AN4910
Application note

Data exchange between wired (I²C) and wireless (RF ISO 15693) using fast transfer mode supported by ST25DV-I2C Series

Introduction

The ST25DV-I2C Series are dual EEPROM devices designed to be accessed via two different interfaces: a wired I²C interface and a standard contactless ISO 15693 RFID interface.

One of the features offered by the ST25DV-I2C Series is fast data transfer between a hand-held controller (e.g. a phone or an RF reader) and an embedded microcontroller managing a local application.

In this mode, ST25DV-I2C Series act as a mailbox accessed successively by the two interfaces that put in or get a message to exchange data. The mailbox can store up to 256 data bytes.

The purpose of this document is to present the way to activate, control and perform such exchanges using both interfaces to operate.

This application note applies to the ST25DV-I2C Series Dynamic NFC Tags and is now referred to as ST25DV-I2C.

This application note applies to the products listed in *Table 1*.

Table 1. Applicable products

Product type	RPN
ST25DV-I2C Series	ST25DV04K ST25DV16K ST25DV64K ST25DV04KC ST25DV16KC ST25DV64KC

Contents

List of tables

List of figures

1 Acronyms and notational conventions

Table 2. List of acronyms

Acronym	Definition
CRC	Cyclic redundancy check
EEPROM	Electrically-erasable programmable read-only memory
EOF	End of frame
FTM	Fast transfer mode
I^2C	Inter-integrated circuit
ISO/IEC	International organization for standardization / International electrotechnical commission
IT	Interrupt
R	Read
RF	Radio frequency
RFID	Radio frequency identification
RO	Read only
R/W	Read / Write
SOF	Start of frame
W	Write

The following conventions and notations apply in this document unless otherwise stated.

1.1 Product family denomination

In this document, ST25DVxxK is referring to ST25DV04K, ST25DV16K and ST25DV64K products. ST25DVxxKC is referring to ST25DV04KC ST25DV16KC and ST25DV64KC products.

1.2 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

1.3 Hexadecimal number representation

Hexadecimal numbers are represented by strings of numbers from 0 to 9 and letters from A to F, and an ‘h’ suffix 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

1.4 Decimal number representation

Decimal numbers are represented without any trailing character.

Example: 245

2 How to prepare for fast transfer mode

By default, the FTM feature of ST25DV-I2C is disabled (ST25DVxxK: MB_MODE bit 0 of register MB_MODE is set to 0b, ST25DVxxKC: MB_MODE bit 0 of register FTM is set to 0b) so that data can be written in user memory.

A requirement to allow FTM usage is to set MB_MODE to 1b. This operation can be done via the RF or I^2C interface.

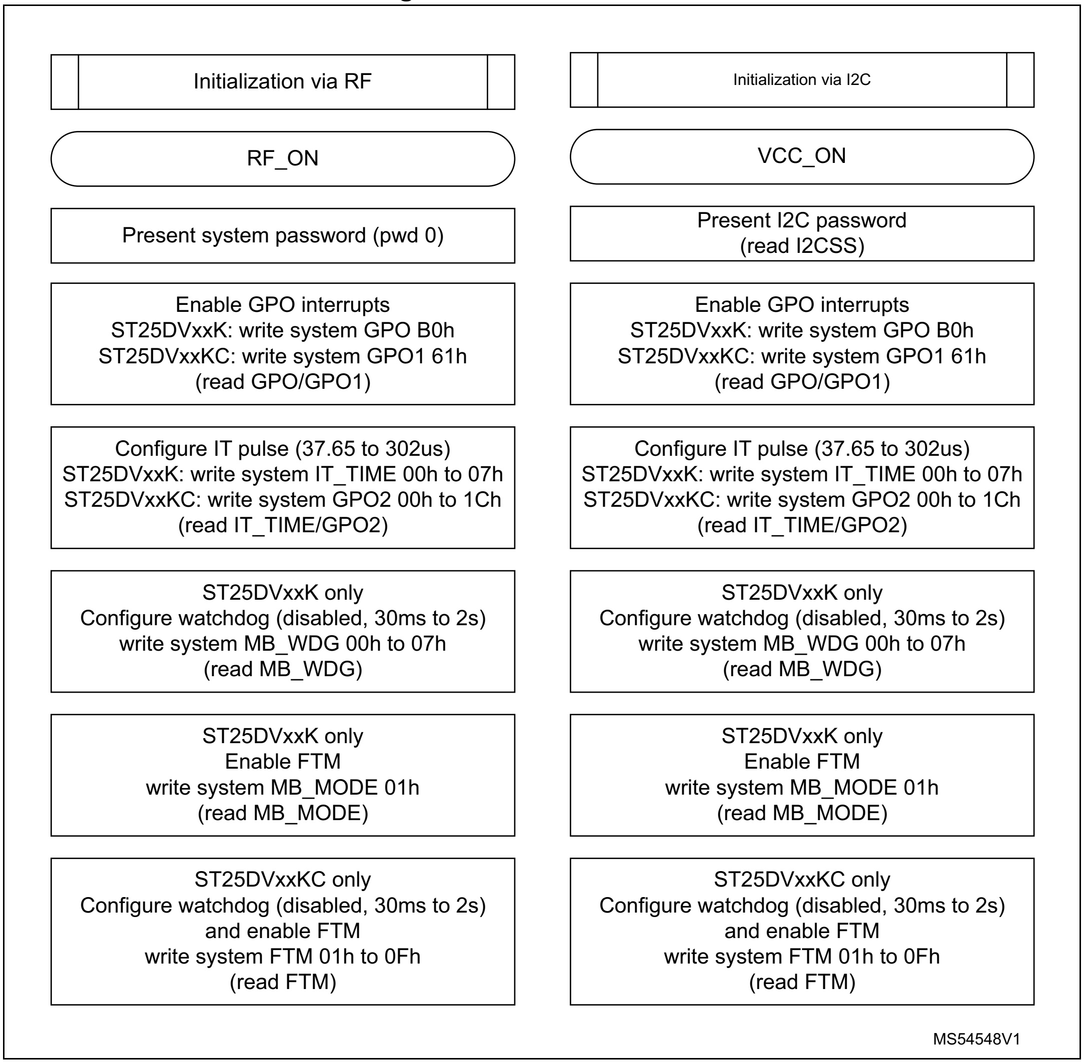
Initially, a super user has to grant access to the system memory where the MB_MODE bit is located.

A secure I^2C session can read register I2C_SSO_Dyn to check that a correct I^2C password was presented.

