ISO18000: 6C Features

Table of Contents

1. Introduction

This document describes in layman the highlighted features of ISO18000:6C protocol (“TypeC” in subsequent text). It is written based on my personal understanding on the TypeC protocol.

The first portion of the document describes the memory structure of the TypeC Tag. This is very useful when we need to perform read/ write operation. The second portion describes the open and secured states of the Tag. This section is largely related to the lock operation. User has to understand the open and secured states in order to protect their data. The third portion describes the standard TypeC features that are inventory, read, write, lock and kill.

2. Objectives

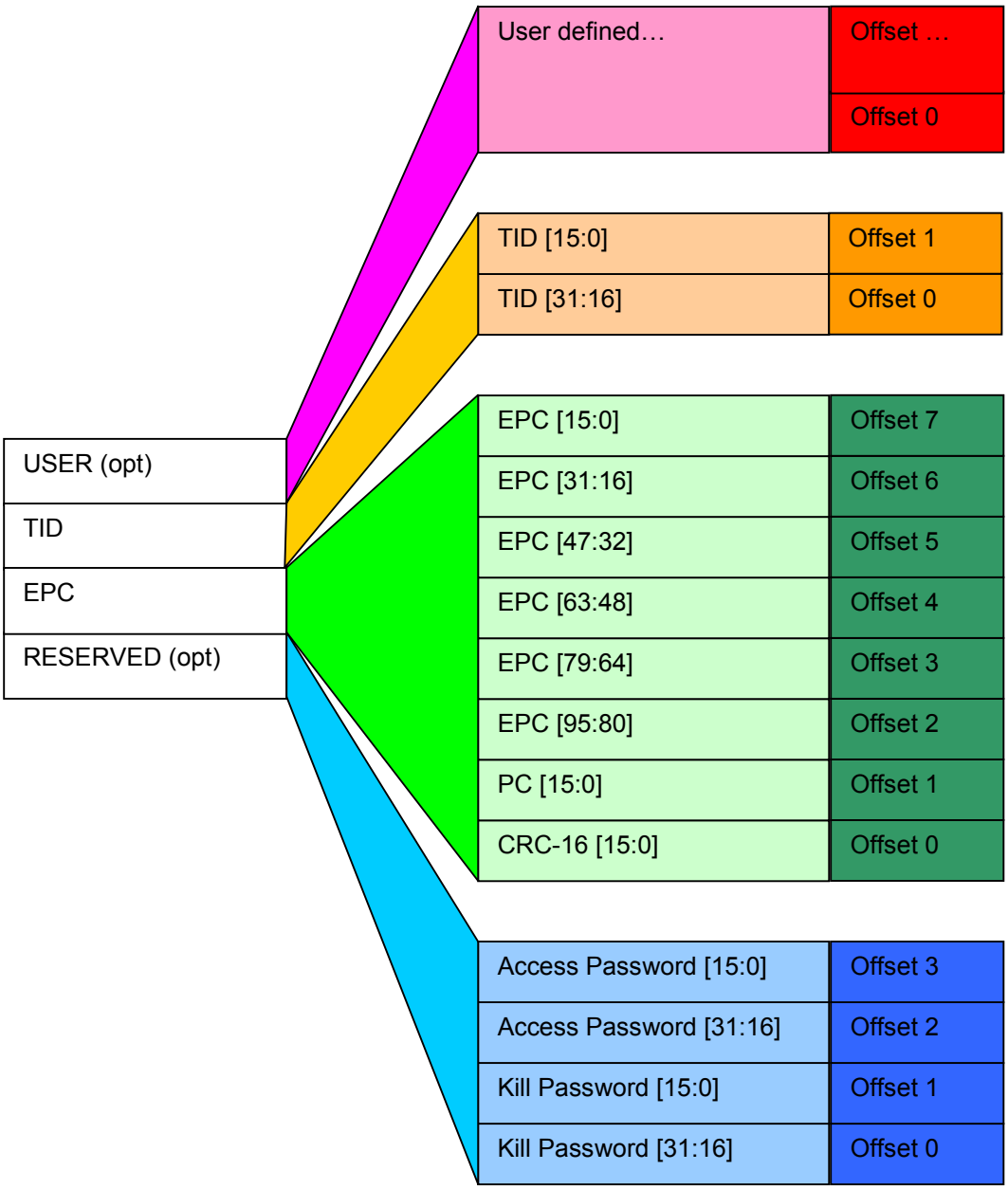
This document aims to give reader a clear idea on TypeC features. Reader should be able to design, test and use TypeC application after this reading.

