AN4910
Application note

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

Introduction

The ST25DV-I2C is a dual EEPROM device 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 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 acts 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 following products: ST25DV04K, ST25DV16K, ST25DV64K.

Contents

List of tables

List of figures

1 Acronyms and notational conventions

Table 1. 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

Product families are abbreviated as follows:

- ST25DV-I2C refers to the following products: ST25DV04K, ST25DV16K, and ST25DV64K.
- The -IE suffix indicates ST25DV-I2C devices with I^2C & GPO Open Drain, fast transfer mode and energy harvesting.
- The -JF suffix indicates ST25DV-I2C devices with I^2C & GPO CMOS, fast transfer mode, energy harvesting and low power mode.

Refer to STMicroelectronics datasheet *Dynamic NFC/RFID tag IC with 4-Kbit, 16-Kbit or 64-Kbit EEPROM and fast transfer mode capability* for details.

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 (MB_MODE bit 0 of register MB_MODE is set to 0b) so that data can be written in user memory.

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

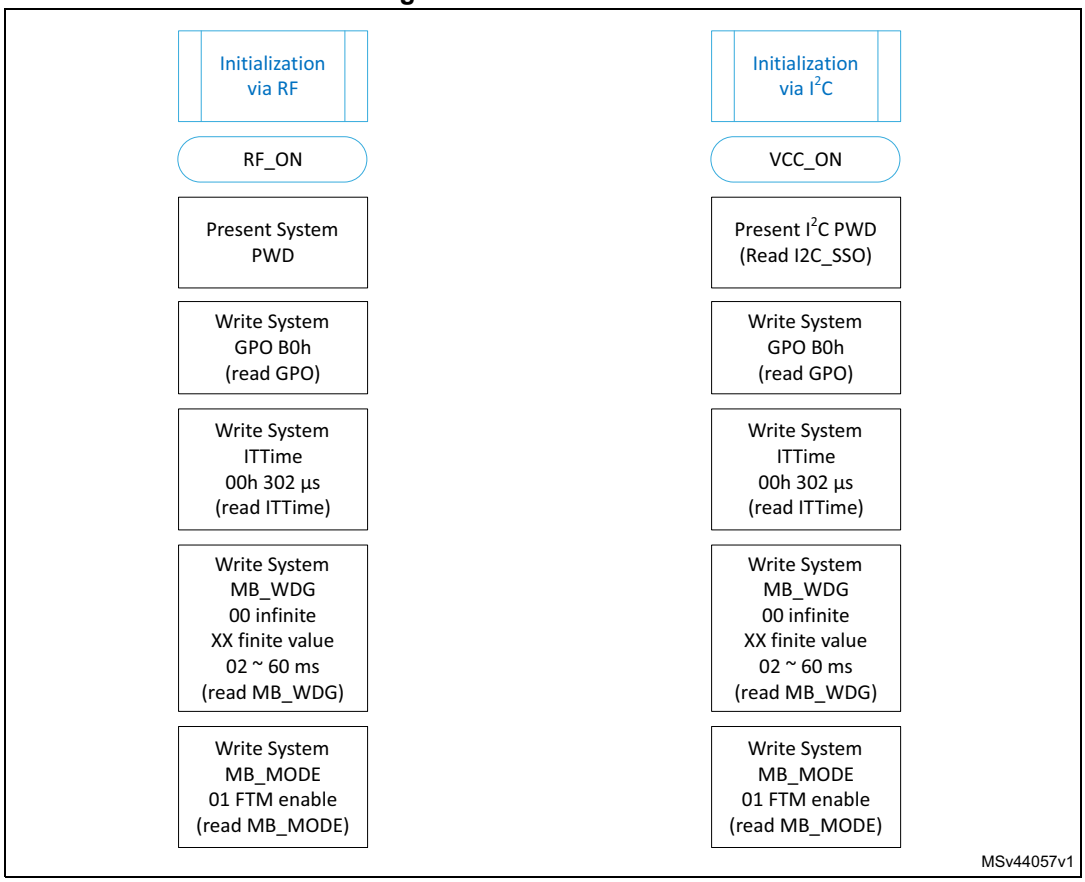
Initially, a super user has to grant access to the system memory where the MB_MODE static register is located.

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:

- GPO register: allows raising dedicated interruptions (RF_PUT_MSG_EN, RF_GET_MSG_EN)
- MB_WDG register: 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 2. RF sequence for FTM preparation

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 B0h	00h	**B0h**: GPO enabled, RF_PUT_MSG enabled, RF_GET_MSG enabled
RF Read Static Register **GPO** (Optional)	02 **A0** 02 00h	00 B0h	-
RF Write Static Register **IT_TIME** (Optional)	02 **A1** 02 01 00h	00h	**00h**: Interruption duration 302 µs
RF Read Static Register **IT_TIME** (Optional)	02 **A0** 02 01h	00 00h	-
RF Write Static Register **MB_WDG** (Optional)	02 **A1** 02 0E 00h	00h	**00h**: Infinite duration of mailbox Watch Dog
RF Read Static Register **MB_WDG** (Optional)	02 **A0** 02 0Eh	00 00h	-
RF Write Static Register **MB_MODE**	02 **A1** 02 0D 01h	00h	**01h**: MB mode enabled
RF Read Static Register **MB_MODE** (Optional)	02 **A0** 02 0Dh	00 01h	-

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 3 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 3. I²C sequence for FTM preparation

Command flow	Request/Response frame	Polling (optional)	Comment
Vcc ON	-	-	DC Power ON
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**	**Start sA6** rAck s20 rAck s04 rAck **Start sA7** rAck r01 sNoack **Stop**	-	Confirm that access rights are granted (optional)
I²C Write System **GPO** (value B0h)	**Start sAE** rAck s00 rAck s00 rAck sB0 rAck **Stop**	**Start sAE** rNoack **Start sAE** rNoack ... **Start sAE** rAck	Set to 00h on ST25DV-I2C delivery. Duration E² programming
I²C Read System **GPO**	Start **sAE** rAck s00 rAck s00 rAck **Start sAF** rAck rB0 sNoack **Stop**	-	Optional
I²C Write System **IT_TIME** (value 00h)	**Start sAE** rAck s00 rAck s01 rAck s00 rAck **Stop**	**Start sAE** rNoack **Start sAE** rNoack ... **Start sAE** rAck	Set to 00h on ST25DV-I2C delivery. Duration E² programming

Table 3. I²C sequence for FTM preparation (continued)

Command flow	Request/Response frame	Polling (optional)	Comment
I²C Read System **IT_TIME**	**Start sAE** rAck s00 rAck s01 rAck **Start sAF** rAck r00 sNoack **Stop**	-	Optional
I²C Write System **MB_WDG** (value 00h)	**Start sAE** rAck s00 rAck s0E rAck s00 rAck **Stop**	**Start sAE** rNoack **Start sAE** rNoack ... **Start sAE** rAck	Set to 00h on ST25DV-I2C delivery. (Infinite duration) Duration E² programming
I²C Read System **MB_WDG**	**Start sAE** rAck s00 rAck s0E rAck **Start sAF** rAck r00 sNoack **Stop**	-	Optional
I²C Write System **MB_MODE** (Value 01h)	**Start sAE** rAck s00 rAck s0D rAck s01 rAck **Stop**	**Start sAE** rNoack **Start sAE** rNoack ... **Start sAE** rAck	Set to 00h on ST25DV-I2C delivery. (Priority access in write user memory) Duration E² programming
I²C Read System **MB_MODE**	**Start sAE** rAck s00 rAck s0D rAck **Start sAF** rAck r01 sNoack **Stop**	-	Optional

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

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.