Other useful static register configurations can be done in this session to optimize future FTM operations:

- RF_PUT_MSG and RF_GET_MSG bits (ST25DVxxK GPO register bits 4 and 5, ST25DVxxKC GPO1 register bits 5 and 6): allows raising dedicated interrupts. IT_TIME (ST25DVxxK IT_TIME register bits [2:0], ST25DVxxKC GPO2 register bits [4:2]): allows configuring pulse duration.
- MB_WDG (ST25DVxxK MB_WDG register bits [2:0], ST25DVxxKC FTM register bits [3:1]): defines the duration after which the message in the mailbox could be overwritten (a value of 00h corresponds to an infinite duration. In that case, only a get or a reset of the FTM allows changing the content of the mailbox)

Figure 1. FTM initialization



2.1 RF sequence to prepare for fast transfer mode

The following table details the RF sequence to be followed to prepare for FTM:

Table 3. RF sequence for FTM preparation (ST25DVxxK)

Command flow	Request frame	Response	Comment
RF Power ON	-	-	-
RF Present System Password (0)	02 B3 02 00 00 00 00 00 00 00 00 00h	00h	Default ST25DV-I2C password is 00 00 00 00 00 00 00 00h
RF Write static register **GPO**	02 A1 02 **00 B0**h	00h	B0h: GPO enabled, RFPutMsg enabled, RFGetMsg enabled
RF Read static register **GPO** (Optional)	02 A0 02 00h	00 B0h	-
RF Write static register **IT_TIME** (Optional)	02 A1 02 **01 03**h	00h	03h: Interruption duration 188 µs
RF Read static register **IT_TIME** (Optional)	02 A0 02 01h	00 03h	-
RF Write static register **MB_WDG** (Optional)	02 A1 02 0E 07h	00h	07h: 2s duration of mailbox Watch Dog
RF Read static register **MB_WDG** (Optional)	02 A0 02 0Eh	00 07h	-
RF Write static register **MB_MODE**	02 A1 02 **0D 01**h	00h	01h: FTM mode allowed
RF Read static register **MB_MODE** (Optional)	02 A0 02 0Dh	00 01h	-

Note: Words in bold are shown for the easiness of reading only.

Table 4. RF sequence for FTM preparation (ST25DVxxKC)

Command flow	Request frame	Response	Comment
RF Power ON	-	-	-
RF Present System Password (0)	02 B3 02 00 00 00 00 00 00 00 00 00h	00h	Default ST25DV-I2C password is 00 00 00 00 00 00 00 00h
RF Write static register **GPO1**	02 A1 02 **00 61**h	00h	61h: GPO enabled, RFPutMsg enabled, RFGetMsg enabled
RF Read static register **GPO1** (Optional)	02 A0 02 00h	00 61h	-
RF Write static register **GPO2**	02 A1 02 **01 0C**h	00h	0Ch: GPO interrupt duration 188 µs
RF Read static register **GPO2** (Optional)	02 A0 02 01h	00 0Ch	-
RF Write static register **FTM** (Optional)	02 A1 02 **0D 0F**h	00h	0Fh: 0Fh: 2s duration of mailbox Watch, FTM allowed
RF Read static register **FTM** (Optional)	02 A0 02 0Dh	00 0Fh	-

Note: Words in bold are shown for the easiness of reading only.

Note: RF operations are reported seen from RF transceiver.
Bytes of the Request frame column represent the commands code sent to ST25DV-I2C.
Bytes of the Response column represent the data returned by ST25DV-I2C.
CRC bytes are not reported.
Words in bold are shown for the easiness of reading only.

2.2 I²C sequence to prepare for fast transfer mode

Table 5 details the I²C sequence to be followed to prepare for FTM using the following abbreviations:

- **Start**: transmit I²C start
- **Stop**: transmit I²C Stop
- **sxx**: send byte xx
- **sAck**: send Acknowledge
- **sNoack**: send No acknowledge
- **rdd**: read byte dd
- **rAck**: read Acknowledge
- **rNoack**: read No acknowledge
- **sA6**: ST25DV-I2C device select for writing in user memory
- **sA7**: ST25DV-I2C device select for reading in user memory
- **sAE**: ST25DV-I2C device select for writing in system memory
- **sAF**: ST25DV-I2C device select for reading in system memory

Table 5. I²C sequence for FTM preparation (ST25DVxxK)

Command flow	Request/Response frame	Polling (optional)	Comment
V_{CC} ON	-	-	DC power
I²C Present Password	Start sAE rAck s09 rAck s00 rAck s00 rAck s00 rAck s00 rAck s00 rAck s00 rAck s00 rAck s00 rAck s00 rAck s09 rAck s00 rAck s00 rAck s00 rAck s00 rAck s00 rAck s00 rAck s00 rAck s00 rAck Stop	Start sAE rAck	Default ST25DV-I2C I²C password is 00 00 00 00 00 00 00 00h (Present Password is immediate)
I²C Read dynamic register **I2C_SSO_Dyn** (optional)	Start sA6 rAck s20 rAck s04 rAck Start sA7 rAck r01 sNoack Stop	-	Confirm that access rights are granted
I²C Write System **GPO**	Start sAE rAck **s00** rAck **s00** rAck **sB0** rAck Stop	Start sAE rNoack Start sAE rNoack ... Start sAE rAck	B0h: GPO enabled, RFPutMsg enabled, RFGetMsg enabled
I²C Read System **GPO** (optional)	Start sAE rAck s00 rAck s00 rAck Start sAF rAck rB0 sNoack Stop	-	-
I²C Write System **IT_TIME**	Start sAE rAck s00 rAck s01 rAck s03 rAck Stop	Start sAE rNoack Start sAE rNoack ... Start sAE rAck	03h: GPO interrupt duration 188 µs
I²C Read System **IT_TIME** (optional)	Start sAE rAck s00 rAck s01 rAck Start sAF rAck r03 sNoack Stop	-	-

Table 5. I^2C sequence for FTM preparation (ST25DVxxK) (continued)

Command flow	Request/Response frame	Polling (optional)	Comment
I^2C Write System **MB_WDG**	Start sAE rAck **s00** rAck **s0E** rAck *s07* rAck Stop	Start sAE rNoack Start sAE rNoack ... Start sAE rAck	07h: 2 s duration of mailbox Warchdog
I^2C Read System **MB_WDG** (optional)	Start sAE rAck s00 rAck s0E rAck Start sAF rAck r07 sNoack Stop	-	-
I^2C Write System **MB_MODE**	Start sAE rAck **s00** rAck **s0D** rAck **s01** rAck Stop	Start sAE rNoack Start sAE rNoack ... Start sAE rAck	01h: FTM allowed
I^2C Read System **MB_MODE** (optional)	Start sAE rAck s00 rAck s0D rAck Start sAF rAck r01 sNoack Stop	-	-

Note: Words in bold are shown for the easiness of reading only.

