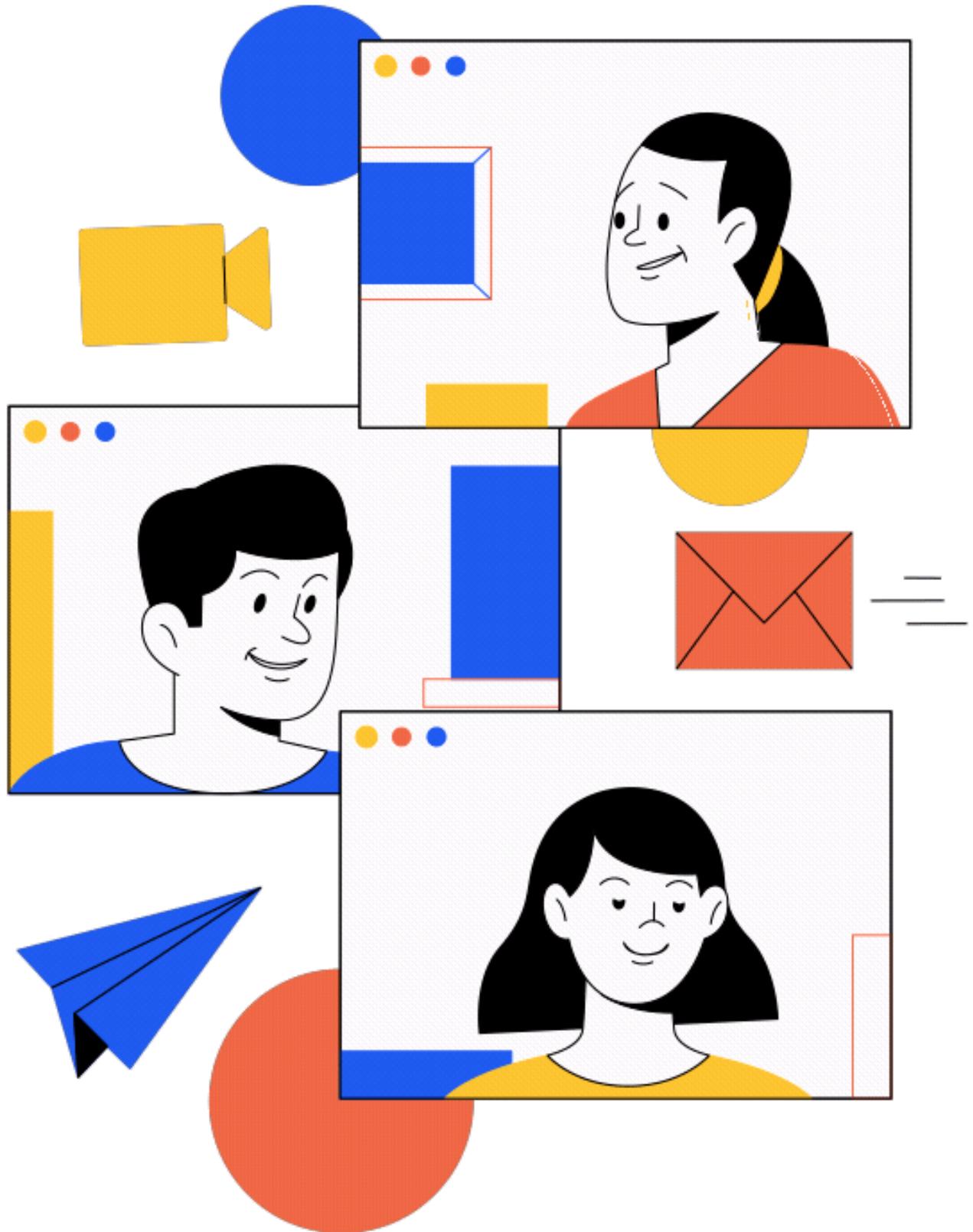


# RFID tag Intro.

**Conductor: Eiko**

2024 August 7th



# Content I

Each RFID tag has 16 sectors and a sector include 4 blocks.  
there are 16 bytes in a block.

