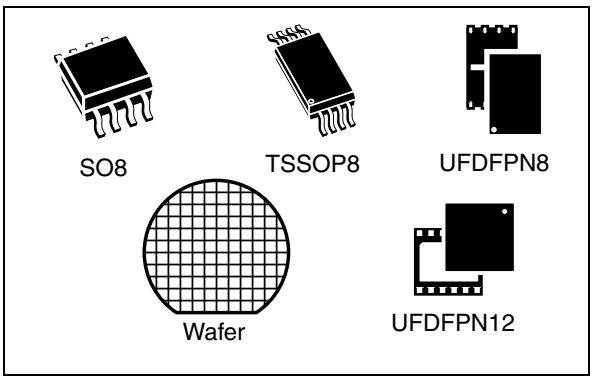
ST25DVxxx

Dynamic NFC/RFID tag IC with 4-Kbit, 16-Kbit or 64-Kbit EEPROM, and Fast Transfer Mode capability

Datasheet - production data



Features

I²C interface

- Two-wire I²C serial interface supports 1MHz protocol
- Single supply voltage: 1.8V to 5.5V
- Multiple byte write programing (up to 256 bytes)

Contactless interface

- Based on ISO/IEC 15693 and NFC Forum Type 5 tag
- Supports all ISO/IEC 15693 modulations, coding, subcarrier modes and data rates
- Custom Fast read access up to 53 Kbit/s
- Single and multiple blocks read(1)
- Single and multiple blocks write(1) (up to 4)
- Internal tuning capacitance: 28.5 pF

Memory

- Up to 64-kbits of EEPROM (depending on version)
- I²C interface accesses bytes
- RF interface accesses blocks of 4 bytes
- Write time:
  - From I²C: typical 5ms for 1 byte
  - From RF: typical 5ms for 1 block
- Data retention: 40 years
- Write cycles endurance:
  - 1 million write cycles at 25 °C
  - 600k write cycles at 85 °C

Fast Transfer Mode

- Fast data transfer between I²C and RF interfaces
- Half-duplex 256-byte dedicated buffer

Energy harvesting

- Analog output pin to power external components

Data protection

- User memory: 1 to 4 configurable areas, protectable in read and/or write by three 64-bit passwords in RF and one 64-bit password in I²C
- System configuration: protected in write by a 64-bit password in RF and a 64-bit password in I²C

GPO

- Interruption pin configurable on multiple RF events (field change, memory write, activity, Fast Transfer end, user set/reset/pulse)
- Open Drain or CMOS output (depending on version)

Low power mode(2)

- Input pin to trigger low power mode

RF management

- RF command interpreter enabled/disabled from I²C host controller

Temperature range

- From - 40 to 85 °C

Package

- 8-pin and 12-pin packages
- ECOPACK2® (RoHS compliant)

Table 1. Device summary

Reference	Part number
ST25DVxxx	ST25DV04K ST25DV16K ST25DV64K

1. Same for Extended commands.

2. 12-pin package only.