Table 6. I^2C sequence for FTM preparation (ST25DVxxKC)

Command flow	Request/Response frame	Polling (optional)	Comment
V_{CC} ON	-	-	DC power
I^2C Present Password	Start sAE rAck s09 rAck s00 rAck s00 rAck s00 rAck s00 rAck s00 rAck s00 rAck s00 rAck s09 rAck s00 rAck s00 rAck s00 rAck s00 rAck s00 rAck s00 rAck s00 rAck s00 rAck s00 rAck s00 rAck Stop	Start sAE rAck	Default ST25DV-I2C I^2C password is 00 00 00 00 00 00 00 00h (Present Password is immediate)
I^2C Read dynamic register **I2C_SSO_Dyn** (optional)	Start sA6 rAck s20 rAck s04 rAck Start sA7 rAck r01 sNoack Stop	-	Confirm that access rights are granted
I^2C Write System **GPO1**	Start sAE rAck **s00** rAck **s00** rAck **s61** rAck Stop	Start sAE rNoack Start sAE rNoack ... Start sAE rAck	B0h: GPO enabled, RFPutMsg enabled, RFGetMsg enabled
I^2C Read System **GPO1** (optional)	Start sAE rAck s00 rAck s00 rAck Start sAF rAck r61 sNoack Stop	-	-
I^2C Write System **GP02**	Start sAE rAck s00 rAck s01 rAck s0C rAck Stop	Start sAE rNoack Start sAE rNoack ... Start sAE rAck	0Ch: GPO interrupt duration 188 µs

Table 6. I²C sequence for FTM preparation (ST25DVxxKC) (continued)

Command flow	Request/Response frame	Polling (optional)	Comment
I²C Read System **GPO2** (optional)	Start sAE rAck s00 rAck s01 rAck Start sAF rAck r0C sNoack Stop	-	-
I²C Write System **FTM**	Start sAE rAck **s00** rAck **s0D** rAck **s0F** rAck Stop	Start sAE rNoack Start sAE rNoack ... Start sAE rAck	0Fh: 2 s duration of mailbox Warchdog
I²C Read System **FTM** (optional)	Start sAE rAck s00 rAck s0E rAck Start sAF rAck r07 sNoack Stop	-	-

Note: Words in bold are shown for the easiness of reading only.

Note: I²C operations are reported seen from the master's side.
Words in bold are shown for the easiness of reading only.

Note: It is assumed that I²C slave address of ST25DVxxKC is set to AE/AF for system memory.

3 How to initiate the fast transfer mode

Fast transfer mode requires a valid DC supply which could be checked by reading Dynamic register EH_CTRL_Dyn (bit b3).

FTM can be temporarily enabled or disabled by using the MB_CTRL_Dyn dynamic register and by setting the MB_EN bit to 1b for setting or to 0b for resetting. The MB_EN bit can only be set if MB_MODE was previously set to 1b during the FTM setting phase.

Once FTM is set, it is possible to check the content of the mailbox.

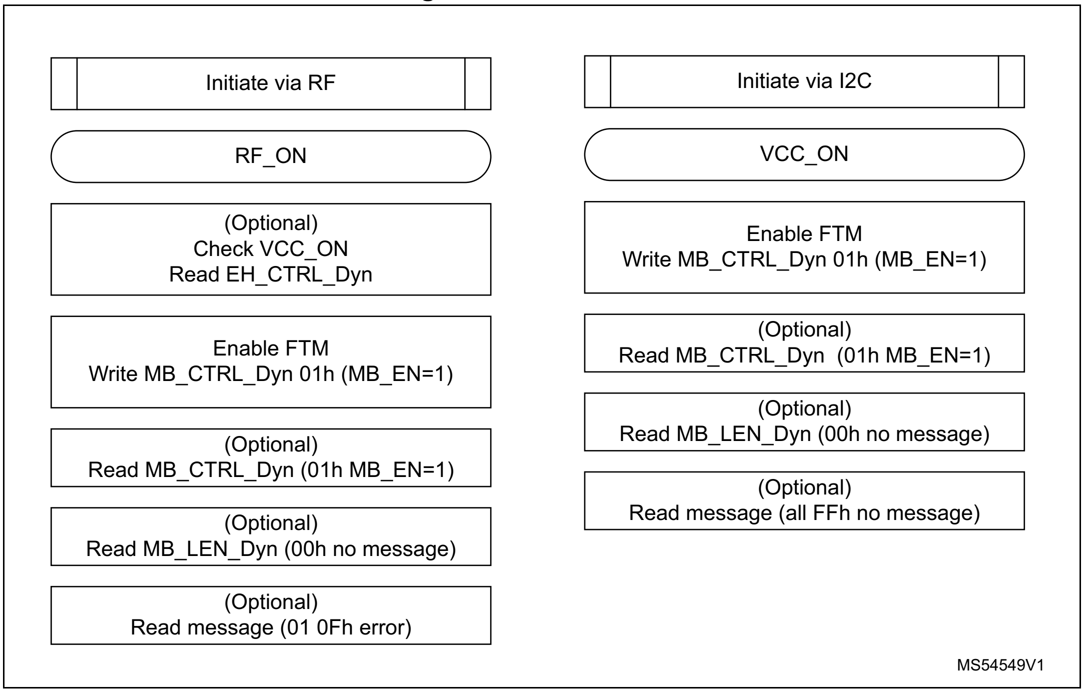
When FTM is reset, the access to dynamic registers, message length or message content, returns the following:

- In RF Read MB_LEN_Dyn & read message returns error code 01 0Fh.
- In I^2C message Length is set to "00" and message content reads FFh.

It is assumed in following figure and tables that the FTM mode as previously been initialized correctly. After initiation:

- The mailbox is empty
- The message length is null
- Read access returns
  - An error in RF
  - FFh in I^2C

Figure 2. FTM initiation



3.1 RF sequence to initiate FTM

Table 7. RF sequence to initiate FTM

Command flow	Request frame	Response	Comment
RF Power ON	-	-	-
RF Read dynamic register **EH_CTRL_Dyn** (optional)	02 AD 02 **02**h	00 0Ch	0Ch: FIELD_ON, VCC_ON
RF Write dynamic Register **MB_CTRL_Dyn**	02 **AE** 02 **0D 01**h	00h	Enable FTM
RF Read dynamic register **MB_CTRL_Dyn** (optional)	02 AD 02 **0D**h	00 01h	FTM enabled
RF Read dynamic register **MB_LEN_Dyn** (optional)	02 **AB** 02h	00 00h	Mailbox empty
RF Read message (optional)	02 AC 02 00 **00**h	01 0Fh	no message

Note: Words in bold are shown for the easiness of reading only.