3. Tag Memory Layout

The entire Tag memory is divided into 4 different memory banks: RESERVED, EPC, TID and USER. Please note that the following information applied to most, but not all, TypeC Tag.

Memory Bank	Purpose
RESERVED	Used to store kill password and access password. Each password occupies 32 bits memory space. The existence of this bank is optional. If a particular Tag does not support kill and access command, the value of this bank is virtually zero(ed). In general, the total space of RESERVE bank is 64 bits.
EPC	Used to store CRC, Protocol Control (PC) and EPC. CRC is the checksum result of PC and EPC. The existence of this bank is compulsory. CRC is a 16-bit data. PC is a 16-bit data. EPC in general occupies 96 bits memory space.
TID	Store Tag specification identifier. The existence of this bank is compulsory. In general, this bank occupies 32 bits memory space. We deal with this bank when we need identify a special Tag. In this case, the Tag might be specially made to perform a proprietary feature.
USER	USER bank is optional. This bank does not exist in most of the Tag. The size of the bank is theoretically unlimited according to the specification.

Figure below gives an overview on TypeC memory layout. RESERVED bank consists of “Kill Password” and “Access password”. EPC bank consists of “CRC-16”, “PC” and “EPC”. TID bank consists of “TID”. USER memory is solely user defined.



The memory sizes of the banks are 16-bit dividable because the smallest memory unit that we can access is a 16-bit block. The blocks are indexed and the index offset of the block in each bank starts with 0. We also refer the memory block index as word pointer or memory offset. In other words, we can only perform read/ write operation with data length that is 16 bits exactly, or dividable by 16 bits.

Possible to read 16-bit data from EPC [40:25]?
No. Although the intended data is 16 bits, you cannot access data that is not bounded in 16-bit block. If you want to read EPC [40:25], you have to divide your read operation into 2 parts. The first part reads EPC [31:16]. The second part reads EPC [47: 32]. Then, you can do data manipulation to get your final result after 2 reads.

What is "Kill Password"?
"Kill password" is a 32 bits length data. It is required when we need to kill a Tag permanently. A "dead" Tag no longer response to any command.

Where to write "Kill Password"?
If the required "Kill Password" is 0x12345678, we need to write the 16 most significant bits (MSB) or 0x1234 into RESERVED bank offset 0. This is because the MSB of the password is stored at the lower memory offset in a memory bank. 16 lease significant bits (LSB) of "Kill Password", 0x5678, will be stored at RESERVED bank offset 1.

What is "Access Password"?
"Access password" is a 32 bits length data. It is required when we need to put a Tag into Secured State.

Where to write "Access Password"?
If we need 0x9ABCDEF0 as our "Access Password", we need to write the MSB of "Access Password" or 0x9ABC into RESERVED bank offset 2 and LSB of "Access Password" or 0xDEF0 into RESERVED bank offset 3. We are not allowed to change the offset location of "Kill Password" and "Access Password" because this is defined by the TypeC protocol.

What is the purpose of "CRC-16"?
"CRC-16" or EPC bank offset 0 stores the checksum for "PC" and "EPC" data. It is generated automatically by the Tag. We can only read "CRC-16". Write is not allowed because this memory location has been locked by the Tag manufacturer.

Shall we modify "PC"?
"PC" stands for Protocol Control. The most important information contains in "PC" is the total length in word, "PC" + "EPC". "PC" is a 16 bits data. 5 MSB is used to store the "PC" and "EPC" length. The remaining 11 bits can be zero(ed) in most situation. You should modify "PC" if you want to change the display length of "PC" and "EPC". The following table shows all the possible "PC" values and the corresponding "EPC" length.