This is information on a product in full production. www.st.com

Contents

1 **Description 17**

1.1 ST25DVxxx block diagram 17

1.2 ST25DVxxx packaging 18

2 **Signal descriptions 20**

2.1 Serial link (SCL, SDA) 20

2.1.1 Serial clock (SCL) 20

2.1.2 Serial data (SDA) 20

2.2 Power control (V_{CC}, LPD,V_{SS}) 20

2.2.1 Supply voltage (V_{CC}) 20

2.2.2 Low Power Down (LPD) 20

2.2.3 Ground (V_{SS}) 20

2.3 RF link (AC0 AC1) 21

2.3.1 Antenna coil (AC0, AC1) 21

2.4 Process control (V_{DCG}, GPO) 21

2.4.1 Driver Supply voltage (V_{DCG}) 21

2.4.2 General purpose output (GPO) 21

2.5 Energy harvesting analog output (V_EH) 21

3 **Power management 22**

3.1 Wired interface 22

3.2 Contactless interface 23

4 **Memory management 24**

4.1 Memory organization overview 24

4.2 User memory 25

4.2.1 User memory areas 27

4.3 System configuration area 31

4.4 Dynamic configuration 34

4.5 Fast Transfer Mode mailbox 35

5 **ST25DVxxx specific features 36**

5.1 Fast transfer mode (FTM) 37

5.1.1 Fast Transfer Mode registers . . . 37
5.1.2 Fast Transfer Mode usage . . . 39
5.2 GPO . . . 44
5.2.1 ST25DVxxx interrupt capabilities on RF events . . . 44
5.2.2 GPO and power supply . . . 52
5.2.3 GPO registers . . . 53
5.2.4 Configuring GPO . . . 57
5.3 Energy Harvesting (EH) . . . 59
5.3.1 Energy harvesting registers . . . 59
5.3.2 Energy harvesting feature description . . . 60
5.3.3 EH delivery state diagram . . . 61
5.3.4 EH delivery sequence . . . 62
5.4 RF management feature . . . 63
5.4.1 RF management registers . . . 63
5.4.2 RF management feature description . . . 63
5.5 Interface Arbitration . . . 65
5.6 Data Protection . . . 66
5.6.1 Data protection registers . . . 66
5.6.2 Passwords and security sessions . . . 75
5.6.3 User memory protection . . . 78
5.6.4 System memory protection . . . 80
5.7 Device Parameter Registers . . . 80

6 I^2C operation . . . 85
6.1 I2C protocol . . . 85
6.1.1 Start condition . . . 85
6.1.2 Stop condition . . . 86
6.1.3 Acknowledge bit (ACK) . . . 86
6.1.4 Data input . . . 86
6.2 I^2C timeout . . . 86
6.2.1 I^2C timeout on Start condition . . . 86
6.2.2 I^2C timeout on clock period . . . 87
6.3 Device addressing . . . 87
6.4 I^2C Write operations . . . 88
6.4.1 I^2C Byte write . . . 89
6.4.2 I^2C Sequential write . . . 89

6.4.3 Minimizing system delays by polling on ACK . . . 90
6.5 I²C read operations . . . 92
6.5.1 Random Address Read . . . 92
6.5.2 Current Address Read . . . 92
6.5.3 Sequential Read access . . . 93
6.5.4 Acknowledge in Read mode . . . 94
6.6 I²C password management . . . 95
6.6.1 I²C present password command description . . . 95
6.6.2 I²C write password command description . . . 96

7 RF operations . . . 97
7.1 RF communication . . . 97
7.1.1 Access to a ISO/IEC 15693 device . . . 97
7.2 RF communication and energy harvesting . . . 98
7.3 Fast Transfer Mode mailbox access in RF . . . 98
7.4 RF protocol description . . . 98
7.4.1 Protocol description . . . 98
7.4.2 ST25DVxxx states referring to RF protocol . . . 99
7.4.3 Modes . . . 101
7.4.4 Request format . . . 101
7.4.5 Request flags . . . 101
7.4.6 Response format . . . 103
7.4.7 Response flags . . . 103
7.4.8 Response and error code . . . 104
7.5 Timing definition . . . 104
7.6 RF Commands . . . 106
7.6.1 RF command code list . . . 106
7.6.2 Command codes list . . . 108
7.6.3 General Command Rules . . . 109
7.6.4 Inventory . . . 109
7.6.5 Stay Quiet . . . 110
7.6.6 Read Single Block . . . 111
7.6.7 Extended Read Single Block . . . 112
7.6.8 Write Single Block . . . 113
7.6.9 Extended Write Single Block . . . 115
7.6.10 Lock block . . . 116

