barcode or an EPC tag. Many of these pallets (few 10s or few 100s) are again packed in a rather bigger box. That box is of course indentified with another ID. Now, without going back to the data base it is difficult to identify which box contains an interested pallet. One may think of a piece of paper attached to the box. That paper lists all the pallets inside the box. It seems to work fairly well but it is too old fashion with several drawbacks. Toward our knowledge and the literatures we do have, we are thinking of an electronic paper. It works exactly as the paper does but smarter in term of size, processing, electronic information compatibility, and reusability. Here comes another tag with bigger memory. To be more precise, we are looking at gen-2 tag with quite memory and UCODE product. However for the memory in such tag or UCODE product to be effectively used:- we are trying compress the information before storing them.

The rest of this paper is organized as follows: In section 3 we continue giving more background on EPC and UCODE. Then in section 4 we discuss several existing text compression algorithms, for example: Bzip2, Deflate, Huffman, and arithmetic coding. In section 5 we detail some simulation and results of selected algorithms. Finally, in section 6 we conclude the work and discuss the possibility of future extension.

3. RFID Methodology

3.1 EPCglobal & EPC

The main focus of the EPCglobal [5] is to create both a worldwide standard for RFID and the use of the Internet to share data via the EPCglobal Network. The EPCglobal network architecture enumerate at a high level which the part of the hardware, software and data standard. The interoperability is clear for end-to-end supply chain applications when goods move through the difference trading partners and business. The EPCglobal Architecture Framework [6], separate in 3 groups of standard as EPC Physical Object Exchange, EPC Infrastructure for Data Capture and EPC Data Exchange.

EPC [7] is a numbering scheme that allows assignment of a unique identifier to any physical object. The current format of EPC standard includes the following fields of Header, Manager number, Object class and Serial number from below as Fig. 1 with 1-D barcode standard EAN-13 [8].

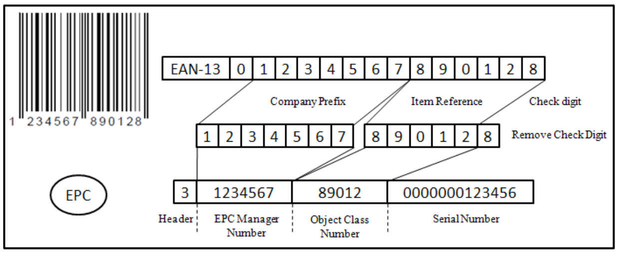

Fig. 1: the EPC standard fields compare with EAN-13

3.2 RFID communication protocol

The protocol for enabling communications and interface between a tag and a reader for Class 1 operation are in range of UHF. The EPC tag to reader interface standard for interoperability is UHF generation 2 standard, commonly gen-2. Tags are complying with this standard on feature Read/Write capabilities and communicate equally with reader operating at various frequencies between 860MHz and 930 MHz. The gen-2 standard will align with ISO/IEC 18000-6.

3.3 UCODE RFID tags

Current the global market place with the expansion of RF chip, package, software/middleware and system integration. RFID market is developing rapidly. The ideal for supply chain and logistics applications requiring high speed operation, UCODE passive UHF transponder from many companies are produce in various type and memory usage. The below Table 1 is example UCODE Passive RFID ICs characteristic for tags that populate in marketplace which we choose for prototype from NXP smart labelandtags IC [9].

TABLE I
UCODE Passive UHF Transponder (UHF: 860MHz – 960 MHz)

Product Feature	UCODE EPC Gen-2	UCODE G2XL	UCODE G2XM
Total memory (bit)	512	368	880
Write Cycles	100,000	100,000	100,000
Data rentation (Yrs)	10	50	50
RF Interface	EPC Class-1 Gen-2	EPC Class-1 Gen-2	EPC Class-1 Gen-2
Oper. Distance (m)	Up to 7	Up to 10	Up to 10

4. Lossless Data Compression

The compression is the mathematical process that tries to reduce the amount of data while preserving the original information and meaning. With lossless compression, the original data can be 100% retrieved. The main approaches to text (or lossless) compression are dictionary and statistical bases. In dictionary based methods, those consecutive characters are replaced with a pointer to an entry in a dictionary. Statistical based compression calculates the frequencies of word occurrences and builds a statistical table for later conversion. By using this table, each character can be converted to specified code, and storage space is decreased.