3.2 I²C sequence to initiate FTM

The codes used in *Table 8* and *Table 9* below are described in *Section 2.2 on page 11*.

Table 8. I²C sequence to initiate FTM

Command flow	Request/Response frame	Polling (optional)	Comment
VCC_ON	-	-	DC Power ON
I²C Write dynamic register **MB_CTRL_Dyn**	Start sA6 rAck **s20** rAck **s06** rAck **s01** rAck Stop	-	Enable FTM
I²C Read dynamic register **MB_CTRL_Dyn** (optional)	Start sA6 rAck **s20** rAck **s06** rAck Start sA7 rAck r01 sNoack Stop	-	FTM enabled
I²C Read dynamic register **MB_LEN_Dyn** (optional)	Start sA6 rAck **s20** rAck **s07** rAck Start sA7 rAck r00 sNoack Stop	-	Mailbox empty
I²C Read message (8 bytes as example) (optional)	Start sA6 rAck **s20** rAck **s08** rAck Start sA7 rAck rFF sAck rFF sAck rFF sAck rFF sAck rFF sAck rFF sAck rFF sNoack Stop	-	No message

Note: Words in bold are shown for the easiness of reading only.

Note: It is assumed that I²C slave address of ST25DVxxKC is set to A6/A7 for user memory.

4 How to be informed of fast transfer mode progress

The MB_CTRL_Dyn dynamic register contains most of the information to understand the state of the FTM. It tells if a message is present in the mailbox, which interface has put this message, and if the addressee has got the message or missed it.

- HOST_PUT_MSG and RF_PUT_MSG bits
  - HOST_PUT_MSG and RF_PUT_MSG indicate which interface has put the message in the mailbox. When set, it is only possible to verify or to get the message; It is not possible to overwrite it.
  - HOST_PUT_MSG and RF_PUT_MSG are reset after the addressee has got the message or missed it or after MB_CTRL_Dyn has been reset.
- HOST_MISS_MSG and RF_MISS_MSG bits
  - HOST_MISS_MSG and RF_MISS_MSG are set after the watchdog's time limit is exceeded. The addressee interface has not got the message.

    Afterwards both interfaces are free to read or overwrite the message.
  - HOST_MISS_MSG and RF_MISS_MSG are reset when resetting MB_CTRL_Dyn or when MB_CTRL_Dyn is read by the interface that missed the message.
- HOST_CURRENT_MSG and RF_CURRENT_MSG bits
  - HOST_CURRENT_MSG and RF_CURRENT_MSG are set when the addressee interface Gets or misses the message. CurrentMsg indicates the origin of the message located in the mailbox. A new message can be put in the mailbox.
  - HOST_CURRENT_MSG and RF_CURRENT_MSG bits are reset when a new message is Put in the mailbox by the other interface or when resetting MB_CTRL_Dyn.

Note: RF handset must poll the MB_CTRL_Dyn dynamic register to detect a new event affecting FTM.

Wired device can be directly informed by Interrupt. For adapted GPO setting a new RF event affecting the mailbox will generate an Interruption pulse on the GPO pin. The Host can read bit b5 (RF_PUT_MSG) or bit b6 (RF_GET_MSG) of the IT_STS_Dyn register. Those bits are set according to the origin of the IT.

Once read, IT_STS_Dyn is reset, ready to indicate a new upcoming event.

5 Control and execution of the fast transfer mode

Only one message is resident in the mailbox. It is loaded as a first-come first-served basis, whether from RF or from I^2C.

After putting a message in the mailbox memory, it is possible to check it but it is not possible to modify it.

Messages put in the mailbox by one interface must be taken by the other one in order for the mailbox to become ready for a new cycle

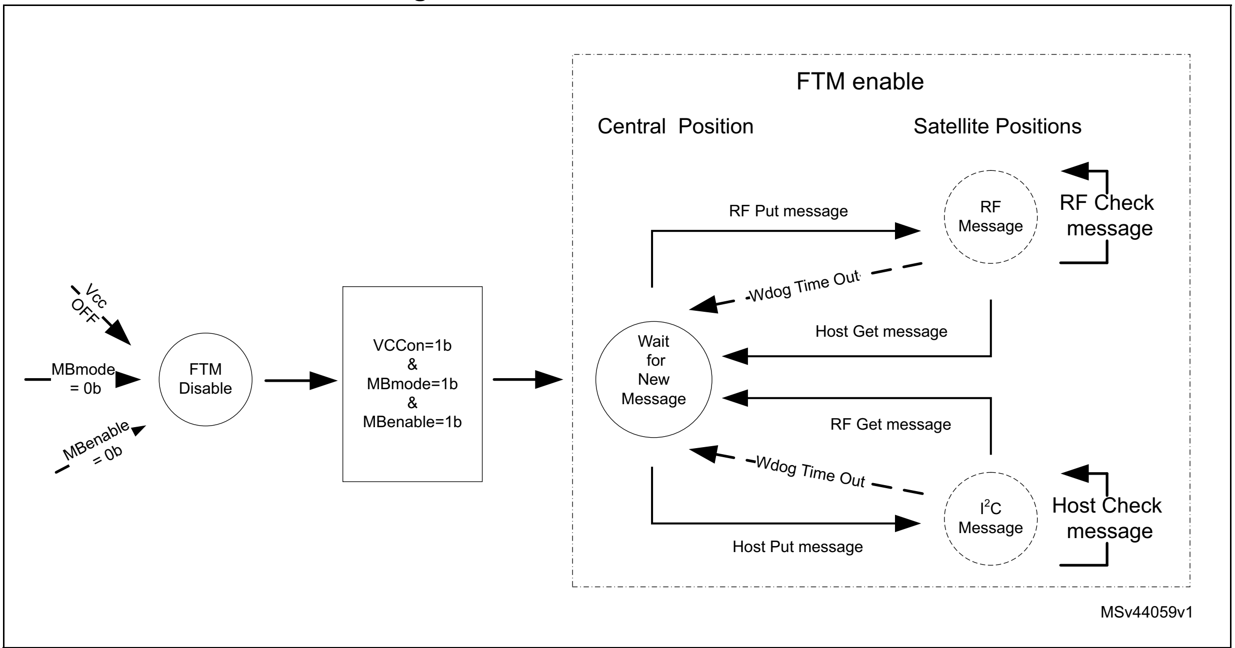
When using a finite MB watchdog duration, the mailbox becomes available again after the watchdog times out. The message in the mailbox then becomes accessible and can be overwritten by any of the two interfaces.

When using an infinite watchdog duration, if you need to modify your message, you must reset the FTM first. The easier way is to set MB_EN bit to 0.