7.6.11 Extended Lock block 118
7.6.12 Read Multiple Blocks 119
7.6.13 Extended Read Multiple Blocks 121
7.6.14 Write Multiple Blocks 122
7.6.15 Extended Write Multiple Blocks 124
7.6.16 Select 125
7.6.17 Reset to Ready 126
7.6.18 Write AFI 128
7.6.19 Lock AFI 129
7.6.20 Write DSFID 130
7.6.21 Lock DSFID 131
7.6.22 Get System Info 132
7.6.23 Extended Get System Info 134
7.6.24 Get Multiple Block Security Status 138
7.6.25 Extended Get Multiple Block Security Status 139
7.6.26 Read Configuration 141
7.6.27 Write Configuration 142
7.6.28 Read Dynamic Configuration 144
7.6.29 Write Dynamic Configuration 145
7.6.30 Manage GPO 146
7.6.31 Write Message 148
7.6.32 Read Message Length 149
7.6.33 Read Message 150
7.6.34 Fast Read Message 151
7.6.35 Write Password 152
7.6.36 Present Password 154
7.6.37 Fast Read Single Block 155
7.6.38 Fast Extended Read Single Block 156
7.6.39 Fast Read Multiple Blocks 158
7.6.40 Fast Extended Read Multiple Block 160
7.6.41 Fast Write Message 162
7.6.42 Fast Read Message Length 163
7.6.43 Fast Read Dynamic Configuration 164
7.6.44 Fast Write Dynamic Configuration 165

8 Unique identifier (UID) 167

9 Device parameters 168

9.1 Maximum rating 168

9.2 I²C DC and AC parameters 169

9.3 GPO Characteristics 174

9.4 RF electrical parameters 175

10 Package information 178

10.1 SO8N package information 178

10.2 TSSOP8 package information 179

10.3 UFDFN8 package information 181

10.4 UFDFPN12 package information 183

11 Part numbering 184

Appendix A Bit representation and coding for fast commands 185

A.1 Bit coding using one subcarrier 185

A.1.1 High data rate 185

A.1.2 Low data rate 185

A.2 ST25DVxxx to VCD frames 186

A.3 SOF when using one subcarrier 186

A.3.1 High data rate 186

A.3.2 Low data rate 186

A.4 EOF when using one subcarrier 187

A.4.1 High data rate 187

A.4.2 Low data rate 187

Appendix B I2C sequences 188

B.1 Device select codes 188

B.2 I²C Byte writing and polling 188

B.2.1 I²C byte write in user memory 188

B.2.2 I²C byte writing in dynamic registers and polling 190

B.2.3 I²C byte write in mailbox and polling 191

B.2.4 I²C byte write and polling in system memory 192

B.3 I²C sequential writing and polling 194

B.3.1 I²C sequential write in user memory and polling 194

B.3.2 I²C sequential write in mailbox and polling 196
B.4 I²C Read current address 197
B.4.1 I²C current address read in User memory 197
B.5 I²C random address read 198
B.5.1 I²C random address read in system memory 198
B.5.2 I²C Random address read in system memory 199
B.5.3 I²C Random address read in dynamic registers 199
B.6 I²C sequential read 200
B.6.1 I²C sequential read in user memory 200
B.6.2 I²C sequential read in system memory 202
B.6.3 I²C sequential read in dynamic registers 203
B.6.4 I²C sequential read in mailbox 205
B.7 I²C password relative sequences 207
B.7.1 I²C write password 207
B.7.2 I²C present password 208