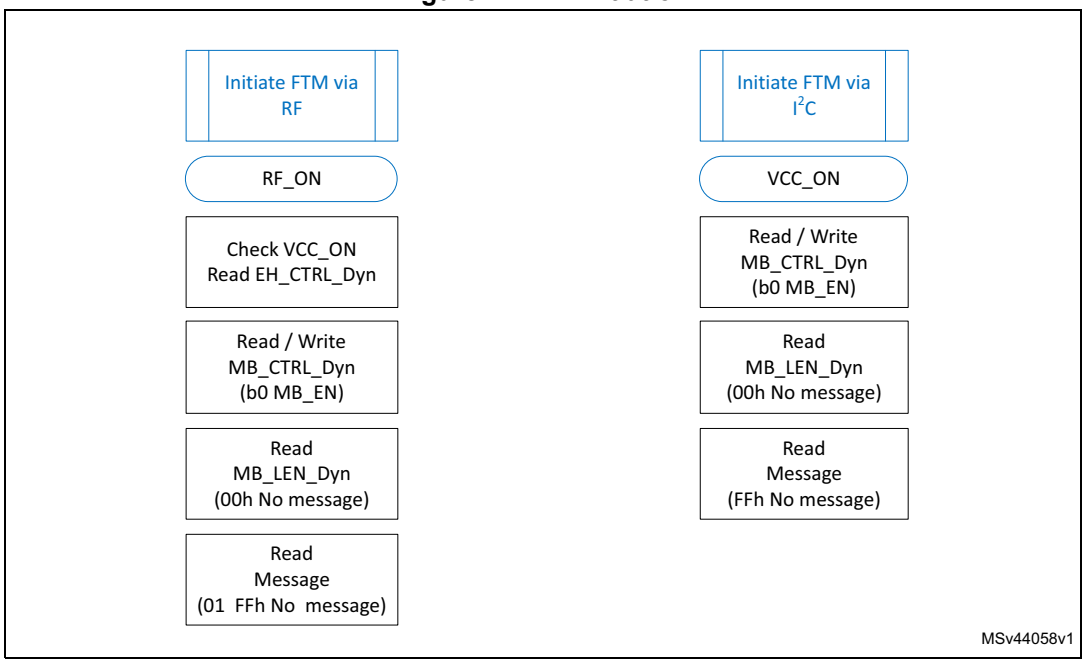
When FTM is reset, the access to registers, message length or message content, returns the following:

- 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.

After initialization:

- 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 4. RF sequence to initiate FTM

Command flow	Request frame	Response	Comment
RF Power ON	-	-	-
RF Read Dynamic Register **EH_CTRL_Dyn**	02 **AD** 02 02	00 0C	b3 Vcc_ON, b2 RF Field_ON
RF Read Dynamic Register **MB_CTRL_Dyn**	02 **AD** 02 0D	00 00	FTM reset
RF Read Message Length	02 **AB** 02	01 0F	Error, FTM reset
RF Read Message	02 **AC** 02 00 00	01 0F	Error, FTM reset
RF Write Dynamic Register **MB_CTRL_Dyn**	02 **AE** 02 0D 01	00	Enable FTM
RF Read Dynamic Register **MB_CTRL_Dyn**	02 **AD** 02 0D	00 01	Enable FTM
RF Read message Length	02 **AB** 02	00 00	Mailbox empty
RF Read Message	02 **AC** 02 00 00	01 0F	Mailbox empty

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

3.2 I²C sequence to initiate FTM

The codes used in *Table 5* and *Table 6* below are described in *Section 2.2 on page 10*.

FTM disabled

Table 5. I²C sequence to initiate FTM (1/2)

Command flow	Request/Response frame	Polling (optional)	Comment
Vcc_ON	-	-	DC Power ON
I²C Read Dynamic Register **MB_CTRL_Dyn**	**Start sA6** rAck s20 rAck s06 rAck **Start sA7** rAck r00 sNoack **Stop**	-	FTM reset
I²C Read Dynamic Register **MB_LEN_Dyn**	**Start sA6** rAck s20 rAck s07 rAck **Start sA7** rAck r00 sNoack **Stop**	-	FTM reset MB_LEN 0
I²C Read Message (16 bytes as example)	**Start sA6** rAck s20 rAck s08 rAck **Start sA7** rAck rFF sAck rFF sAck rFF sAck rFF sAck rFF sAck rFF sAck rFF sAck rFF sAck rFF sAck rFF sAck rFF sAck rFF sAck rFF sAck rFF sAck rFF sAck rFF sNoack **Stop**	-	FTM reset

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

Enable FTM

Table 6. I²C sequence to initiate FTM (2/2)

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**	**Start sA6** rAck	Enable FTM
I²C Read Dynamic Register **MB_CTRL_Dyn**	**Start sA6** rAck s20 rAck s06 rAck **Start sA7** rAck r01 sNoack **Stop**	-	Enable FTM
I²C Read Dynamic Register **MB_LEN_Dyn**	**Start sA6** rAck s20 rAck s07 rAck **Start sA7** rAck r00 sNoack **Stop**	-	No message
I²C Read Message (16 bytes as example)	**Start sA6** rAck s20 rAck s08 rAck **Start sA7** rAck rFF sAck rFF sAck rFF sAck rFF sAck rFF sAck rFF sAck rFF sAck rFF sAck 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.

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.

4.1 RF sequence to detect progress in FTM

Table 7. RF sequence to detect progress in FTM

Command flow	Request frame	Response	Comment
RF Power ON	-	-	-
RF Read Dynamic Register **MB_CTRL_Dyn**	02 **AD** 02 0Dh	00 01h	Enable FTM
RF Read Message Length	02 **AB** 02h	00 00h	Mailbox empty
RF Read Message	02 **AC** 02 00 00h	01 0Fh	Mailbox empty

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

4.2 I²C sequence to detect progress in FTM

When the host receives an IT on GPO, it must check IT_STS_Dyn to know its origin.