DC supply is mandatory to run in FTM and will guarantee the integrity of messages present in the mailbox. After Putting a new message, the previous one is discarded. The length is temporarily set to zero, then the new message is loaded and the new length is set upon the successful completion of the Put command.

It is not recommended to use Energy Harvesting to power FTM and at the same time supply power to the application circuit.

Figure 3. FTM control and execution



5.1 FTM transmission from RF to I²C

The following sequence lists the main steps of a FTM transmission from RF to I²C. It is further described in *Figure 4* and detailed in *Table 9* and *Table 10*.

1. RF Put message
2. Host detect event
3. Host get message
4. RF poll for message being read by host
5. Repeat from step 1 until all data are transmitted

It is assumed in following figure and tables that the FTM mode as previously been initialized and initiated correctly.

Figure 4. FTM transmission from RF to I²C

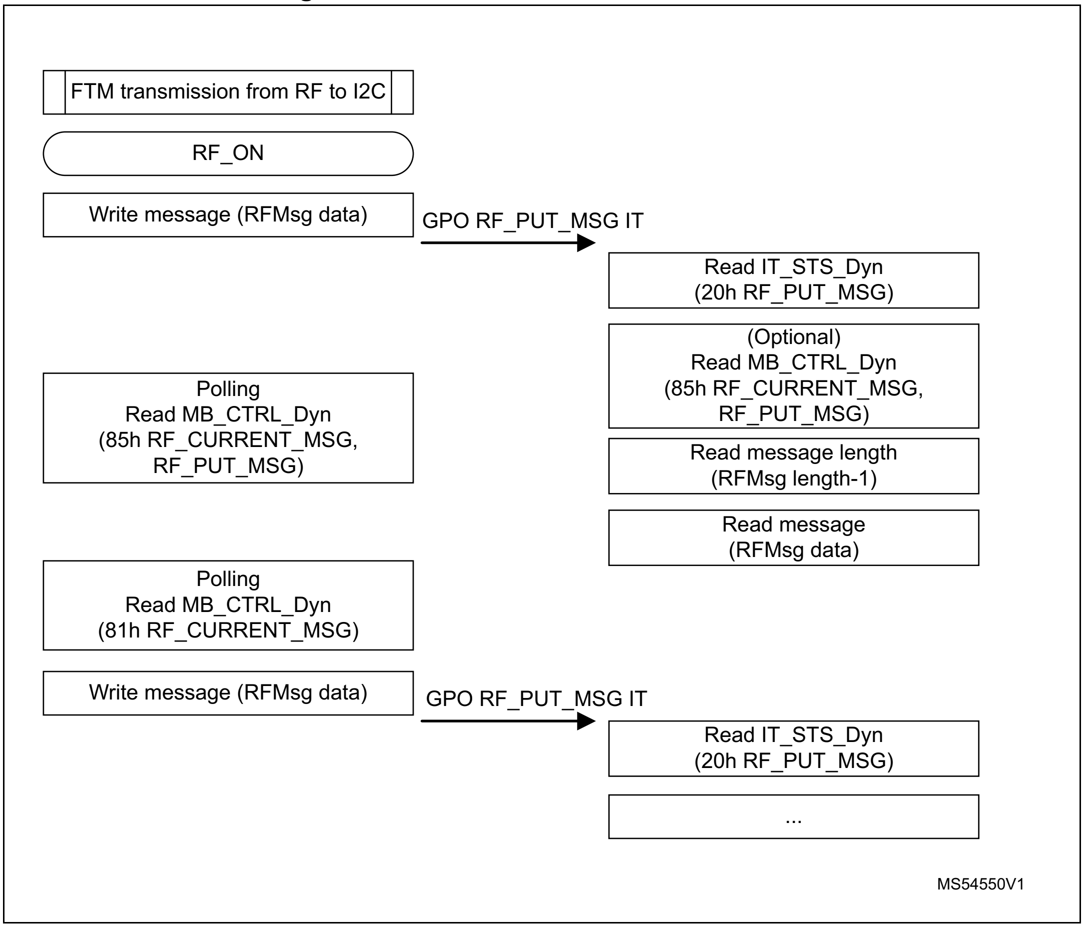


Table 9. FTM transmission from RF to I^2C: RF side

Command flow	Request frame	Response	Comment
RF Power ON	-	-	-
RF Write message (8 Bytes message as example)	02 **AA** 02 07 11 22 33 44 55 66 77 88h	00h	RF put message GPO interrupt RF_PUT_MSG triggered
RF Read dynamic register **MB_CTRL_Dyn**	02 AD 02 **0D**h	00 85h(1)	Polling 85h: RF_CURRENT_MSG, RF_PUT_MSG
I^2C read message			
RF Read dynamic register **MB_CTRL_Dyn**	02 AD 02 **0D**h	00 81h	Polling 81h: RF_CURRENT_MSG, Mailbox is ready for a new sequence
RF Write message (next 8 Bytes message as example)	02 **AA** 02 07 09 AA BB CC DD EE FFh	00h	RF put message GPO interrupt RF_PUT_MSG triggered
Continue until all data are transmitted	...	...	...

1. This command can possibly be answered with error 01 0Fh or not answered if I^2C is busy reading the message.

Note: Words in bold are shown for the easiness of reading only.

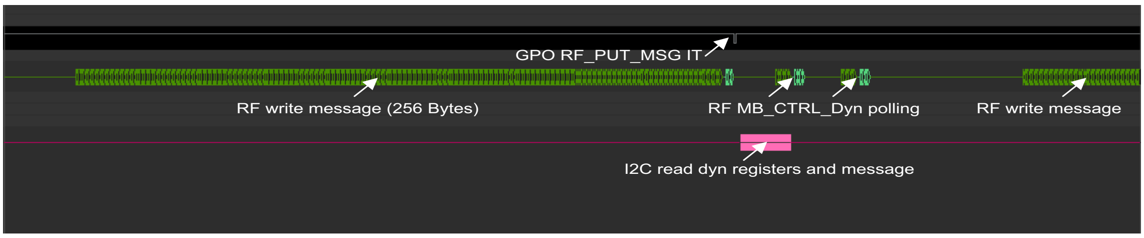
Table 10. FTM transmission from RF to I²C: I²C side

Command flow	Request/Response frame	Polling (optional)	Comment
RF write message			
Interrupt received	-	-	GPO RF_PUT_MSG interrupt
I²C Read dynamic register **IT_STS_Dyn, MB_CTRL_Dyn, MB_LEN_Dyn**	Start sA6 rAck **s20** rAck **s05** rAck Start sA7 rAck r20 sAck r85 sAck r07 sNoack Stop	-	20h: RF_PUT_MSG 85h: RF_CURRENT_MSG, RF_PUT_MSG 07h: 8 Byte message Message data
I²C read message			
I²C read message	Start sA6 rAck **s20** rAck **s08** rAck Start sA7 rAck r11 sAck r22 sAck r33 sAck r44 sAck r55 sAck r66 sAck r77 sAck r88 sNoack Stop	-	8 Bytes message
Read for next message interrupt	...	...	...