Revision history 210

List of tables

Table 1. Device summary . . . 1
Table 2. Signal names . . . 18
Table 3. User memory as seen by RF and by I2C . . . 26
Table 4. Maximum user memory Block and Byte addresses and ENDAi value . . . 28
Table 5. Areas and limit calculation from ENDAi registers . . . 28
Table 6. ENDA1 . . . 30
Table 7. ENDA2 . . . 31
Table 8. ENDA3 . . . 31
Table 9. System configuration memory map . . . 32
Table 10. Dynamic registers memory map . . . 34
Table 11. Fast Transfer Mode mailbox memory map . . . 35
Table 12. MB_MODE . . . 37
Table 13. MB_WDG . . . 38
Table 14. MB_CTRL_Dyn . . . 38
Table 15. MB_LEN_Dyn . . . 39
Table 16. FIELD_CHANGE when RF is disabled or in sleep mode . . . 48
Table 17. GPO interrupt capabilities in function of RF field . . . 53
Table 18. GPO interrupt capabilities in function of VCC power supply . . . 53
Table 19. GPO . . . 54
Table 20. IT_TIME . . . 55
Table 21. GPO_CTRL_Dyn . . . 55
Table 22. IT_STS_Dyn . . . 57
Table 23. Enabling or disabling GPO interruptions . . . 58
Table 24. EH_MODE . . . 59
Table 25. EH_CTRL_Dyn . . . 59
Table 26. Energy harvesting at power-up . . . 60
Table 27. RF_MNGT . . . 63
Table 28. RF_MNGT_Dyn . . . 63
Table 29. RFA1SS . . . 66
Table 30. RFA2SS . . . 67
Table 31. RFA3SS . . . 68
Table 32. RFA4SS . . . 69
Table 33. I2CSS . . . 70
Table 34. LOCK_CCFILE . . . 71
Table 35. LOCK_CFG . . . 72
Table 36. I2C_PWD . . . 72
Table 37. RF_PWD_0 . . . 73
Table 38. RF_PWD_1 . . . 73
Table 39. RF_PWD_2 . . . 74
Table 40. RF_PWD_3 . . . 74
Table 41. I2C_SSO_Dyn . . . 75
Table 42. Security session type . . . 75
Table 43. LOCK_DSFID . . . 80
Table 44. LOCK_AFI . . . 81
Table 45. DSFID . . . 81
Table 46. AFI . . . 82
Table 47. MEM_SIZE . . . 82
Table 48. BLK_SIZE . . . 83

Table 49. IC_REF . . . 83
Table 50. UID . . . 84
Table 51. IC_REV . . . 84
Table 52. Device select code . . . 87
Table 53. Operating modes . . . 88
Table 54. Address most significant byte . . . 88
Table 55. Address least significant byte . . . 88
Table 56. ST25DVxxx response depending on Request_flags . . . 100
Table 57. General request format . . . 101
Table 58. Definition of request flags 1 to 4 . . . 102
Table 59. Request flags 5 to 8 when inventory_flag, Bit 3 = 0 . . . 102
Table 60. Request flags 5 to 8 when inventory_flag, Bit 3 = 1 . . . 103
Table 61. General response format . . . 103
Table 62. Definitions of response flags 1 to 8 . . . 103
Table 63. Response error code definition . . . 104
Table 64. Timing values . . . 105
Table 65. Command codes . . . 108
Table 66. Inventory request format . . . 109
Table 67. Inventory response format . . . 109
Table 68. Stay Quiet request format . . . 110
Table 69. Read Single Block request format . . . 111
Table 70. Read Single Block response format when Error_flag is NOT set . . . 111
Table 71. Block security status . . . 111
Table 72. Read Single Block response format when Error_flag is set . . . 112
Table 73. Extended Read Single Block request format . . . 112
Table 74. Extended Read Single Block response format when Error_flag is NOT set . . . 113
Table 75. Block security status . . . 113
Table 76. Extended Read Single Block response format when Error_flag is set . . . 113
Table 77. Write Single Block request format . . . 114
Table 78. Write Single Block response format when Error_flag is NOT set . . . 114
Table 79. Write Single Block response format when Error_flag is set . . . 114
Table 80. Extended Write Single request format . . . 115
Table 81. Extended Write Single response format when Error_flag is NOT set . . . 115
Table 82. Extended Write Single response format when Error_flag is set . . . 116
Table 83. Lock block request format . . . 116
Table 84. Lock block response format when Error_flag is NOT set . . . 117
Table 85. Lock single block response format when Error_flag is set . . . 117
Table 86. Extended Lock block request format . . . 118
Table 87. Extended Lock block response format when Error_flag is NOT set . . . 118
Table 88. Extended Lock block response format when Error_flag is set . . . 119
Table 89. Read Multiple Block request format . . . 119
Table 90. Read Multiple Block response format when Error_flag is NOT set . . . 120
Table 91. Block security status . . . 120
Table 92. Read Multiple Block response format when Error_flag is set . . . 120
Table 93. Extended Read Multiple Block request format . . . 121
Table 94. Extended Read Multiple Block response format when Error_flag is NOT set . . . 121
Table 95. Block security status . . . 122
Table 96. Extended Read Multiple Block response format when Error_flag is set . . . 122
Table 97. Write Multiple Block request format . . . 123
Table 98. Write Multiple Block response format when Error_flag is NOT set . . . 123
Table 99. Write Multiple Block response format when Error_flag is set . . . 123
Table 100. Extended Write Multiple Block request format . . . 124