Sector	Block	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	AccessBits
15	63	00	00	00	00	00	00	FF	07	80	69	FF	FF	FF	FF	FF	[ 0 0 1 ]	
	62	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	[ 0 0 0 ]	
	61	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	[ 0 0 0 ]	
	60	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	[ 0 0 0 ]	
14	59	00	00	00	00	00	00	FF	07	80	69	FF	FF	FF	FF	FF	[ 0 0 1 ]	
	58	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	[ 0 0 0 ]	
	57	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	[ 0 0 0 ]	
	56	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	[ 0 0 0 ]	
13	55	00	00	00	00	00	00	FF	07	80	69	FF	FF	FF	FF	FF	[ 0 0 1 ]	
	54	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	[ 0 0 0 ]	
	53	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	[ 0 0 0 ]	
	52	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	[ 0 0 0 ]	
12	51	00	00	00	00	00	00	FF	07	80	69	FF	FF	FF	FF	FF	[ 0 0 1 ]	
	50	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	[ 0 0 0 ]	
	49	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	[ 0 0 0 ]	
	48	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	[ 0 0 0 ]	



# Content II

Block 0 is RFID tag which cannot be modified.

0	3	00	00	00	00	00	00	FF	07	80	69	FF	FF	FF	FF	FF	FF	[	0	0	1	]
2	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	[	0	0	0	]
1	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	[	0	0	0	]
0	33	BF	6A	AC	4A	08	04	00	62	63	64	65	66	67	68	69	69	[	0	0	0	]



# Content III

The third block of a sector is called control block.

Byte 0 to Byte 5 are key A (default value is 0xFFFFFFFFFFFF )

Byte 10 to Byte 15 are key B (default value is 0xFFFFFFFFFFFF)

Byte 6 to Byte 8 are access Byte (default value is 0XFF0780)

To write data, we have to get both key A and key B.

Sector	Block	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	AccessBits
15	63	00	00	00	00	00	00	FF	07	80	69	FF	FF	FF	FF	FF	[ 0 0 1 ]	
	62	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	[ 0 0 0 ]	
	61	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	[ 0 0 0 ]	
	60	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	[ 0 0 0 ]	



# Code

```
// Prepare the key (used both as key A and as key B)
// using FFFFFFFFFFFFFFh which is the default at chip delivery from the factory
for (byte i = 0; i < 6; i++) {
    key.keyByte[i] = 0xFF;
}
```

```
byte sector          = 15;
byte blockAddr       = 62;
byte dataBlock[]     = {
    0x00, 0x00, 0x00, 0x00, // color
    0x00, 0x00, 0x00, 0x00, // category
    0x00, 0x00, 0x00, 0x00, // material (wight)
    0x00, 0x00, 0x00, 0x00  // (prodece date)
};
byte trailerBlock    = 63;
MFRC522::StatusCode status;
byte buffer[18];
byte size = sizeof(buffer);
```



# Demo

```
Reading data from block 60 ...
Data in block 60:
00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00

Authenticating again using key B...
Writing data into block 60 ...
0A 0A 0A 04 0A 0A 0A 08 0A 0A 0A 0B 0A 0A 0A 0F

Reading data from block 60 ...
Data in block 60:
0A 0A 0A 04 0A 0A 0A 08 0A 0A 0A 0B 0A 0A 0A 0F
Checking result...
Number of bytes that match = 16
Success :-)

Current data in sector:
 15      63  00 00 00 00  00 00 FF 07  80 69 FF FF  FF FF FF FF  [ 0 0 1 ]
 62      00 00 00 00  00 00 00 00  00 00 00 00  00 00 00 00  [ 0 0 0 ]
 61      00 00 00 00  00 00 00 00  00 00 00 00  00 00 00 00  [ 0 0 0 ]
 60      0A 0A 0A 04  0A 0A 0A 08  0A 0A 0B 0A  0A 0A 0A 0F  [ 0 0 0 ]
```