Despite the computational complexity, many modern compression algorithms make best use of both approaches. For instances, Bzip2 uses the Burrows-Wheeler compression (BWT) to convert frequently character sequences into strings of identical letters then applies a move to front (MTF) transform and finally use Huffman coding. The Zlib, (also called Deflate) utilizes LZ77 algorithm and Huffman coding.

Two most common statistical compression methods are Huffman and Arithmetic coding. Traditional Huffman utilizes a static table to represent all the characters and their frequencies, then generates a code table accordingly. Sine the frequencies are integers and the probabilities can be written as scaled integers and write to the Huffman tree on the file. In a Huffman tree where each node is the sum of its children have weighted sum of the leaves. The weights are the distances of the leaves from the root, equals the sum of the internal nodes. Huffman coding is optimal for a symbol-by-symbol coding with input probability distribution. For Adaptive Huffman coding, the tree and corresponding encoding scheme change accordingly base on technique of algorithm FGK developed by Kunth and Gallager. A binary tree with n leaf nodes has 2n-1 nodes and 2n-2 nodes by ordering node y_1, y_2, $_{y2n-2}$ and weight on y_k is non-decreasing with k, so y_{2k-1} and y_{2k} are siblings.

Arithmetic coding [10] represents frequently characters using low bit and infrequently characters using high bit. The source symbols are mapped onto intervals of range [0,1) based on their probability of occurrence. The processed symbols sequence is represented the codeword. For example of input symbol as JAVA, the probability of symbol A is p(A) = 0.50, symbol J is p(J)=0.25 and symbol V is p(V)=0.25. The partitioning represent by the intervals range of [0,0.50), [0.50,0.75) and [0.75,1.0) consecutively. An interval can be specified by its lower and upper limits as (low, high) or by one limit and width. The intervals during the arithmetic coding process are in the new intervals value, defined by set of Range (High to low). For the Adaptive arithmetic, the encoder should be given the cumulative frequencies of each symbol. The adaptive arithmetic model keeps the symbols, their counts frequencies of occurrence and their cumulative frequencies. The frequencies could be changed each time it is encoded and update the cumulative frequencies.

5. Investigation and results

Each built product unit hold 17 characters assigned for its ID, so called an identifier. 40-80 units are packed in single box and shipped. A 1-D barcode that links to database is attached to such a box. In this work a 1-D barcode and database system is intended to be replaced by a RFID tags. Therefore we need to design a method to put all the information in such a passive tag. However, as the tag holds very restricted amount of memory. Data compression is one possible solution to the problem. Toward this case, massive compression is very desirable. Upon our note, very fortunate, data of the same lot product hold quite high degree of redundancy. This is shown in Fig. 3. Instead of compress the whole original serial shipment data, we can compress only those variable portions that hold only 5-8 character digits.

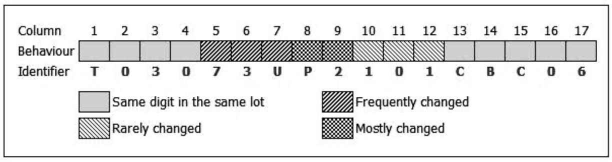

Fig. 3: A 17-character digit used for product unit identification

In our investigation, digit 5th to 12th is considered to lie in the variable portion. Digit 5th to 9th are for Serial number, digit 10th to 12th denote circuit version code. Digit 5th to 7th found frequently changed. However, data in these sets still hold some degree of redundancy. Likewise digit 8th and 9th found mostly changed. Digit 10th is rarely changed (mostly stay unchanged for the same lot). Digit 11th and 12th are mostly changed. For the pre-coding to be more efficient we do ascending sorting of all input identifiers. The obtained pre-coded data is then compressed with the algorithms of interest.

Pre-coding Example:

For coding demonstration, let's consider Fig. 4 where 15 identifiers are given.

C1: count one column each of digit 5th, 6th, 7th. The coded data are as *7,5,B,5,Y,5,6,5,V,5,7,5,8,3,C,1,E,1,L,2, M,1,N,2,1,1,2,3,E,1*

C2: count two columns at digit 5th-6th and one column at digit 7th. The coded data are as *276,5,2BV,5,2Y7,5, 8,3,C,1,E,1,L,2,M,1,N,2,1,1,2,3,E,1*

C3: count three columns at digit 5th-6th-7th. The coded data are as *3768,3,376C,1,376E,1,3BVL,2,3BVM,1, 3BVN,2,3Y71,1,3Y72,3,3Y7E,1*

Since digit 8th and 9th have less duplicated symbol, it is not so worthwhile to apply any coding scheme to them. Raw data are fed: i.e. *QQV2C3UPFGY0016* and *RVS2NAXHY6BRW07* respectively.