Note: Words in bold are shown for the easiness of reading only.

As shown in Table 10, the I²C side can be optimized by reading sequentially in one command the dynamics registers (and eventually the mailbox content).

Figure 5. Capture example of FTM transmission from RF to I²C



5.2 FTM transmission from I²C to RF

The following sequence lists the main steps of a FTM transmission from I²C to RF. It is further described in *Figure 6* and detailed in subsequent tables from *Table 11* and *Table 13*.

1. Host puts message
2. RF polls MB_CTRL_Dyn
3. RF detects host message and gets it
4. Host gets message
5. Host detects RF get message event
6. Repeat from step 1 until all data are transmitted

It is assumed in following figure and tables that the FTM mode as previously been initialized and initiated correctly

Figure 6. FTM transmission from I^2C to RF

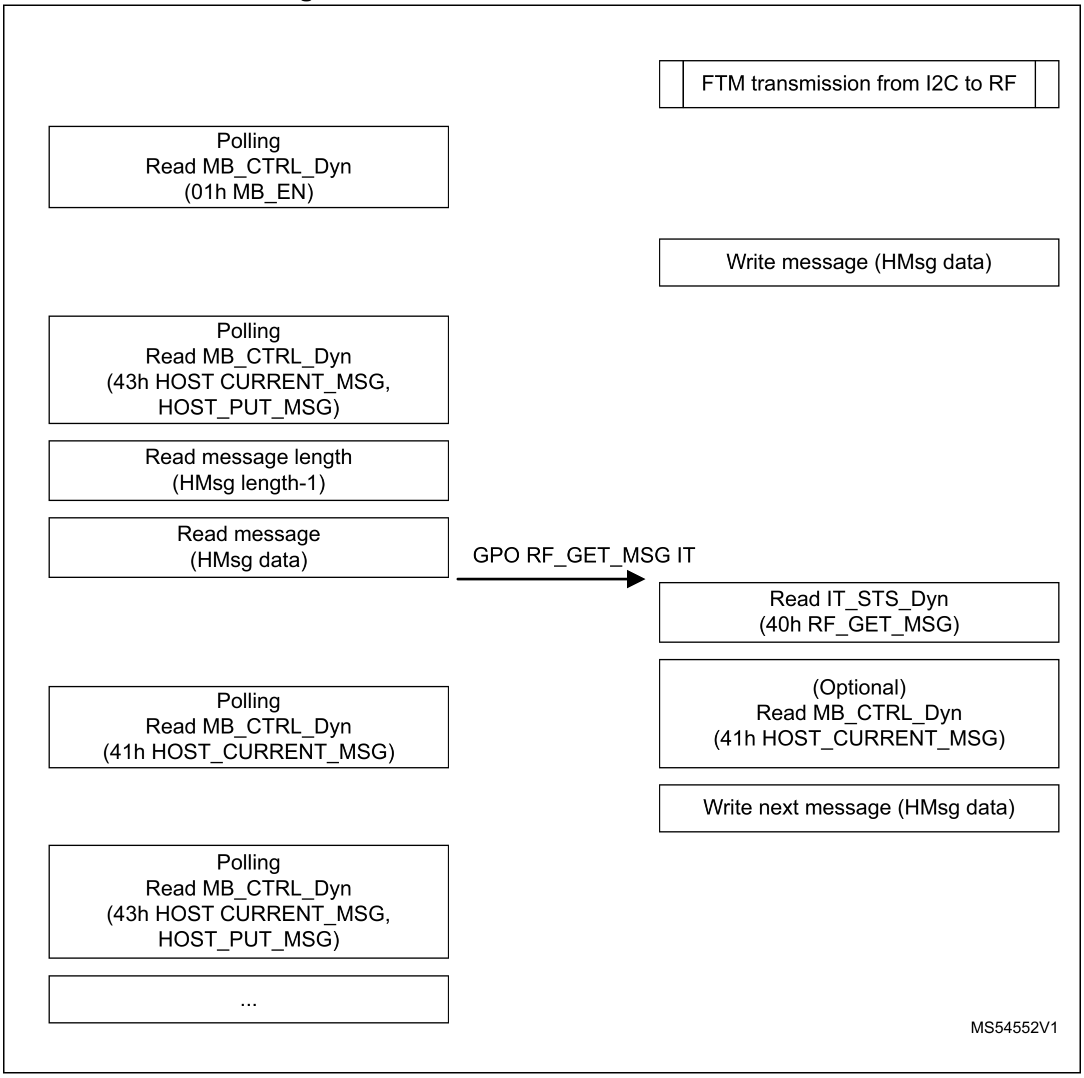


Table 11. FTM transmission from I²C to RF: RF side

Command flow	Request frame	Response	Comment
RF Power ON	-	-	-
RF Read dynamic register **MB_CTRL_Dyn**	02 AD 02 **0D**h	00 01h(1)	Polling 01h: MB_EN
I²C write message			
RF Read dynamic register **MB_CTRL_Dyn**	02 AD 02 **0D**h	00 43h(1)	Polling 43h: HOST_CURRENT_MSG, HOST_PUT_MSG
RF Read dynamic register **MB_LEN_Dyn**	02 AB 02h	00 07h	07h: 8 Bytes message Message data
RF Read message (8 bytes starting byte 00)	02 **AC** 02 00 07h	00 11 22 33 44 55 66 77 88h	8 Bytes message
RF Read dynamic register **MB_CTRL_Dyn**	02 AD 02 **0D**h	00 41h(1)	Polling 41h: HOST_CURRENT_MSG,
Continue until all data are transmitted	...	...	...

1. This command can possibly be answered with error 01 0Fh or not answered if I²C is busy reading the message.

Note: Words in bold are shown for the easiness of reading only.

RF side can be optimized by skipping the read of message length and setting 00h in the “Number of bytes” filed of the Read Message command. By setting 00h in number of bytes to read by the Read Message command, ST25DV-I2C automatically read the entire message. This is illustrated in *Table 12*.

Table 12. FTM transmission from I²C to RF: RF side, full message read

Command flow	Request frame	Response	Comment
RF Power ON	-	-	-
RF Read dynamic register **MB_CTRL_Dyn**	02 AD 02 **0D**h	00 01h(1)	Polling 01h: MB_EN
I²C write message			
RF Read dynamic register **MB_CTRL_Dyn**	02 AD 02 **0D**h	00 43h(1)	Polling 43h: HOST_CURRENT_MSG, HOST_PUT_MSG
RF Read message (full message starting byte 00)	02 AC 02 00 **00**h	00 11 22 33 44 55 66 77 88h	8 Bytes message
RF Read dynamic register **MB_CTRL_Dyn**	02 AD 02 **0D**h	00 41h(1)	Polling 41h: HOST_CURRENT_MSG,
Ready to receive next transmission	...	...	...

