Application note

Augmented NDEF messages with ST25TN512 and ST25TN01K devices

Introduction

The purpose of this document is to display practical usage of the ANDEF (augmented NDEF) feature with ST25TN512 and ST25TN01K (hereinafter referred to as ST25TN) devices, whose performance are detailed in DS13433 *NFC Forum Type 2 tag IC with up to 1.6 Kbits of EEPROM*.

NDEF (NFC data exchange format) messages are stored in the EEPROM of ST25TN devices, and contain various kinds of information, such as URI, text, images, Bluetooth® or Wi-Fi® credentials. When reading an NDEF message, NFC compatible smartphones automatically trigger native actions, such as opening the web browser with an URI content, or connecting to a device with Bluetooth pairing data.

ANDEF indicates an NDEF message that contains data dynamically mapped from system registers of an ST25TN device, to enhance the use cases available through native actions.

1 Acronyms

Table 1. List of acronyms

Acronym	Definition
AM	ANDEF memory
ANDEF	Augmented NDEF
CC	Capability container file as defined by the NFC Forum T5T specification
CRC	Cyclic redundancy check
EEPROM	Electrically erasable programmable read-only memory
IC	Integrated circuit
IEC	International electrotechnical commission
ISO	International organization for standardization
NA	Not applicable
NDEF	NFC data exchange format defined by the NFC Forum
NFC	Near field communication
RF	Radio frequency
UID	Unique identifier
UM	User memory
URI	Uniform resource identifier
UTC	Unique tap code

2 Conventions

The following conventions apply in this document unless otherwise stated.

- **Binary number representation**:

  Binary numbers are represented by strings of 0 and 1 digits, with the most significant bit (MSB) on the left, the least significant bit (LSB) on the right, and a **b** suffix added at the end.

  Example: 11110101b

- **Hexadecimal number representation**:

  Hexadecimal numbers are represented by strings of numbers from 0 to 9 and letters from A to F, and an **h** added at the end. The most significant byte (MSB) is shown on the left and the least significant byte (LSB) on the right.

  Example: F5h

- **Decimal number representation**:

  Decimal numbers are represented without any trailing character.

  Example: 245

- **ASCII character representation**:

  Character between single quotes.

  Example: ASCII character **'0'** corresponds to hexadecimal value 30h

3 ANDEF overview

The ANDEF feature allows ST25TN devices to respond smart NDEF messages that are automatically updated to be unique to the device and/or unique to the tap thanks to the automatic insertion of a custom field (such as UID) and of a unique tap code (UTC) without modification of the EEPROM by the user.

When this feature is enabled, user memory data at byte addresses ranging from ANDEF_START to ANDEF_END are replaced by the content of a virtual memory ANDEF_MEM in the response to a READ command.

Figure 1. User and ANDEF memory contents

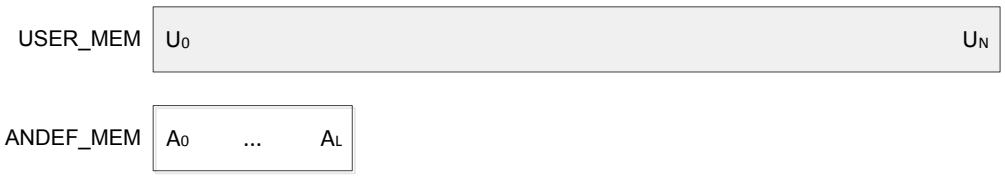
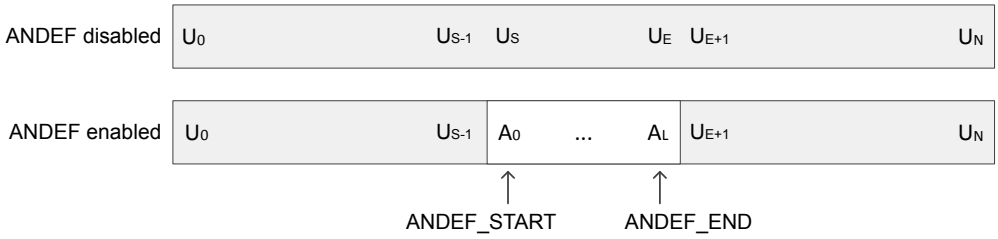


Figure 2. Data responded to READ request when ANDEF is disabled / enabled



In Figure 1 and Figure 2 byte indexes S, E, L and N correspond, respectively, to ANDEF_START, ANDEF_END, ANDEF_LEN - 1 and to the last memory byte.