Table 8. I²C sequence to detect progress in FTM

Command flow	Request/Response frame	Polling (optional)	Comment
Vcc_ON	-	-	DC Power ON
I²C Read Dynamic Register **IT_STS_Dyn**	**Start sA6** rAck s20 rAck s05 rAck **Start sA7** rAck rdd sNoack **Stop**	-	b5: RF_PUT_MSG b6: RF_GET_MSG
I²C Write Dynamic Register **MB_CTRL_Dyn**	**Start sA6** rAck s20 rAck s06 rAck s01 rAck **Stop**	**Start sA6** rAck	Enable FTM
I²C Read Dynamic Register **MB_LEN_Dyn**	**Start sA6** rAck s20 rAck s06 rAck **Start sA7** rAck r01 sNoack **Stop**	-	Enable FTM
I²C Read Dynamic Register **MB_LEN_Dyn**	**Start sA6** rAck s20 rAck s07 rAck **Start sA7** rAck r00 sNoack **Stop**	-	No message
I²C Read Message (16 bytes as example)	**Start sA6** rAck s20 rAck s08 rAck **Start sA7** rAck rFF sAck rFF sAck rFF sAck rFF sAck rFF sAck rFF sAck rFF sAck rFF sAck 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.

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

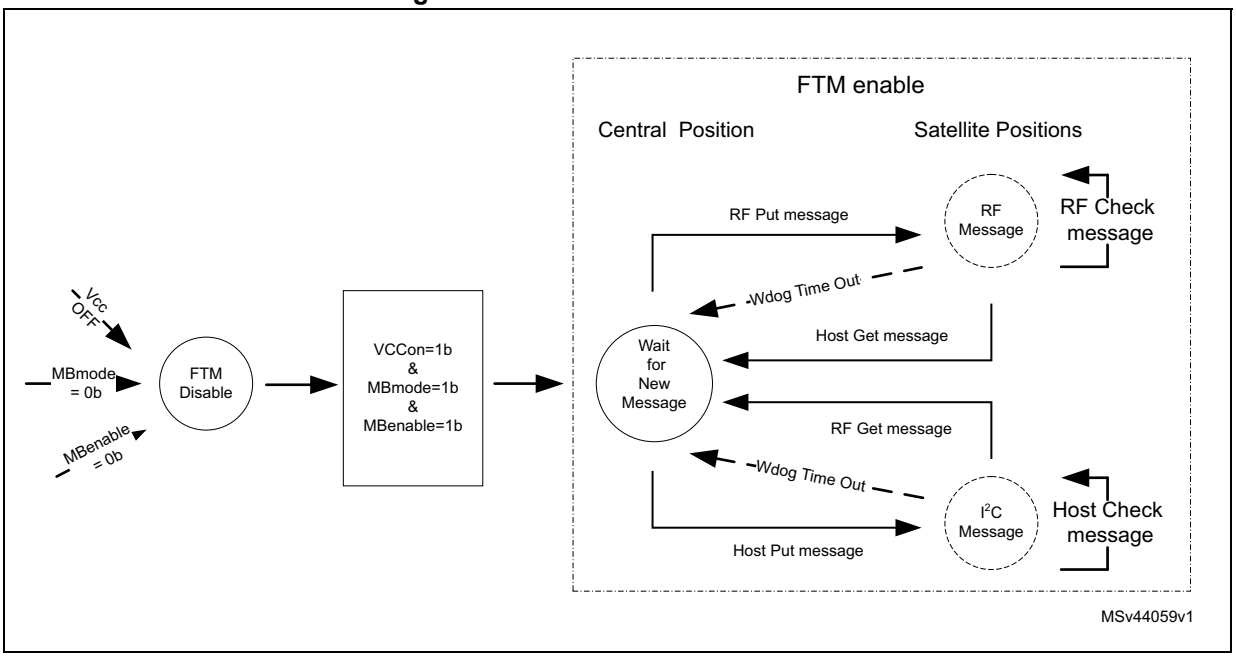
When using a finite MB watchdog duration, the mailbox becomes available again after the watchdog times out. The message in the mailbox then becomes accessible and can be overwritten by any of the two interfaces.

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 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 RF sequence: example of FTM driven by RF

The following sequence lists the main steps of a RF driven FTM. It is further described in *Figure 4* and detailed in the subsequent tables from *Table 9* to *Table 13*.

1. RF Put message
2. Host detect event
3. Host Get message
4. Host Put a message
5. RF poll and detect Host message
6. RF Get message