"PC" Value	"PC" + "EPC"	"PC" + "EPC" + "CRC-16"
0x0000	1 word (2 bytes)	2 words (4 bytes)
0x0800	2 words (4 bytes)	3 words (6 bytes)
0x1000	3 words (6 bytes)	4 words (8 bytes)

0x1800	4 words (8 bytes)	5 words (10 bytes)
0x2000	5 words (10 bytes)	6 words (12 bytes)
0x2800	6 words (12 bytes)	7 words (14 bytes)
0x3000	7 words (14 bytes)	8 words (16 bytes)
0x3800	8 words (16 bytes)	9 words (18 bytes)
0x4000	9 words (18 bytes)	10 words (20 bytes)
0x4800	10 words (20 bytes)	11 words (22 bytes)
0x5000	11 words (22 bytes)	12 words (24 bytes)
0x5800	12 words (24 bytes)	13 words (26 bytes)
0x6000	13 words (26 bytes)	14 words (28 bytes)
0x6800	14 words (28 bytes)	15 words (30 bytes)
0x7000	15 words (30 bytes)	16 words (32 bytes)
0x7800	16 words (32 bytes)	17 words (34 bytes)
0x8000	17 words (34 bytes)	18 words (36 bytes)
0x8800	18 words (36 bytes)	19 words (38 bytes)
0x9000	19 words (38 bytes)	20 words (40 bytes)
0x9800	20 words (40 bytes)	21 words (42 bytes)
0xA000	21 words (42 bytes)	22 words (44 bytes)
0xA800	22 words (44 bytes)	23 words (46 bytes)
0xB000	23 words (46 bytes)	24 words (48 bytes)
0xB800	24 words (48 bytes)	25 words (50 bytes)
0xC000	25 words (50 bytes)	26 words (52 bytes)
0xC800	26 words (52 bytes)	27 words (54 bytes)
0xD000	27 words (54 bytes)	28 words (56 bytes)
0xD800	28 words (56 bytes)	29 words (58 bytes)
0xE000	29 words (58 bytes)	30 words (60 bytes)
0xE800	30 words (60 bytes)	31 words (62 bytes)
0xF000	31 words (62 bytes)	32 words (64 bytes)
0xF800	32 words (64 bytes)	33 words (66 bytes)

Shall we modify “TID”?

“TID” contains Tad identifier information. A general Tag has default value 0xE200 at TID bank offset 0. You are not recommended to modify the value in this bank.

4. Open State & Secured State

There are 7 states available in TypeC Tag: “Ready”, “Arbitrate”, “Reply”, “Acknowledged”, “Open”, “Secured” and “Killed”. Some states can be treated as virtually exist from user point of view. The highlight in this section is “Open” and “Secured” states.

The access right to each bank can be locked so that accessing data from the bank requires the input of “Access Password”. If a particular bank is unlocked, we can perform read/ write operation freely in the “Open” state. If a particular bank is locked, we can only perform limited read/ write operation at “Secured” state. To put a Tag into “Secured” state, we need to provide an “Access Password”. This has to be done before any read/ write operation.

What we need “Secured” state?

In situation where we need to protect the Tag password (kill or access), we will have to lock the RESERVED bank so that third party will not be able to read or change our password. Once locked, the subsequent access to the RESERVED bank can only be done in the “Secured” state. Putting Tag into “Secured” state required “Access Password”.

Sequences to protect “Access Password”.

1. From a brand new Tag, write “Access Password” to the Tag. This operation is done in “Open” state and no password (or 0x00000000) is required.
2. Lock the right of “Access Password” to writeable from the “Secured” state only.
3. Any further modification of “Access password” is required to be done in “Secured” state where “Access Password” is needed.

Sequences to protect “Kill Password”.

1. From a brand new Tag, Write “Kill Password” and “Access Password” to the Tag.
2. Lock the right of “Kill Password” and “Access Password” to writeable from the “Secured” state only.
3. Further modification of “Kill Password” is required to be done in “Secured” state where “Access password” is needed.

5. Tag features

5 major features are available for Type C Tag: Inventory, Read, Write, Lock and Kill.

5.1 Inventory

Inventory returns the values in the EPC bank in the order of “PC” + “EPC” + “CRC-16”. This is in contrast with the memory layout of EPC bank, where the order is “CRC-16”, followed by “PC”, followed by “EPC”. Another different in the inventory is that Tag takes into account of the value of “PC” and returns the desired “EPC” length.