# Spec

**Define:** two types of attribute, one is original attributes the other one is modified attributes.

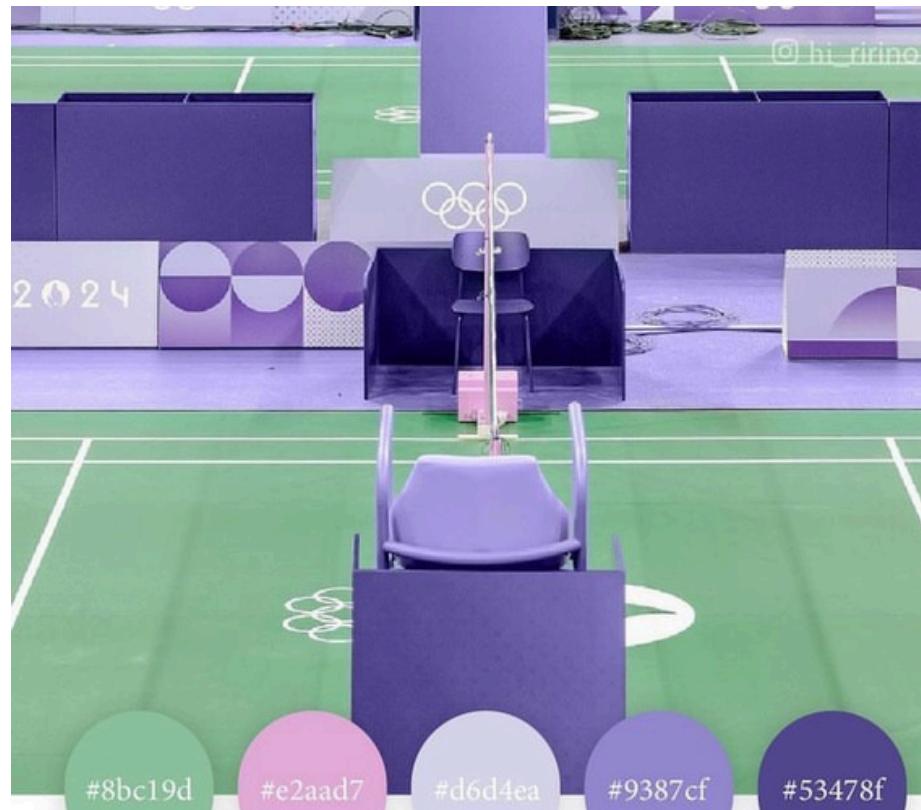
**Define:** original attributes contain color, category, material and produce date four categories and the size of each category is 4 byte.

**First, we have to write data into the 62th block and then we will continuously update the information in the 61th block.**



# Spec (Color)

Sector	Block	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	AccessBits
15	63	00	00	00	00	00	00	FF	07	80	69	FF	FF	FF	FF	FF	[ 0 0 1 ]	
	62	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	[ 0 0 0 ]	
	61	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	[ 0 0 0 ]	
	60	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	[ 0 0 0 ]	



# Spec (Color)

```
dark=0;  
if(red<128) dark++;  
if(green<128) dark++;  
if(blue<128) dark++;
```

#000000
#272727
#3C3C3C
#4F4F4F
#5B5B5B
#6C6C6C
#7B7B7B

#930000
#AE0000
#CE0000
#EA0000
#FF0000
#FF2D2D
#FF5151
#FF7575

#FF95CA
#FFAAD5
#FFC1E0
#FFD9EC
#FFECF5
#FFF7FB



# Spec (Category)

underware=0;

if(isunderware) underware=1;

Sector	Block	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	AccessBits
15	63	00	00	00	00	00	00	FF	07	80	69	FF	FF	FF	FF	FF	[ 0 0 1 ]	
	62	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	[ 0 0 0 ]	
	61	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	[ 0 0 0 ]	
	60	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	[ 0 0 0 ]	

[女裝首頁](#)

外套類	>	所有外套類
T恤		機能/休閒外套 NEW
襯衫/設計上衣		Miracle Air 西裝系列/風衣
針織衫/開襟外套		抗UV/防曬外套
下身類		針織外套
洋裝/裙裝/孕婦類		特級極輕羽絨/厚羽絨



# Spec (Category)

underware=0;

if(isunderware) underware=1;

Sector	Block	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	AccessBits
15	63	00	00	00	00	00	00	FF	07	80	69	FF	FF	FF	FF	FF	[ 0 0 1 ]	
	62	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	[ 0 0 0 ]	
	61	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	[ 0 0 0 ]	
	60	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	[ 0 0 0 ]	

[女裝首頁](#)

外套類	>	所有外套類
T恤		機能/休閒外套 NEW
襯衫/設計上衣		Miracle Air 西裝系列/風衣
針織衫/開襟外套		抗UV/防曬外套
下身類		針織外套
洋裝/裙裝/孕婦類		特級極輕羽絨/厚羽絨



# Spec (Material, Prodece date)

Sector	Block	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	AccessBits
15	63	00	00	00	00	00	00	FF	07	80	69	FF	FF	FF	FF	FF	[ 0 0 1 ]	
	62	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	[ 0 0 0 ]	
	61	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	[ 0 0 0 ]	
	60	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	[ 0 0 0 ]	

Sector	Block	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	AccessBits
15	63	00	00	00	00	00	00	FF	07	80	69	FF	FF	FF	FF	FF	[ 0 0 1 ]	
	62	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	[ 0 0 0 ]	
	61	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	[ 0 0 0 ]	
	60	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	[ 0 0 0 ]	

Sector	Block	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	AccessBits
15	63	00	00	00	00	00	FF	07	80	69	FF	FF	FF	FF	FF	FF	[ 0 0 1 ]	
	62	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	[ 0 0 0 ]	
	61	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	[ 0 0 0 ]	
	60	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	[ 0 0 0 ]	