Figure 4. FTM execution driven by RF

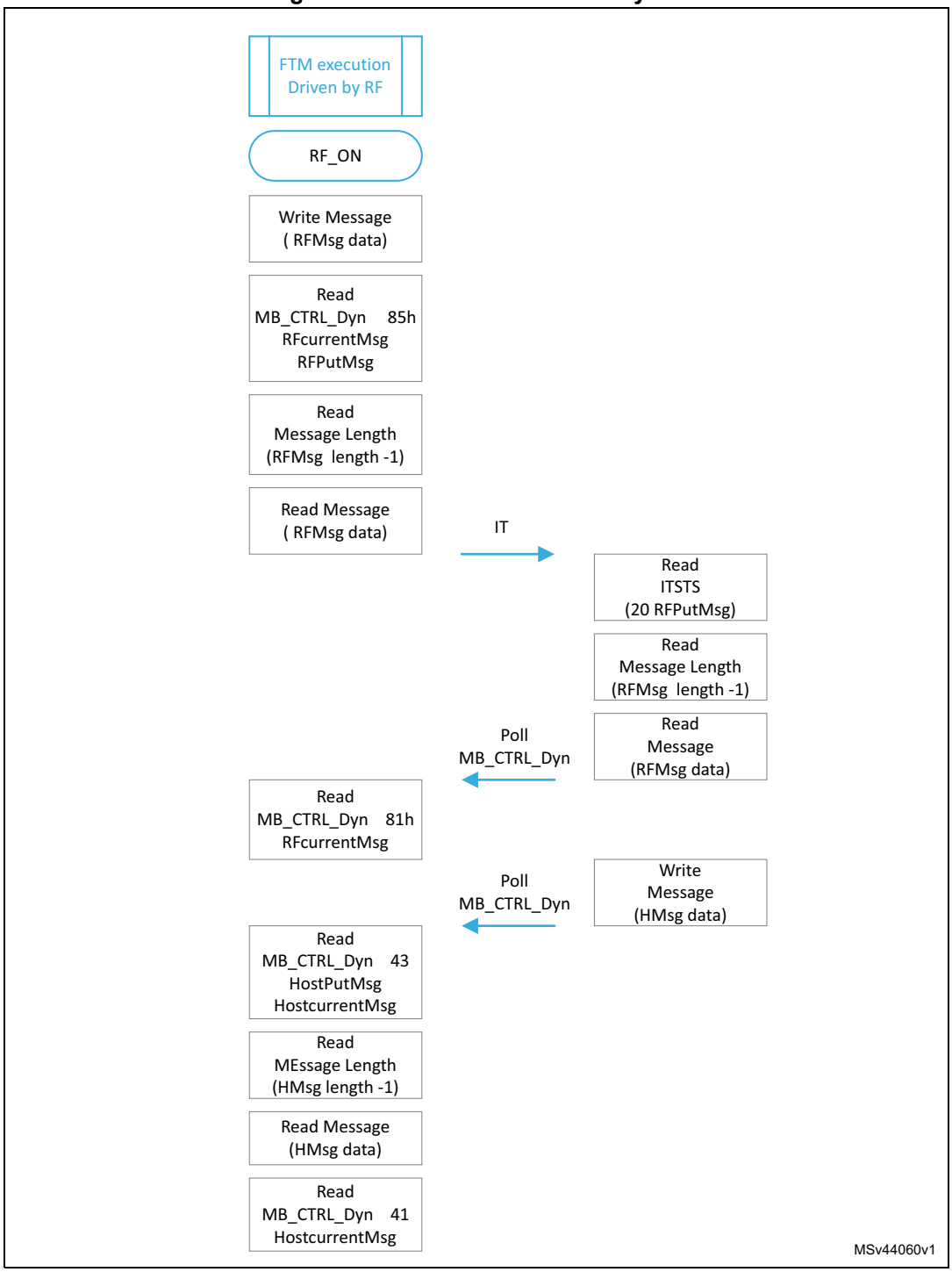


Table 9. FTM execution driven by RF (1/5)

Command flow	Request frame	Response	Comment
RF Power ON	-	-	-
RF Write Message	02 **AA** 02 07 BF212C0123456789	00h	RF put message1
RF Read Dynamic Register **MB_CTRL_Dyn**	02 **AD** 02 0D	00 85h	RF_CURRENT_MSG RF_PUT_MSG MB_EN
RF Read Message Length	02 **AB** 02	00 07h	MB_LEN 7 8 data bytes
RF Read Message	02 **AC** 02 00 00	<- 00 BF21IC0 123456789	RF check message1
RF Write Message	02 **AA** 02 05 BF212CAAAAAA	01 0Fh	**Fail RF message already present** (RF put message2)

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

Break: Host receives interrupt and reads message

Table 10. FTM execution driven by RF (2/5)

Command flow	Request/Response frame	Polling (optional)	Comment
Vcc_ON	-	-	DC Power ON
I^2C Read Dynamic Register **IT_STS_Dyn**	**Start sA6** rAck s20 rAck s05 rAck **Start sA7** rAck r20 sNoack **Stop**	-	20h: RF_PUT_MSG
I^2C Read Dynamic Register **MB_LEN_Dyn**	**Start sA6** rAck s20 rAck s07 rAck **Start sA7** rAck r07 sNoack **Stop**	-	MB_LEN 07h
I^2C Read Message (16 bytes as example)	**Start sA6** rAck s20 rAck s08 rAck **Start sA7** rAck rBF sAck r21 sAck r2C sAck r01 sAck r23 sAck r45 sAck r67 sAck r89 sNoack **Stop**	-	Read RF message1

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

Break: RF Poll MB_CTRL_Dyn to detect then Host Get message

Table 11. Execution driven by RF (3/5)

Command flow	Request frame	Response	Comment
RF Power ON	-	-	-
RF Read Dynamic Register **MB_CTRL_Dyn**	02 **AD** 02 0D	00 81h	RF_CURRENT_MSG MB_EN

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

Break: Host puts message

Table 12. Execution driven by RF (4/5)

Command flow	Request/Response frame	Polling (optional)	Comment
Vcc_ON	-	-	DC Power ON
I^2C Read Dynamic Register **IT_STS_Dyn**	**Start sA6** rAck s20 rAck s05 rAck **Start sA7** rAck r00 sNoack **Stop**	-	-
I^2C Write Message (16 bytes as example)	**Start sA6** rAck s20 rAck s08 rAck sB0 rAck sB1 rAck sB2 rAck sB3 rAck sB4 rAck sB5 rAck sB6 rAck sB7 rAck sB8 rAck sB9 rAck sBA rAck sBB rAck sBC rAck sBD rAck sBE rAck sBF rAck sC0 rAck sC1 rAck sC2 rAck sC3 rAck **Stop**	**Start sA6** rAck	-

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

Break: RF Poll MB_CTRL_Dyn to detect then Host Get message.

Table 13. Execution driven by RF (5/5)

Command flow	Request frame	Response	Comment
RF Power ON	-	-	-
RF Read Dynamic Register **MB_CTRL_Dyn**	02 **AD** 02 0D	00 43h	HOST_PUT_MSG HOST_CURRENT_MSG MB_EN
RF Read Message Length	02 **AB** 02	00 13h	MB_LEN 13 20 data bytes

Table 13. Execution driven by RF (5/5) (continued)

Command flow	Request frame	Response	Comment
RF Read Message	02 **AC** 02 00 00	00 B0B1B2B3B4B5B6B7B8B9 BABBBCBDBEBEC0C1C2C3	RF Get Host message1 (MB_LEN 00, Full message received)
RF Read Dynamic Register **MB_CTRL_Dyn**	02 **AD** 02 0D	<- 00 41h	Mailbox is ready for a new sequence (HOST_CURRENT_MSG; MB_EN)

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

5.2 I²C sequence: example of FTM driven by I²C

The following sequence lists the main steps of a I²C driven FTM. It is further described in *Figure 5* and detailed in the subsequent tables from *Table 14* to *Table 18*.

1. Host puts message
2. RF polls MB_CTRL_Dyn
3. RF detects Host Message and gets it
4. Host gets message
5. Host puts a message
6. RF polls, detects Host message and gets it