What is Q value?

Q is an important parameter because it decides the probability of Tag reply in response to an inventory command.

Q	Probability of reply
0	1/1 = 100%
1	1/2 = 50%
2	1/4 = 25%
3	1/8 = 12.5%
4	1/16 = 6.25%
5	1/32 = 3.125%
6	1/64 = 1.5625%
7	1/128 = 0.78125%
... up to 15	…

Largest value of Q is 15. We seldom use very high Q unless we cope with a large Tag population.

Why Q = 0 is not always the best?

If Q = 0 and 2 Tags are placed nearby the Reader, both Tags will give 100% reply rate to the inventory command. Their replies will then clashed with each other and the Reader end up getting wrong reply. If Q = 1, there are chances where only 1 Tag replies at a time. In this situation, the Reader will be able to identify the correct Tag reply. The same theory apply to larger Tag population where higher Q value is needed.

Inventory returned value analysis

If an inventory returns value 3000289833B2DDD9048035050000C231. The 16 MSB or 3000 is the “PC” value, indicates that the total length (“PC” + “EPC” + “CRC-16”) is 8 words or 16 bytes. 289833B2DDD9048035050000 is “EPC” value where it stores our product information. 16 LSB or C231 is the checksum.

Now we change the value of “PC” to 1800. The new inventory will return 1800289833B2DDD9EFC7. “PC” value indicates that the total length of return value is now 5 words. Hence, the “EPC” section was truncated to 3 words or 6

bytes as what we see 289833B2DDD9. We also have a new checksum value EFC7. Please note that 048035050000 does not get deleted because we did not overwrite any information on the “EPC” bank. It shall reappear if we change the “PC” value back to 3000.

5.2 Read

Read operation retrieves 16-bit data from any word address in the memory bank if they are unlocked. You need to provide the following parameters to do a Read.

Parameter	Description
Memory Bank	Select between RESERVED, EPC, TID and USER banks
Address Offset	The word pointer of the corresponding memory bank.
Access Password	If the memory location is not locked, this parameter can be omitted or set to 0x00000000. Otherwise, this should be 32-bit “Access Password”.

5.3 Write

Write operation writes 16-bit data to any word address in the memory bank if they are unlocked. You need to provide the following parameters to do a Write.

Parameter	Description
Memory Bank	Select between RESERVED, EPC, TID and USER banks
Address Offset	The word pointed of the corresponding memory bank.
Access Password	If the memory location is not locked, this parameter can be omitted or set to 0x00000000. Otherwise, this should be 32-bit “Access Password”.
Data	The data to be written in the to the Tag

To check if the Write operation is performed successfully, you can do a read back at the same memory location. Please note that the write distance is shorter than the read distance.

5.4 Lock

You can Lock on “Kill Password”, “Access Password”, EPC bank, TID bank and USER bank. The purpose of Lock is to set the access right in various memory banks to prevent data from being stolen or changed. The pre-requisite of Lock operation is “Access Password” because the lock operation has to be done under “Secured” state. Please provide the Tag with a non-zero “Access Password” before you can do a Lock.

There are 4 possible lock setting for “Kill Password” and “Access Password”.

1. Associated password is readable and writeable from either “Open” or “Secured” states.
2. Associated password is permanently readable and writeable from either “Open” or “Secured” states.
3. Associated password only readable and writeable at “Secured” state but not “Open” state.
4. Associated password is permanently not readable or writeable from any state.

There are 4 possible lock setting for EPC bank, TID bank and USER bank.

1. Associated bank is writeable from either “Open” or “Secured” states.
2. Associated bank is permanently writeable from either “Open” or “Secured” states.
3. Associated bank only writeable from “Secured” state but not “Open” state.
4. Associated bank is permanently not writeable from any state.

5.5 Kill

Kill operation will permanently destroy the Tag. The pre-requisite of Kill is the “Kill Password”. Please refer to other section on how to write a “Kill Password”. The password has to be non-zero or else the Kill operation will end up with an error.

To ensure that the tag is killed successfully, you are advised to do an inventory on the Tag. You should not get any reply from the killed Tag.

End of Document