Table 101. Extended Write Multiple Block response format when Error_flag is NOT set . . . 125
Table 102. Extended Write Multiple Block response format when Error_flag is set . . . 125
Table 103. Select request format . . . 126
Table 104. Select Block response format when Error_flag is NOT set . . . 126
Table 105. Select response format when Error_flag is set . . . 126
Table 106. Reset to Ready request format . . . 127
Table 107. Reset to Ready response format when Error_flag is NOT set . . . 127
Table 108. Reset to ready response format when Error_flag is set . . . 127
Table 109. Write AFI request format . . . 128
Table 110. Write AFI response format when Error_flag is NOT set . . . 128
Table 111. Write AFI response format when Error_flag is set . . . 128
Table 112. Lock AFI request format . . . 129
Table 113. Lock AFI response format when Error_flag is NOT set . . . 129
Table 114. Lock AFI response format when Error_flag is set . . . 129
Table 115. Write DSFID request format . . . 130
Table 116. Write DSFID response format when Error_flag is NOT set . . . 130
Table 117. Write DSFID response format when Error_flag is set . . . 131
Table 118. Lock DSFID request format . . . 131
Table 119. Lock DSFID response format when Error_flag is NOT set . . . 132
Table 120. Lock DSFID response format when Error_flag is set . . . 132
Table 121. Get System Info request format . . . 132
Table 122. Get System Info response format Error_flag is NOT set . . . 133
Table 123. Get System Info response format when Error_flag is set . . . 133
Table 124. Extended Get System Info request format . . . 134
Table 125. Parameter request list . . . 134
Table 126. Extended Get System Info response format when Error_flag is NOT set . . . 134
Table 127. Response Information Flag . . . 135
Table 128. Response other field: ST25DVxxx VICC memory size . . . 135
Table 129. Response other field: ST25DVxxx IC Ref . . . 136
Table 130. Response other field: ST25DVxxx VICC command list . . . 136
Table 131. Response other field: ST25DVxxx VICC command list Byte 1 . . . 136
Table 132. Response other field: ST25DVxxx VICC command list Byte 2 . . . 136
Table 133. Response other field: ST25DVxxx VICC command list Byte 3 . . . 137
Table 134. Response other field: ST25DVxxx VICC command list Byte 4 . . . 137
Table 135. Extended Get System Info response format when Error_flag is set . . . 137
Table 136. Get Multiple Block Security Status request format . . . 138
Table 137. Get Multiple Block Security Status response format when Error_flag is NOT set . . . 138
Table 138. Block security status . . . 139
Table 139. Get Multiple Block Security Status response format when Error_flag is set . . . 139
Table 140. Extended Get Multiple Block Security Status request format . . . 139
Table 141. Extended Get Multiple Block Security Status response format when Error_flags NOT set . . . 140
Table 142. Block security status . . . 140
Table 143. Extended Get Multiple Block Security Status response format when Error_flag is set . . . 140
Table 144. Read Configuration request format . . . 141
Table 145. Read Configuration response format when Error_flag is NOT set . . . 141
Table 146. Read Configuration response format when Error_flag is set . . . 141
Table 147. Write Configuration request format . . . 142
Table 148. Write Configuration response format when Error_flag is NOT set . . . 143
Table 149. Write Configuration response format when Error_flag is set . . . 143