Figure 5. FTM execution driven by I²C

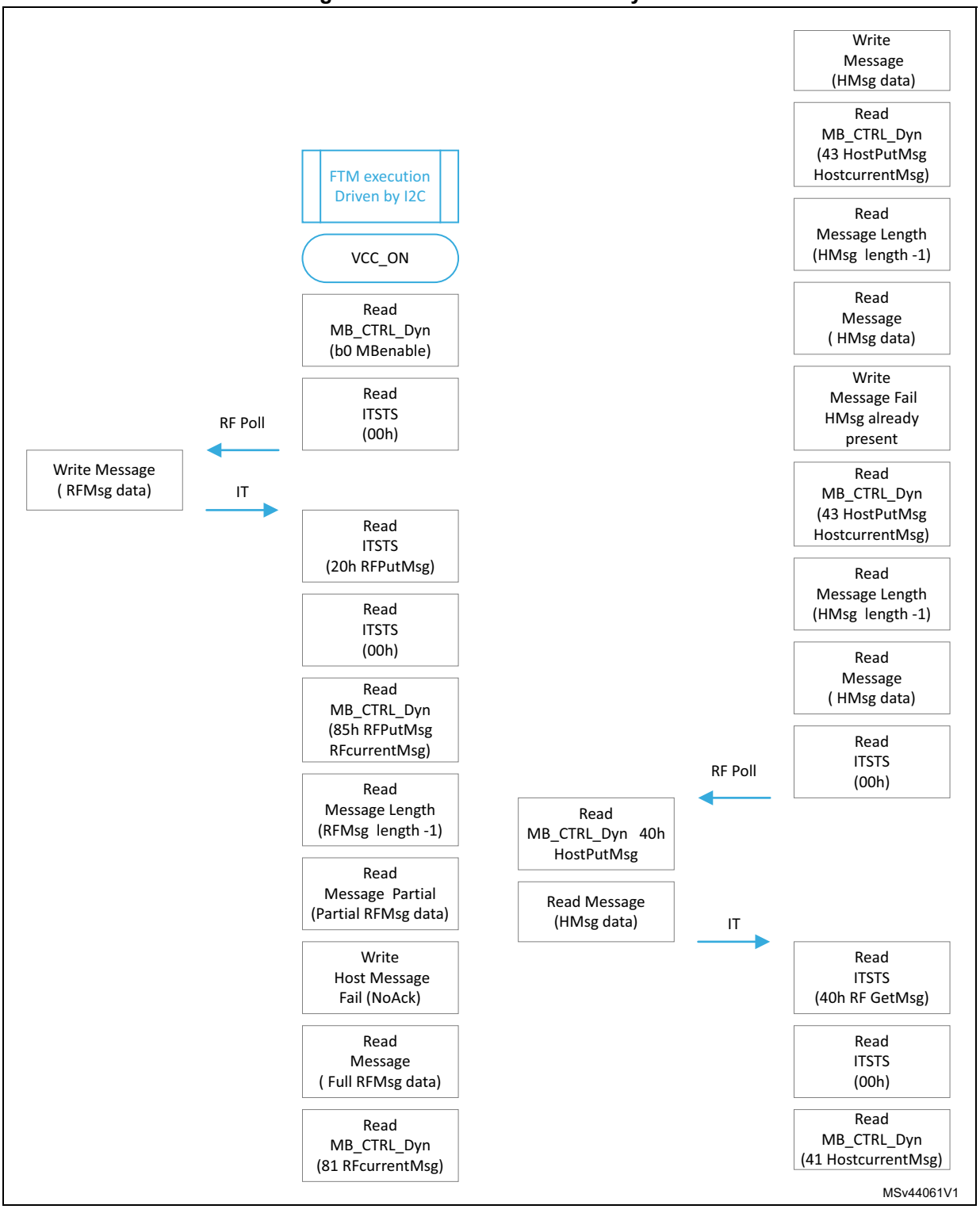


Table 14. FTM execution driven by I²C (1/5)

Command flow	Request/Response frame	Polling (optional)	Comment
Vcc_ON	-	-	DC Power ON
I²C Read Dynamic Register **MB_CTRL_Dyn**	**Start sA6** rAck s20 rAck s06 rAck **Start sA7** rAck r01 sNoack **Stop**	-	Enable FTM
I²C Read Dynamic Register **IT_STS_Dyn**	**Start sA6** rAck s20 rAck s05 rAck **Start sA7** rAck r00 sNoack **Stop**	-	No RF IT yet

Break: RF puts message

Table 15. Execution driven by I²C (2/5)

Command flow	Request frame	Response	Comment
RF Power ON	-	-	-
RF Write Message	02 **AA** 02 0D 03 B0 B1 B2 B3	00h	RF put message1

Table 16. FTM execution driven by I²C (3/5)

Command flow	Request/Response frame	Polling (optional)	Comment
Vcc_ON	-	-	DC Power ON
I²C Read Dynamic Register **IT_STS_Dyn**	**Start sA6** rAck s20 rAck s05 rAck **Start sA7** rAck r20 sNoack **Stop**	-	20h = RF_PUT_MSG
I²C Read Dynamic Register **IT_STS_Dyn**	**Start sA6** rAck s20 rAck s05 rAck **Start sA7** rAck r00 sNoack **Stop**	-	00h = IT_STS_Dyn reset
I²C Read Dynamic Register **MB_CTRL_Dyn**	**Start sA6** rAck s20 rAck s06 rAck **Start sA7** rAck r85 sNoack **Stop**	-	85h = RF_PUT_MSG RF_CURRENT_MSG Host could not change Msg
I²C Read Dynamic Register **MB_LEN_Dyn**	**Start sA6** rAck s20 rAck s07 rAck **Start sA7** rAck r03 sNoack **Stop**	-	MB_LEN 03 4 bytes
I²C Read Message (2 bytes as example)	**Start sA6** rAck s20 rAck s08 rAck **Start sA7** rAck rB0 sAck rB1 sNoack **Stop**	-	Partial Read RF message 1
I²C Read Dynamic Register **MB_CTRL_Dyn**	**Start sA6** rAck s20 rAck s06 rAck **Start sA7** rAck r85 sNoack **Stop**	-	85h = RF_PUT_MSG RF_CURRENT_MSG Host could not change Msg

Table 16. FTM execution driven by I²C (3/5) (continued)

Command flow	Request/Response frame	Polling (optional)	Comment
I²C Write Message	**Start sA6** rAck s20 rAck s08 rAck sB0 rNoAck **Stop**	-	Host tries to write message, No Ack Write is forbidden RF message not read
I²C Read Message	**Start sA6** rAck s20 rAck s08 rAck **Start sA7** rAck rB0 sAck rB1 sAck rB2 sAck rB3 sNoack **Stop**	-	Full Read RF message 1
I²C Read Dynamic Register **MB_CTRL_Dyn**	**Start sA6** rAck s20 rAck s06 rAck **Start sA7** rAck r85 sNoack **Stop**	-	81h = RF_CURRENT_MSG Msg could not be updated by RF or Host
I²C Write Message (16 bytes as example)	**Start sA6** rAck s20 rAck s08 rAck sB0 rAck sB1 rAck sB2 rAck sB3 rAck sB4 rAck sB5 rAck sB6 rAck sB7 rAck sB8 rAck sB9 rAck sBA rAck sBB rAck sBC rAck sBD rAck sBE rAck sBF rAck sC0 rAck sC1 rAck sC2 rAck sC3 rAck **Stop**	**Start sA6** rAck	Write Host message1 Ack after first polling write in register is immediate
I²C Read Dynamic Register **MB_CTRL_Dyn**	**Start sA6** rAck s20 rAck s06 rAck **Start sA7** rAck r43 sNoack **Stop**	-	43h: HOST_PUT_MSG HOST_CURRENT_MSG
I²C Read Dynamic Register **MB_LEN_Dyn**	**Start sA6** rAck s20 rAck s07 rAck **Start sA7** rAck r13 sNoack **Stop**	-	MB_LEN 13h 20 bytes
I²C Read Message (16 bytes as example)	**Start sA6** rAck s20 rAck s08 rAck Start sA7 rAck rB0 sAck rB1 sAck rB2 sAck rB3 sAck rB4 sAck rB5 sAck rB6 sAck rB7 sAck rB8 sAck rB9 sAck rBA sAck rBB sAck rBC sAck rBD sAck rBE sAck rBF sAck rC0 sAck rC1 sAck rC2 sAck rC3 sNoack **Stop**	-	Host Check message1
I²C Write Message (16 bytes as example)	**Start sA6** rAck s20 rAck s08 rAck sC0 rNoAck sC1 rNoAck sC2 rNoAck **Stop**	-	Host tries to overwrite message1. This is forbidden: NoAck of data
I²C Read Dynamic Register **MB_CTRL_Dyn**	**Start sA6** rAck s20 rAck s06 rAck **Start sA7** rAck r43 sNoack **Stop**	-	No Change (43h: HOST_PUT_MSG HOST_CURRENT_MSG)
I²C Read Dynamic Register **MB_LEN_Dyn**	**Start sA6** rAck s20 rAck s07 rAck **Start sA7** rAck r13 sNoack **Stop**	-	No Change (MB_LEN 13h 20 bytes)