No.	Identifier	C1-C4	**C5-C12**	C5	C6	C7	C8	C9	C10	C11	C12	C13-C17
1	T030768QR101A3C2J	T030	768QR101	7	6	8	Q	R	1	0	1	A3C2J
2	T030768QV101A3C2J	T030	768QV101	7	6	8	Q	V	1	0	1	A3C2J
3	T030768VS101A3C2J	T030	768VS101	7	6	8	V	S	1	0	1	A3C2J
4	T03076C22101A3C2J	T030	76C22101	7	6	C	2	2	1	0	1	A3C2J
5	T03076ECN101A3C2J	T030	76ECN101	7	6	E	C	N	1	0	1	A3C2J
6	T030BVL3A149A3C2	T030	BVL3A149	B	V	L	3	A	1	4	9	A3C2J
7	T030BVLUX149A3C2	T030	BVLUX149	B	V	L	U	X	1	4	9	A3C2J
8	T030BVMPH149A3C2	T030	BVMPH149	B	V	M	P	H	1	4	9	A3C2J
9	T030BVNFY149A3C2	T030	BVNFY149	B	V	N	F	Y	1	4	9	A3C2J
10	T030BVNG6149A3C2	T030	BVNG6149	B	V	N	G	6	1	4	9	A3C2J
11	T030Y71YB325A3C2J	T030	Y71YB325	Y	7	1	Y	B	3	2	5	A3C2J
12	T030Y720R325A3C2J	T030	Y720R325	Y	7	2	0	R	3	2	5	A3C2J
13	T030Y720W325A3C2	T030	Y720W325	Y	7	2	0	W	3	2	5	A3C2J
14	T030Y7210325A3C2J	T030	Y7210325	Y	7	2	1	0	3	2	5	A3C2J
15	T030Y7E67325A3C2J	T030	Y7E67325	Y	7	E	6	7	3	2	5	A3C2J

Fig. 4 Fifteen identifiers for coding examples (non-sorting)

From investigated information above, we can apply the similar method in arrangement of digit 11th-12th. Since digit 10th always hold same data in the same lot, the pre-coding can be made as follows:

D1: count one column each of digit 10th,11th,12th. The coded data is as *1,10,3,5,0,5,4,5,2,5,1,5,9,5,5,5*

D2: count one column at digit 10th and two columns at digit 11th-12th. The coded as *1,10,3,5,201,5,249,5,225,5*

D3: count three columns at digit 10th-11th-12th. The coded as *3101,5,3149,5,3325,5*

Above arrangement of digit 5th to 12th as C1 to C3 column 8th and 9th, and D1 to D3, we have set to 9 pre-coded patterns. Those are shown in the first column of Table II below. Table II is then describe all the most possible as well as each compression algorithm outcome. The compression algorithms under investigation are Bzip2, Zlib, Huffman, and Arithmetic.

TABLE II

Achieved information size according to compression scheme

Digit 5th-12th	Information size after compression (bits)				
Pattern§	Pre-coded	Bz2	Zlib	Huffman	Arithmetic
C1,89,D1	1,840	1,720	1,336	1,096	1,116
C1,89,D2	1,816	1,720	1,304	1,088	1,088
C1,89,D3	1,784	1,688	1,320	1,072	1,069
C2,89,D1	1,816	1,688	1,312	1,088	1,087
C2,89,D2	1,792	1,688	1,304	1,072	1,076
C2,89,D3	1,760	1,688	1,304	1,056	1,059
C3,89,D1	1,992	1,688	1,336	1,192	1,191
C3,89,D2	1,968	1,784	1,336	1,184	1,178
C3,89,D3	1,936	1,768	1,336	1,160	1,156
Average	1,856	1,715	1,321	1,112	1,113

§ There is no-pre-coding for data in column 8 and 9

Be noted that the original raw data include 17 digits with carriage return and new line as total 19 digits per 1 serial. So, total size of 80 identifiers is 12,144 bits. After rearrangement (or pre-coding) the data size is reduced from 12,144 bits to 1,992 bits or 84% reduction as the worst case. Bz2, Zlib, Huffman, and Arithmetic compressions show the reduction performance of 7.6%, 28.8%, 40.1 and 40.0% respectively. Human coding and arithmetic coding are similar in performance and they offer vey prominent compression ratio. According to the different patterns of arrangement for second stage compression, we can notice that one or two columns at a time (say: C1, C2, D1, D2) and their combination offers better compression ratio compared to the combination that includes three column at a time (say: C3 and/or D3).