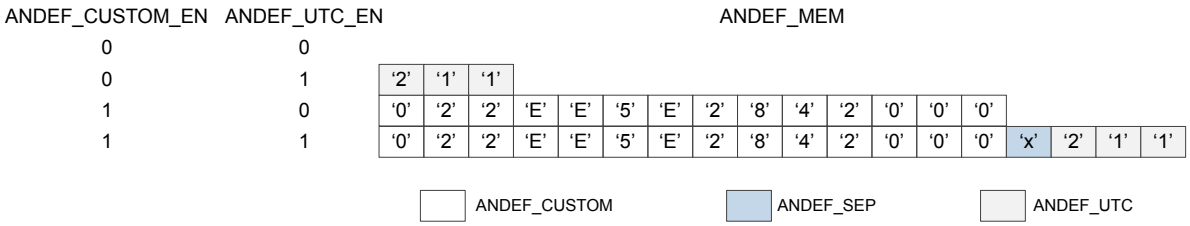
Note: The ANDEF feature has no effect on the WRITE command. When the feature is enabled, and a WRITE command is issued on a block crossing the ANDEF_START to ANDEF_END range, the data from the command is written to the EEPROM at the requested block address, without replacement by volatile memory content.

The values ANDEF_START and ANDEF_END depend on the ANDEF configuration defined in Table 2:

- ANDEF_START = ANDEF_BLOCK * 4 + ANDEF_BYTE
- ANDEF_END = ANDEF_START + ANDEF_CUSTOM_EN * 14 + ANDEF_UTC_EN * 3 + ANDEF_SEP_EN -1
- ANDEF_SEP_EN = 1 if both ANDEF_CUSTOM_EN is 1 and ANDEF_UTC_EN is 1, otherwise ANDEF_SEP_EN = 0

The content of ANDEF_MEM is constructed from a concatenation of up to three fields (custom message such as UID, unique tap code and separator), according to the ANDEF configuration defined in Table 2. An illustration of the possible ANDEF_MEM content is given in Figure 3.

Figure 3. Illustration of ANDEF_MEM content



4 ANDEF settings

4.1 ANDEF configuration

The fields that configure the ANDEF behavior are listed in Table 2.

Table 2. ANDEF_CFG

Block address (hex)	Bits	Name	Function	Factory value
2Eh	b5-b0	ANDEF_BLOCK	Block address for the beginning of ANDEF_MEM	0Fh
	b7-b6	RFU	-	0
	b8	ANDEF_CUSTOM_EN	1b: ANDEF_CUSTOM is appended in ANDEF_MEM 0b: ANDEF_CUSTOM is not appended in ANDEF_MEM	0
	b9	RFU	-	0
	b10	ANDEF_UTC_EN	1b: ANDEF_UTC is appended in ANDEF_MEM 0b: ANDEF_UTC is not appended in ANDEF_MEM	0
	b13-b11	RFU	-	0
	b15-b14	ANDEF_BYTE	Byte number into ANDEF_BLOCK for the beginning of ANDEF_MEM	0

ANDEF feature is disabled when both ANDEF_CUSTOM_EN and ANDEF_UTC_EN are reset to 0.

ANDEF_CFG is always readable.

Modifications of ANDEF_CFG can be prevented according to what detailed in the section dedicated to access restriction in DS13433.

4.2 ANDEF custom

Table 3. ANDEF_CUSTOM

Block address (hex)	Bits	Name	Function	Factory value
3Ch-3Eh	b31-b0	ANDEF_CUSTOM	Field inserted in ANDEF_MEM when ANDEF_CUSTOM_EN = 1	ASCII coding of the device UID in hexadecimal representation
3Fh	b15-b0			

ANDEF_CUSTOM is initialized at factory with ASCII coding of the device UID in hexadecimal representation. Consequently, several ST25TN devices with the same content in T2T_AREA can answer different NDEF messages, each one with a device-specific content.