Table 16. FTM execution driven by I²C (3/5) (continued)

Command flow	Request/Response frame	Polling (optional)	Comment
I²C Read Message (16 bytes as example)	**Start sA6** rAck s20 rAck s08 rAck Start sA7 rAck rB0 sAck rB1 sAck rB2 sAck rB3 sAck rB4 sAck rB5 sAck rB6 sAck rB7 sAck rB8 sAck rB9 sAck rBA sAck rBB sAck rBC sAck rBD sAck rBE sAck rBF sAck rC0 sAck rC1 sAck rC2 sAck rC3 sNoack **Stop**	-	No Change (Host Check message1)
I²C Read Dynamic Register **IT_STS_Dyn**	**Start sA6** rAck s20 rAck s05 rAck **Start sA7** rAck r00 sNoack **Stop**	-	00h: IT_STS_Dyn reset

Break: RF polls MB_CTRL_Dyn, detects, then Host puts message and reads it

Table 17. Execution driven by I²C (4/5)

Command flow	Request frame	Response	Comment
RF Power ON	-	-	-
RF Read Dynamic Register **MB_CTRL_Dyn**	02 **AD** 02 0D	00 43h	HOST_PUT_MSG HOST_CURRENT_MSG MB_EN
RF Read Message Length	02 **AB** 02	00 13h	MB_LEN 13 20 data bytes
RF Read Message	02 **AC** 02 00 00	00 B0B1B2B3B4B5B6B7B8B9 BABBBCBD BEBEC0C1 C2C3	RF Get Host message1 (MB_LEN 00h, Full message received)
RF Read Dynamic Register **MB_CTRL_Dyn**	02 **AD** 02 0D	<- 00 41h	Mailbox is ready for a new sequence (HOST_CURRENT_MSG; MB_EN)

Table 18. FTM execution driven by I²C (5/5)

Command flow	Request/Response frame	Polling (optional)	Comment
Vcc_ON	-	-	DC Power ON
I²C Read Dynamic Register **IT_STS_Dyn**	**Start sA6** rAck s20 rAck s05 rAck **Start sA7** rAck r40 sNoack **Stop**	-	40h: RF_GET_MSG Host is informed that the remote device gets the message in the mailbox
I²C Read Dynamic Register **IT_STS_Dyn**	**Start sA6** rAck s20 rAck s05 rAck **Start sA7** rAck r00 sNoack **Stop**	-	00h: IT_STS_Dyn reset
I²C Read Dynamic Register **MB_CTRL_Dyn**	**Start sA6** rAck s20 rAck s06 rAck **Start sA7** rAck r41 sNoack **Stop**	-	Mailbox is ready for a new sequence (HOST_CURRENT_MSG; MB_EN)

6 How to disable fast transfer mode

There are three ways to stop FTM:

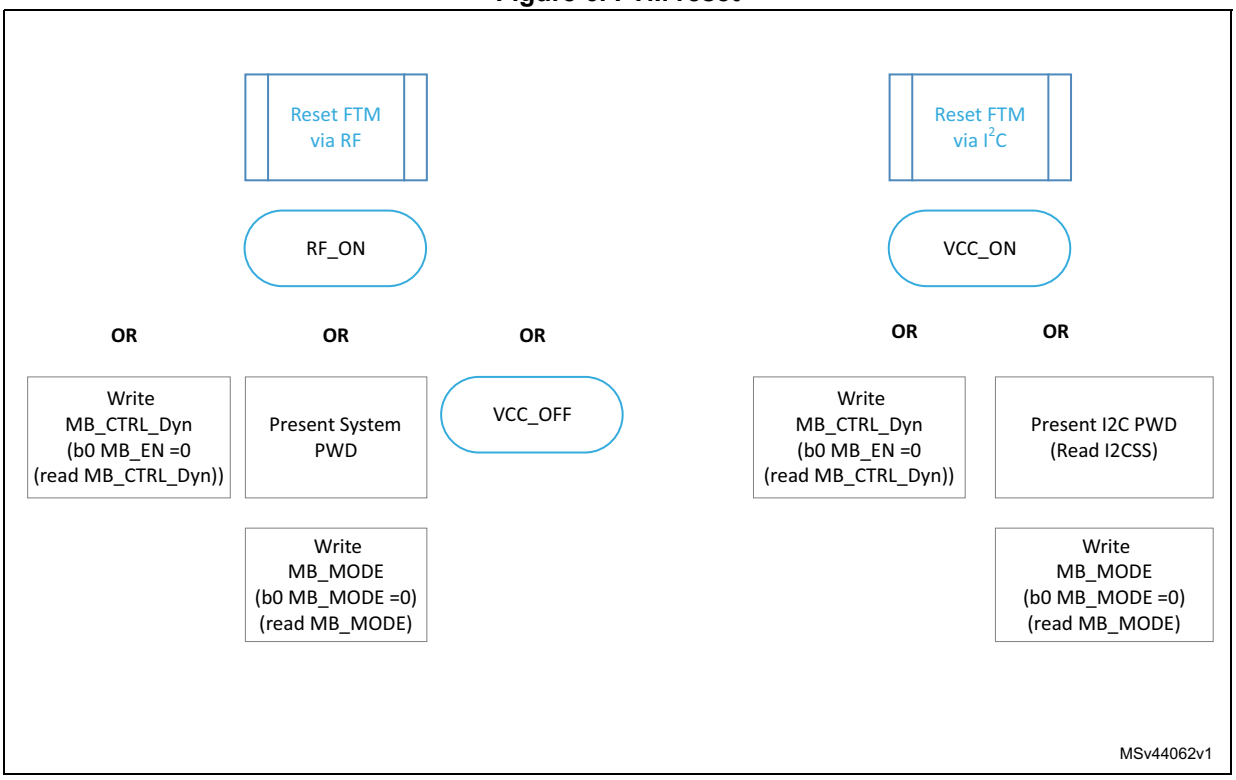
1. The faster way is to drive the MB_EN dynamic bit to 0b. The mailbox buffer will be accessible for writing in user memory. Return to FTM will be easy by driving MB_EN back to 1b.
2. The second way is to set the MB_MODE system bit to 0b, which requires system access rights but will protect from an unauthorized 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 6 summarizes the different ways to reset FTM which are further detailed in *Section 6.1* and *Section 6.2*.