1. This command can possibly be answered with error 01 0Fh or not answered if I²C is busy reading the message.

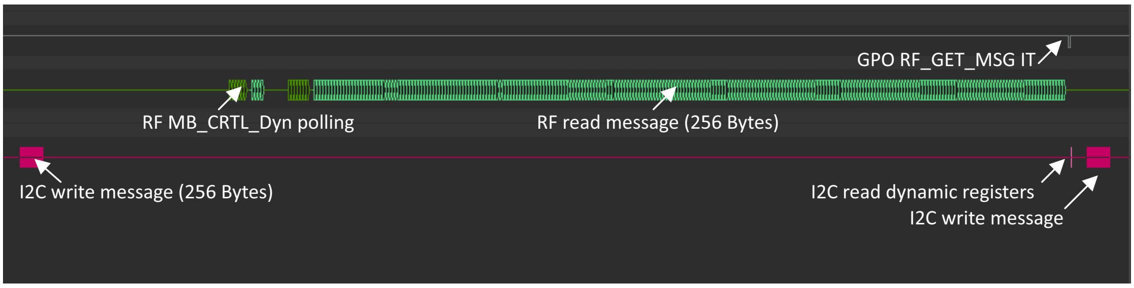
Note: Words in bold are shown for the easiness of reading only.

Table 13. FTM transmission from I²C to RF: I²C side

Command flow	Request/Response frame	Polling (optional)	Comment
I²C Write message	Start sA6 rAck **s20** rAck **s08** rAck s11 rAck s22 rAck s33 rAck s44 rAck s55 rAck s66 rAck s77 rAck s88 rAck Stop	-	8 Bytes message
RF read message			
Interrupt received	-	-	GPO RF_GET_MSG interrupt
I²C Read dynamic register **IT_STS_Dyn, MB_CTRL_Dyn**	Start sA6 rAck **s20** rAck **s05** rAck Start sA7 rAck r40 sAck r41 sNoack Stop	-	40h: RF_GET_MSG 41h: HOST_CURRENT_MSG
I²C read message			
I²C Write next message	Start sA6 rAck **s20** rAck **s08** rAck s99 rAck sAA rAck sBB rAck sCC rAck sDD rAck sEE rAck sFF rAck s00 rAck Stop	-	8 Bytes message
Read for next message interrupt	...	...	...

Note: Words in bold are shown for the easiness of reading only.

Figure 7. Capture example of FTM transmission from I²C to RF



6 How to disable fast transfer mode

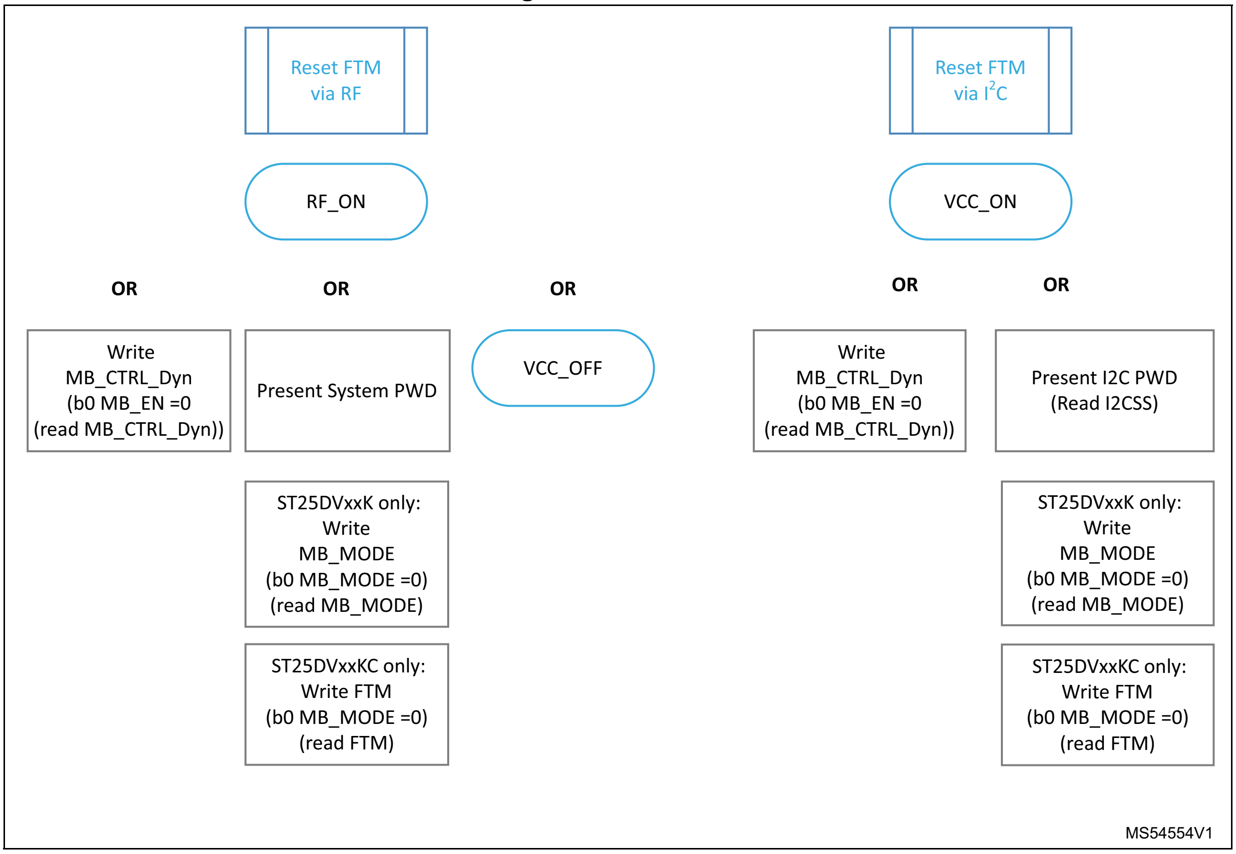
There are three ways to stop FTM:
1. The faster way is to drive the MB_EN dynamic bit to 0b. Return to FTM will be easy by driving MB_EN back to 1b.
2. The second way is to set the MB_MODE system bit to 0b, which requires system access rights but will protect from an unauthorized FTM usage.
3. Finally, switching off DC supply will reset automatically MB_EN to 0b.