ANDEF_CUSTOM is always readable.

Modifications of ANDEF_ CUSTOM can be prevented according to what detailed in the section dedicated to access restriction in DS13433.

Since memory is not locked at manufacturing, it is possible to replace the ANDEF_CUSTOM with another custom message.

4.3 ANDEF unique tap code

ANDEF_UTC is an ASCII value generated once every time the device is powered. It is unique to each user tap of the tag, and predictable. The size of ANDEF_UTC is three bytes.

More details on ANDEF_UTC are provided in AN5628 *Unique tap code for ST25TN512 and ST25TN01K devices*. Contact your STMicroelectronics sales office to get this document.

4.4 ANDEF separator

A separator is inserted between ANDEF_CUSTOM and ANDEF_UTC when both ANDEF_CUSTOM_EN and ANDEF_UTC_EN are set to 1. The value of this separator can be customized by writing ANDEF_SEP (see Table 4).

Table 4. ANDEF_SEP

Block address (hex)	Bits	Name	Function	Factory value
3Fh	b23-b16	ANDEF_SEP	Field inserted in ANDEF_MEM when ANDEF_CUSTOM_EN = 1 and ANDEF_UTC_EN = 1	78h (ASCII code of x)

ANDEF_SEP is always readable, modifications can be prevented according to what detailed in the section dedicated to access restriction in DS13433.

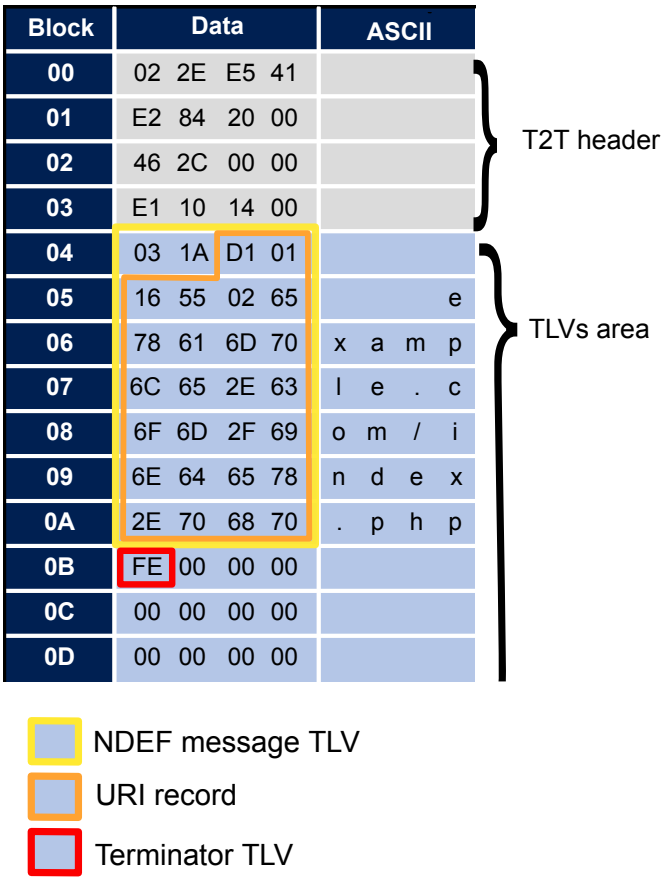
5 ANDEF practical application with URI record

This section shows a practical usage of the ANDEF feature with the example of a dynamic NDEF URI message.

5.1 Basic NDEF URI message

Figure 4 shows the content of the user memory when an ST25TN device is configured as an NFC Forum T2T device containing an NDEF message with one URI record.

Figure 4. Memory content for a basic NDEF URI message



In this basic case the ANDEF feature is not used and the content of the URI record (https://www.example.com/index.php) is static :

- a different URI value must be written in each device to differentiate them from webserver
- the URI read from the same device is always the same at each tap.