Figure 6. FTM reset



6.1 RF sequence to reset FTM

1. Reset MB_EN in dynamic register MB_CTRL_Dyn

Table 19. 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

2. Reset MB_MODE in System

Table 20. Reset MB_MODE in dynamic register MB_CTRL_Dyn (RF sequence)

Command flow	Request frame	Response	Comment
RF Present System Password (0)	02 **B3** 02 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 **MB_MODE**	02 **A1** 02 00 00h	00h	set to 00h on delivery 00h: MB_MODE Disable
RF Read Static Register **MB_MODE**	02 **A0** 02 0Dh	00 00h	Optional

3. VCC Power OFF

6.2 I²C sequence to reset FTM

1. Reset MB_EN in dynamic register MB_CTRL_Dyn

Table 21. Reset MB_EN in dynamic register MB_CTRL_Dyn (I²C sequence)

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 s00 rAck **Stop**	**Start sA7** rNoack **Start sA7** rNoack … **Start sA7** rAck	Disable FTM
I²C Read Dynamic Register **MB_CTRL_Dyn**	**Start sA6** rAck s20 rAck s06 rAck **Start sA7** rAck r00 sNoack **Stop**	-	Disable FTM

2. Reset MB_MODE in System

Table 22. Reset MB_MODE in System (I²C sequence)

Command flow	Request/Response frame	Polling (optional)	Comment
I²C 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
I²C 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)
I²C Write Static Register **MB_MODE**	**Start sAE** rAck s00 rAck s0D rAck s00 rAck **Stop**	Start sAE rNoack Start sAE rNoack … Start sAE rAck	Reset to 00h
I²C Read Static Register **MB_MODE**	**Start sAE** rAck s00 rAck s0D rAck **Start sAF** rAck r00 sNoack **Stop**	-	Optional

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.

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 Appendix

This appendix describes FTM related registers.

9.1 Static registers relative to FTM

This section provides a description of the following registers:

- GPO (see *Table 23*)
- IT_TIME (see *Table 24*)
- MB_MODE (see *Table 25*)
- MB_WDG (see *Table 26*)

Table 23. GPO static register

GPO			
RF	**Command**	**Read Configuration (cmd code A0h) @00h** **Write Configuration (cmd code A1h) @00h**	
	Type	**R always, W if RF configuration security session is open and configuration not locked**	
I^2C	**Address**	**E2=1, 0000h**	
	Type	**R always, W if I^2C security session is open**	
Bit	**Name**	**Function**	**Factory value**
b0	RF_USER_EN	0: disabled 1: GPO output level is controlled by Manage GPO Command (set/reset)	0b
b1	RF_ACTIVITY_EN	0: disabled 1: GPO output level changes from RF command EOF to response EOF	0b
b2	RF_INTERRUPT_EN	0: disabled 1: GPO output level is controlled by Manage GPO Command (pulse)	0b
b3	FIELD_CHANGE_EN	0: disabled 1: A pulse is emitted on GPO, when RF field appears or disappears	1b
b4	RF_PUT_MSG_EN	0: disabled 1: A pulse is emitted on GPO at completion of valid RF Write Message command	0b
b5	RF_GET_MSG_EN	0: disabled 1: A pulse is emitted on GPO at completion of valid RF Read Message command if end of message has been reached	0b

Table 23. GPO static register (continued)

GPO			
RF	**Command**	**Read Configuration (cmd code A0h) @00h** **Write Configuration (cmd code A1h) @00h**	
	Type	**R always, W if RF configuration security session is open and configuration not locked**	
I^2C	**Address**	**E2=1, 0000h**	
	Type	**R always, W if I^2C security session is open**	
Bit	**Name**	**Function**	**Factory value**
b6	RF_WRITE_EN	0: disabled 1: A pulse is emitted on GPO at completion of valid RF write operation in EEPROM	0b
b7	GPO_EN	0: GPO output is disabled. GPO is High-Z (Open drain) or 0 (CMOS) 1: GPO output is enabled. GPO outputs enabled interrupts	1b

Table 24. IT_TIME static register

IT_TIME			
RF	**Command**	**Read Configuration (cmd code A0h) @01h** **Write Configuration (cmd code A1h) @01h**	
	Type	**R always, W if RF configuration security session is open and configuration not locked**	
I^2C	**Address**	**E2=1, 0001h**	
	Type	**R always, W if I^2C security session is open**	
Field	**Name**	**Function**	**Factory value**
b2-b0	IT_TIME	Pulse duration = 301 µs - IT_TIME x 37.65 µs ± 2 µs	011b
b7-b3	RFU	-	00000b

Table 25. MB_MODE static register

MB_MODE			
RF	Command	Read Configuration (cmd code A0h) @0Dh Write Configuration (cmd code A1h) @0Dh	
	Type	R always, W if RF configuration security session is open and configuration not locked	
I^2C	Address	E2=1, 000Dh	
	Type	R always, W if I^2C security session is open	
Field	**Name**	**Function**	**Factory value**
b0	MB_MODE	0: Enabling fast transfer mode is forbidden 1: Enabling fast transfer mode is authorized	0b
b7-b1	RFU	-	0000000b

Table 26. MB_WDG static register

MB_WDG			
RF	Command	Read Configuration (cmd code A0h) @0Eh Write Configuration (cmd code A1h) @0Eh	
	Type	R always, W if RF configuration security session is open and configuration not locked	
I^2C	Address	E2=1, 000Eh	
	Type	R always, W if I^2C security session is open	
Field	**Name**	**Function**	**Factory value**
b2-b0	MB_WDG	Watch dog duration = $2^{(MB\_WDG - 1)}$ x 30 ms ± 6 ms If MB_WDG = 0, then watchdog duration is infinite	111b
b7-b3	RFU	-	00000b

9.2 Dynamic registers relative to FTM

This section provides a description of the following registers:

- GPO_CTRL_Dyn (see *Table 27*)
- IT_STS_Dyn (see *Table 28*)
- I2C_SSO_Dyn (see *Table 29*)
- EH_CTRL_Dyn (see *Table 30*)
- MB_CTRL_Dyn (see *Table 31*)
- MB_LEN_Dyn (see *Table 32*)
- Fast transfer mode buffer (see *Table 33*)