Note: Removing the RF field will not affect mailbox content as long as the chip is powered.

When using an infinite watchdog duration, FTM must be reset prior any message modification can be applied. The easier way is to reset MB_EN to 0.

Figure 8 summarizes the different ways to reset FTM which are further detailed in *Section 6.1* and *Section 6.2*.

Figure 8. FTM reset



6.1 RF sequence to reset FTM

1. Reset MB_EN in dynamic register MB_CTRL_Dyn

Table 14. Reset MB_EN in dynamic register MB_CTRL_Dyn (RF sequence)

Command flow	Request frame	Response	Comment
RF Power ON	-	-	-
RF Write dynamic register **MB_CTRL_Dyn**	02 **AE** 02 0D 00h	00h	Disable FTM

Note: Words in bold are shown for the easiness of reading only.

2. Reset MB_MODE in System

Table 15. Reset MB_MODE in system register FTM (RF sequence)

Command flow	Request frame	Response	Comment
RF Present System Password (0)	02 **B3** 02 00 00 00 00 00 00 00 00 00h	00h	Default ST25DV-I2C password is 00 00 00 00 00 00 00 00h
RF Write static register ST25DVxxK**: MB_MODE** ST25DVxxKC: **FTM**	02 **A1** 02 0D 00h	00h	00h: MB_MODE Disable
RF Read static register ST25DVxxK**: MB_MODE** ST25DVxxKC: **FTM**	02 **A0** 02 0Dh	00 00h	Optional

Note: Words in bold are shown for the easiness of reading only.

3. VCC Power OFF

6.2 I^2C sequence to reset FTM

1. Reset MB_EN in dynamic register MB_CTRL_Dyn

Table 16. Reset MB_EN in dynamic register MB_CTRL_Dyn (I^2C sequence)

Command flow	Request/Response frame	Polling (optional)	Comment
Vcc_ON	-	-	DC Power ON
I^2C Write dynamic register **MB_CTRL_Dyn**	**Start sA6** rAck s20 rAck s06 rAck s00 rAck **Stop**	-	Disable FTM
I^2C Read dynamic register **MB_CTRL_Dyn**	**Start sA6** rAck s20 rAck s06 rAck **Start sA7** rAck r00 sNoack **Stop**	-	Disable FTM

Note: Words in bold are shown for the easiness of reading only.

2. Reset MB_MODE in System

Table 17. Reset MB_MODE in System (I^2C sequence)

Command flow	Request/Response frame	Polling (optional)	Comment
I^2C Present System Password	**Start sAE** rAck s09 rAck s00 rAck s00 rAck s00 rAck s00 rAck s00 rAck s00 rAck s00 rAck s00 rAck s00 rAck s09 rAck s00 rAck s00 rAck s00 rAck s00 rAck s00 rAck s00 rAck s00 rAck s00 rAck **Stop**	Start sAE rNoack Start sAE rNoack … Start sAE rAck	Default value on ST25DV-I2C delivery 00 00 00 00 00 00 00 00h Present Password is immediate
I^2C Read dynamic register **I2C_SSO_Dyn**	**Start sA6** rAck s20 rAck s04 rAck **Start sA7** rAck r01 sNoack **Stop**	-	To confirm that access rights are granted (optional)
I^2C Write static register ST25DVxxK**: MB_MODE** ST25DVxxKC: **FTM**	**Start sAE** rAck s00 rAck s0D rAck s00 rAck **Stop**	Start sAE rNoack Start sAE rNoack … Start sAE rAck	Reset to 00h
I^2C Read static register ST25DVxxK**: MB_MODE** ST25DVxxKC: **FTM**	**Start sAE** rAck s00 rAck s0D rAck **Start sAF** rAck r00 sNoack **Stop**	-	Optional

Note: Words in bold are shown for the easiness of reading only.

7 Fast transfer mode efficiency

The efficiency of FTM is mostly driven by the application software rather than by pure ST25DV-I2C performances.

The I^2C and RF interfaces are not equivalent. I^2C can run up to 1 Mbit/s while RF performs only at 26 Kbit/s. When supported by RF handset, ST25DV-I2C can double the speed of RF uplink to 52 Kbit/s using proprietary fast commands set.

ST25DV-I2C offers a large buffer size of 256 bytes which minimizes the cost of protocol overhead versus transmitted data.

Consequently, the results obtained depend on the means in use and on the main transfer direction selected.

Also, RF reader polling period may have an important impact over long transfer that requires several iterations. Polling period must be carefully adjusted: if too small, it can prevent I^2C host from accessing the device rapidly; if too large, unnecessary time may be lost.

For example a 100-KByte firmware can be upgraded to a host in about 47 seconds.

Similarly, a 100-KByte history file can be uploaded to a handset in about 61 seconds when using proprietary fast commands.

8 Example

Refer to the *Firmware for the ST25DV-DISCOVERY boards* user manual (UM2062) for firmware upgrade and picture upload.

9 Revision history

Table 18. Document revision history

Date	Revision	Changes
01-Mar-2017	1	Initial release.
02-Oct-2019	2	Updated *Table 5: I²C sequence for FTM preparation (ST25DVxxK)*. Replaced generic ST25DVxxx and ST25DV with ST25DV-I2C.
25-May-2021	3	Updated: – *Section : Introduction*, *Section 1.1: Product family denomination*, *Section 2: How to prepare for fast transfer mode*, *Section 3: How to initiate the fast transfer mode*, *Section 6: How to disable fast transfer mode*, *Section 7: Fast transfer mode efficiency* – Completely rework *Section 5.1: FTM transmission from RF to I2C*, *Section 5.2: FTM transmission from I²C to RF* – *Figure 1: FTM initialization*, *Figure 4: FTM transmission from RF to I2C*, *Figure 6: FTM transmission from I²C to RF*, *Figure 8: FTM reset* – *Table 3: RF sequence for FTM preparation (ST25DVxxK)*, *Table 5: I²C sequence for FTM preparation (ST25DVxxK)*, *Table 7: RF sequence to initiate FTM*, *Table 8: I²C sequence to initiate FTM*, *Table 15: Reset MB_MODE in system register FTM (RF sequence)*, *Table 16: Reset MB_EN in dynamic register MB_CTRL_Dyn (I²C sequence)*, *Table 17: Reset MB_MODE in System (I²C sequence)* Added *Table 1: Applicable products*, *Table 4: RF sequence for FTM preparation (ST25DVxxKC)*, *Table 6: I²C sequence for FTM preparation (ST25DVxxKC)* Removed: Section 4.1 RF sequence to detect progress in FTM, Section 4.2 I2C sequence to detect progress in FTM, Section 9 Appendix, Table 6. I2C sequence to initiate FTM