5.2 ANDEF URI message

ANDEF features two outstanding properties:

- ANDEF_CUSTOM is prefilled during ST25TN manufacturing with UID in ASCII, unique for each ST25TN device
- ANDEF_UTC is unique to each tap of the tag.

This can be used to generate a tag-specific and tap-specific content in a URI record.

When an ST25TN is tapped with a phone, the phone automatically opens a web browser with a URL providing device information and tap information to the webserver (such as https://www.example.com/andef.php?data=022EE5E2842000x211).

The webserver can then process it to provide accurate content such as specific product information and registration.

5.2.1 ANDEF_MEM content

Figure 5 shows an example of content of ANDEF_MEM when using ANDEF_CUSTOM pre-filled with UID (white cells) and ANDEF_UTC (light gray cells). In this case the separator (light blue cell) is automatically inserted and the size of ANDEF_MEM is 18 bytes.

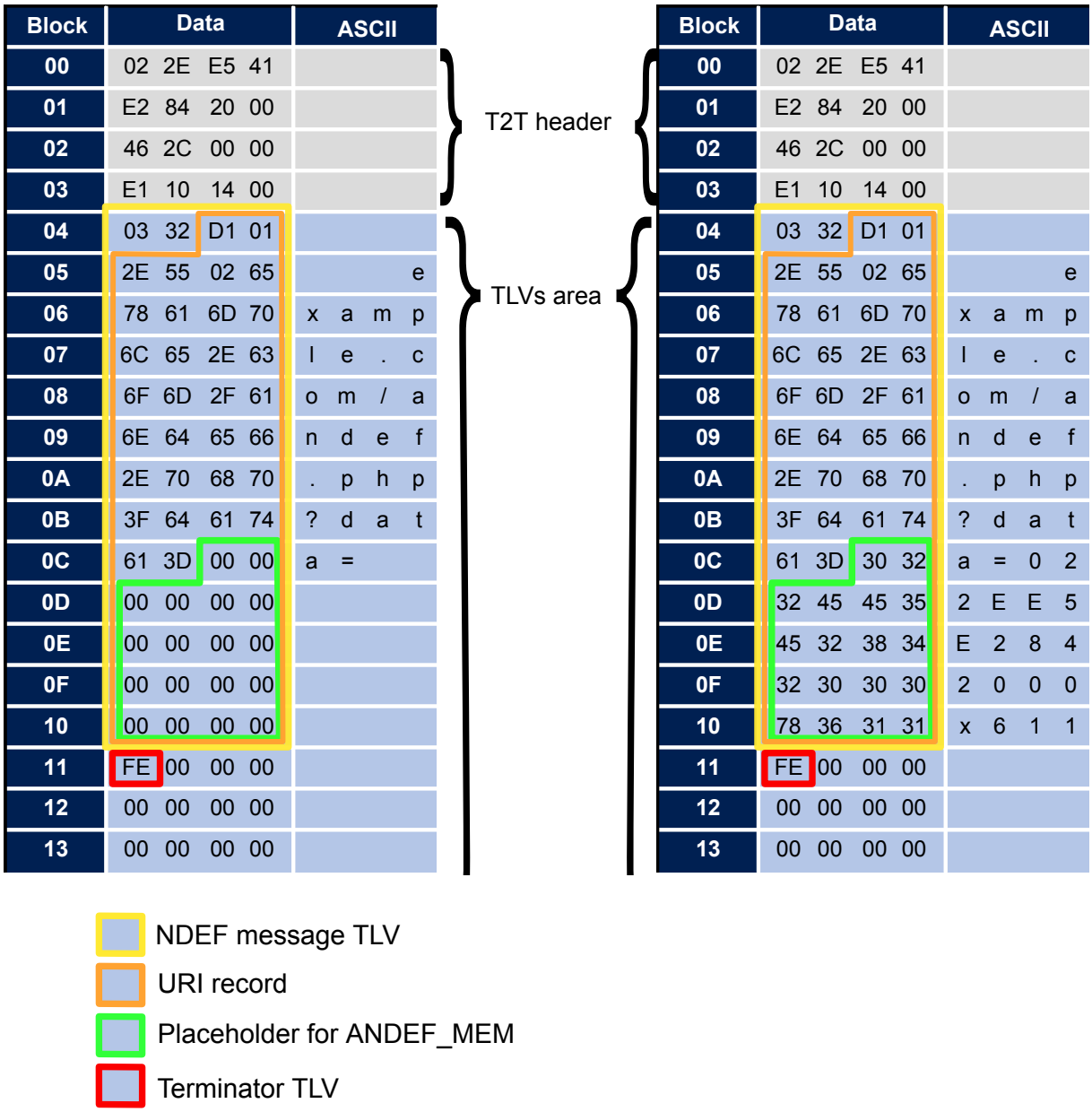
Figure 5. ANDEF_MEM content (ASCII representation)