Table 27. GPO_CTRL_Dyn dynamic register

GPO_CTRL_Dyn			
RF	**Command**	**Read Dynamic Configuration (cmd code ADh) @00h** **Write Dynamic Configuration (cmd code AEh) @00h** **Fast Read Dynamic Configuration (cmd code CDh) @00h** **Fast Write Dynamic Configuration (cmd code CEh) @00h**	
	Type	**RO**	
I²C	**Address**	**E2 = 0, 2000h**	
	Type	**b0-b6: RO - b7: R always, W always**	
Bit	**Name**	**Function**	**Factory value**
b0	RF_USER_EN	0: disabled 1: GPO output level is controlled by Manage GPO Command (set/reset)	0b
b1	RF_ACTIVITY_EN	0: disabled 1: GPO output level changes from RF command SOF to response EOF	0b
b2	RF_INTERRUPT_EN	0: disabled 1: GPO output level is controlled by Manage GPO Command (pulse)	0b
b3	FIELD_CHANGE_EN	0: disabled 1: a pulse is emitted on GPO, when RF field appears or disappears	1b
b4	RF_PUT_MSG_EN	0: disabled 1: a pulse is emitted on GPO at completion of valid RF Write Message command	0b
b5	RF_GET_MSG_EN	0: disabled 1: a pulse is emitted on GPO at completion of valid RF Read Message command if end of message has been reached	0b
b6	RF_WRITE_EN	0: disabled 1: a pulse is emitted on GPO at completion of valid RF write operation in EEPROM	0b
b7	GPO_EN	0: GPO output is disabled. GPO is High-Z (Open Drain) or 0 (CMOS) 1: GPO output is enabled. GPO outputs enabled interrupts	1b

Table 28. IT_STS_Dyn dynamic register

IT_STS_Dyn			
RF	**Command**	**No access**	
	Type		
I²C	**Address**	**E2 = 0, 2005h**	
	Type	**RO**	
Bit	**Name**	**Function**	**Factory value**
b0	RF_USER	0: Manage GPO reset GPO 1: Manage GPO set GPO	0b
b1	RF_ACTIVITY	0: no RF access 1: RF access	0b
b2	RF_INTERRUPT	0: No Manage GPO interrupt request 1: Manage GPO interrupt request	0b
b3	FIELD_FALLING	0: no RF field falling 1: RF Field falling	0b
b4	FIELD_RISING	0: no RF field rising 1: RF field rising	0b
b5	RF_PUT_MSG	0: no message put by RF in FTM mailbox 1: message put by RF in FTM mailbox	0b
b6	RF_GET_MSG	0: no message read by RF from FTM mailbox 1: message read by RF from FTM mailbox and end of message has been reached	0b
b7	RF_WRITE	0: no write in EEPROM 1: write in EEPROM	0b

Table 29. I2C_SSO_Dyn dynamic register

I2C_SSO_Dyn			
RF	**Command**	**No access**	
	Type		
I²C	**Address**	**E2 = 0, 2004h**	
	Type	**RO**	
Field	**Name**	**Function**	**Factory value**
b0	I2C_SSO	0: I²C security session close 1: I²C security session open (Set or reset via I²C Present password command)	0b
b7-b1	RFU	-	0000000b

Table 30. EH_CTRL_Dyn dynamic register

EH_CTRL_Dyn			
RF	**Command**	**Read Dynamic Configuration (cmd code ADh) @02h** **Fast Read Dynamic Configuration (cmd code CDh) @02h** **Write Dynamic Configuration (cmd code AEh) @02h** **Fast Write Dynamic Configuration (cmd code CEh) @02h**	
	Type	**b0: R always, W – b1 - b7: RO**	
I²C	**Address**	**E2 = 0, 2002h**	
	Type	**b0: R always, W always** **b1-b7: RO**	
Field	**Name**	**Function**	**Factory value**
b0	EH_EN	0: disable EH feature 1: enable EH feature	0b
b1	EH_ON	0: EH feature is disabled 1: EH feature is enabled	0b
b2	FIELD_ON	0: RF field is not detected 1: RF field is present and ST25DV-I2C may communicate in RF	Depending o power source
b3	VCC_ON	0: No DC supply detected on VCC pin or Low Power Down mode is forced (LPD is high) 1: VCC supply is present and Low Power Down mode is not forced (LPD is low)	Depending o power source
b7-b4	RFU	-	0000b

Table 31. MB_CTRL_Dyn dynamic register

MB_CTRL_Dyn			
RF	**Command**	**Read Dynamic Configuration (cmd code ADh) @0Dh** **Fast Read Dynamic Configuration (cmd code CDh) @0Dh** **Write Dynamic Configuration (cmd code AEh) @0Dh** **Fast Write Dynamic Configuration (cmd code CEh) @0Dh**	
	Type	**b0: R always, W – b7-b1: RO**	
I^2C	**Address**	**E2 = 0, 2006h**	
	Type	**b0: R always, W - b7 - b1: RO**	
Bit	**Name**	**Function**	**Factory value**
b0	MB_EN (1)	0: disable FTM, FTM mailbox is empty 1: enable FTM	0b
b1	HOST_PUT_MSG	0: no I^2C message in FTM mailbox 1: I^2C has put a message in FTM mailbox	0b
b2	RF_PUT_MSG	0: no RF message in FTM mailbox 1: RF has put a message in FTM mailbox	0b
b3	RFU	-	0b
b4	HOST_MISS_MSG	0: no message missed by I^2C 1: I^2C did not read RF message before watchdog time out	0b
b5	RF_MISS_MSG	0: no message missed by RF 1: RF did not read message before watchdog time out	0b
b6	HOST_CURRENT_MSG	0: no message or message not coming from I^2C 1: current Message in FTM mailbox comes from I^2C	0b
b7	RF_CURRENT_MSG	0: no message or message not coming from RF 1: current Message in FTM mailbox comes from RF	0b

1. MB_EN bit is automatically reset to 0 if MB_MODE register is reset to 0.

Table 32. MB_LEN_Dyn dynamic register

MB_LEN_Dyn			
RF	Command	Read Message Length (cmd code ABh) Fast Read Message Length (cmd code CBh)	
	Type	RO	
I^2C	Address	E2 = 0, 2007h	
	Type	RO	
Field	Name	Function	Factory value
b7-b0	MB_LEN	Size in byte of message contained in FTM mailbox (automatically set by ST25DV-I2C)	0h

Table 33. Fast transfer mode mailbox memory map

RF access		Fast transfer mode buffer		I^2C access		
Address	Type	Name	Function	Device select	Address	Type
00h	R/W	MB_Dyn byte 0	Fast transfer mode buffer (256 bytes)	E2 = 0	2008h	R/W
01h	R/W	MB_Dyn byte 1		E2 = 0	2009h	R/W
...	...	...		E2 = 0	...	...
FEh	R/W	MB_Dyn byte 254		E2 = 0	2106h	R/W
FFh	R/W	MB_Dyn byte 255		E2 = 0	2107h	R/W

10 Revision history

Table 34. Document revision history

Date	Revision	Changes
01-Mar-2017	1	Initial release.
02-Oct-2019	2	Updated *Table 3: I^2C sequence for FTM preparation*. Replaced generic ST25DVxxx and ST25DV with ST25DV-I2C.