We also extend the experiment to another set of 80 identified serial product units. Average and standard deviation of information size are shown in Table 3. It should be noted that those 2 sample sets offer the same trend of compression performance.

TABLE III

Results of Sample Sets (bits); $\overline{x}$ = Average, σ = Standard derivation

Data set	Pre-coding		Bz2		Zlib		Huffman		Arithmetic	
	$\overline{x}$	σ	$\overline{x}$	σ	$\overline{x}$	σ	$\overline{x}$	σ	$\overline{x}$	σ
1	1,856	86.2	1,715	37.5	1,321	15.2	1,112	52.0	1,113	49.6
2	1,692	37.8	1,695	17.6	1,308	12.1	1,021	13.9	1,022	13.9

6. Conclusions

The achievement an effective using of RFID tags, it is reasonable to reduce the amount of data before storing. In this paper we have applied in 2 stage compression of 17 digits with 80 identified serial product unit. At the first stage we apply modified RLE-like technique and at the second stage we investigate several lossless compression algorithms. The original size (12,144 bit) can be reduced by almost 85% according to variable data selection and first stage compression. Huffman and arithmetic coding show the attractive performance that they can perform 40% further data reduction. Bz2 and Zlip are not so good competitors in our case. Moreover their procedures are also quite complex compared to Huffman or arithmetic coding. We hence would consider those latter two in future extension of this work. The selected algorithm could be justified solidly after the firmware implementation has been taken into account. Upon the obtained result given in Table II, there is a high possibility to store 80 identifiers into a tag with 1024 bit user memory.

7. Acknowledgment

This work is supported by Industry/University Cooperative of Data Storage Technology and Applications Research Center (I/UCRC), King Mongkut's Institute of Technology Ladkrabang and National Electronic and Computer Technology Center (NECTEC), National Science and Technology Development Agency (NSTDA) under scholarship HDD-01-52-11M. We also would like to thank Compart Precision (Thailand) company for their continue support to this research.

References

[1] ISO 18000. RFID for item management: Air Interface

[2] "EPC™ Radio-Frequency Identity Protocols Class-1 Geneaton-2 UHF RFID Protocol for Communications at 860 MHz-960 MHz version 1.2.0," http://www.epcglobalinc.org/home, October 2008.

[3] A.Juels, D.Molnar, D.wagner. "Security and Privacy Issues in E-passports", *Proceedings of IEEE/CreateNetSecure Comm.,* pp.103-111,2003.

[4] Manish Bhuptani, Shahram Moradpour, "RFID Field Guide Deploying Radio Frequency Identification Systems," Sun Microsystems Press, Prentice Hall, 2005.

[5] "RFID Implementation Cookbook," http://www.epcglobalinc.org/home

[6] "The EPCglobal Architecture Framework", http://www.epcglobalinc.org/home, March 2009.

[7] Yunjia Liu, Dong Wang, "An RFID Middleware Business Process Integration Framework Based on EPC Modeling and Complex Event Processing," *Proc. of 4th International Conference on Computer Sciences and Convergence Information Technology,* pp. 64-69, 2009.

[8] http://www.gs1thailand.org/

[9] "Short data sheet : SL3ICS10 UCODE EPC GEN2", http://www.nxp.com/documents/short_data_sheet/SFS129430.pdf

[10] Hyungjin Kim, Jiangtao Wen, John D.Villasenor, "Secure Arithmetic Coding", *IEEE Transactions on Signal Processing*, Vol.55, No.5, May 2007.

[11] http://broadcast.oreilly.com/2009/01/pymotw-bz2.html

[12] http://broadcast.oreilly.com/2008/12/pymotw-zlib.html

Patcharaporn Choeysuwan received the B.Eng. degrees in Electronics Engineering, Faculty of Engineering, King Mongkut's Institute of Technology Ladkrabang. She is currently studying under M.Eng. program at King Mongkut's Institute of Technology Ladkrabang. Her research interests include compression, encryption algorithms and RFID technology.