‘0’	‘2’	‘2’	‘E’	‘E’	‘5’	‘E’	‘2’	‘8’	‘4’	‘2’	‘0’	‘0’	‘0’	‘x’	‘6’	‘1’	‘1’

5.2.2 ST25TN memory content

Figure 6 shows the content of the ST25TN memory (actual content on the left, response to READ command on the right).

Figure 6. **Actual memory content vs. response to READ command**



The actual content (on the left side) is similar to a basic NDEF content, but there is a 18-byte placeholder in the URI record. The size of the NDEF message TLV and the size of URI record must include this placeholder, whose content is not important as it is replaced by the ANDEF_MEM content when read.

The right part of the image shows the memory content as responded to READ commands. This is the content read by the NFC reader. The placeholder content is filled with the content of ANDEF_MEM, and the NFC reader sees the URI record “https://www.example.com/andef.php?data=022EE5E2842000x611”.

To achieve this result ANDEF_CFG is initialized to 850Ch, so that:

- ANDEF_CUSTOM_EN and ANDEF_UTC_EN are enabled
- ANDEF_MEM starts at byte 2 of block 0Ch

5.2.3 Webserver application

Considering the use case of an ANDEF URI message, the webserver targeted by this URI may keep track of the unique tap code of an ST25TN device, and interpret the sequence of values to determine the validity of a product and consequently display appropriate information on the generated webpage.

A demonstration of such webserver application is available on demand. Contact your STMicroelectronics sales office for further information.

5.3 Character set constraint

To maximize phone interoperability an NDEF URI message should only contain the ASCII characters listed in Table 5.

Table 5. List of ASCII characters safe for encoding of URI

Hex	Char	Hex	Char	Hex	Char	Hex	Char	Hex	Char	Hex	Char
-		30h	0	-		50h	P	-		70h	p
21h	!	31h	1	41h	A	51h	Q	61h	a	71h	q
-		32h	2	42h	B	52h	R	62h	b	72h	r
23h	#	33h	3	43h	C	53h	S	63h	c	73h	s
24h	$	34h	4	44h	D	54h	T	64h	d	74h	t
25h	%	35h	5	45h	E	66h	U	65h	e	75h	u
-		36h	6	46h	F	56h	V	66h	f	76h	v
		37h	7	47h	G	57h	W	67h	g	77h	w
28h	(	38h	8	48h	H	58h	X	68h	h	78h	x
29h	)	39h	9	49h	I	59h	Y	69h	i	79h	y
2Ah	*	-		4Ah	J	5Ah	Z	6Ah	j	7Ah	z
2Bh	+			4Bh	K	-		6Bh	k	-	
2Ch	,			4Ch	L	5Ch	\	6Ch	l		
2Dh	-			4Dh	M	-		6Dh	m		
2Eh	.			4Eh	N			6Eh	n	7Eh	~
2Fh	/			4Fh	O	5Fh	_	6Fh	o	-	

Revision history

Table 6. **Document revision history**

Date	Version	Changes
17-Jun-2021	1	Initial release.
02-Sep-2021	2	Changed document classification, from ST restricted to Public.

Contents

List of tables

List of figures