Table 150. Read Dynamic Configuration request format . . . 144
Table 151. Read Dynamic Configuration response format when Error_flag is NOT set. . . . 144
Table 152. Read Dynamic Configuration response format when Error_flag is set . . . 144
Table 153. Write Dynamic Configuration request format . . . 145
Table 154. Write Dynamic Configuration response format when Error_flag is NOT set. . . . 145
Table 155. Write Dynamic Configuration response format when Error_flag is set . . . 145
Table 156. ManageGPO request format. . . . 146
Table 157. GPOVAL . . . 147
Table 158. ManageGPO response format when Error_flag is NOT set . . . 147
Table 159. ManageGPO response format when Error_flag is set . . . 147
Table 160. Write Message request format . . . 148
Table 161. Write Message response format when Error_flag is NOT set. . . . 148
Table 162. Write Message response format when Error_flag is set . . . 148
Table 163. Read Message Length request format . . . 149
Table 164. Read Message Length response format when Error_flag is NOT set . . . 149
Table 165. Read Message Length response format when Error_flag is set . . . 149
Table 166. Read Message request format . . . 150
Table 167. Read Message response format when Error_flag is NOT set . . . 150
Table 168. Write Password request format . . . 152
Table 169. Write Password response format when Error_flag is NOT set . . . 153
Table 170. Write Password response format when Error_flag is set . . . 153
Table 171. Present Password request format . . . 154
Table 172. Present Password response format when Error_flag is NOT set . . . 154
Table 173. Present Password response format when Error_flag is set . . . 155
Table 174. Fast Read Single Block request format . . . 155
Table 175. Fast Read Single Block response format when Error_flag is NOT set . . . 156
Table 176. Block security status . . . 156
Table 177. Fast Read Single Block response format when Error_flag is set . . . 156
Table 178. Fast Extended Read Single Block request format . . . 157
Table 179. Fast Extended Read Single Block response format when Error_flag is NOT set . . . 157
Table 180. Block security status . . . 157
Table 181. Fast Extended Read Single Block response format when Error_flag is set . . . 157
Table 182. Fast Read Multiple Block request format . . . 158
Table 183. Fast Read Multiple Block response format when Error_flag is NOT set. . . . 159
Table 184. Block security status if Option_flag is set . . . 159
Table 185. Fast Read Multiple Block response format when Error_flag is set . . . 159
Table 186. Fast Extended Read Multiple Block request format . . . 160
Table 187. Fast Extended Read Multiple Block response format when Error_flag is NOT set . . . 161
Table 188. Block security status if Option_flag is set . . . 161
Table 189. Fast Read Multiple Block response format when Error_flag is set . . . 161
Table 190. Fast Write Message request format . . . 162
Table 191. Fast Write Message response format when Error_flag is NOT set. . . . 162
Table 192. Fast Write Message response format when Error_flag is set . . . 162
Table 193. Fast Read Message Length request format . . . 163
Table 194. Fast Read Message Length response format when Error_flag is NOT set . . . 163
Table 195. Fast Read Message Length response format when Error_flag is set. . . . 163
Table 196. Fast Read Dynamic Configuration request format . . . 164
Table 197. Fast Read Dynamic Configuration response format when Error_flag is NOT set . . . 164

Table 198. Fast Read Dynamic Configuration response format when Error_flag is set . . . 165
Table 199. Fast Write Dynamic Configuration request format . . . 165
Table 200. Fast Write Dynamic Configuration response format when Error_flag is NOT set . . . 166
Table 201. Fast Write Dynamic Configuration response format when Error_flag is set . . . 166
Table 202. UID format . . . 167
Table 203. Absolute maximum ratings . . . 168
Table 204. I^2C operating conditions . . . 169
Table 205. AC test measurement conditions . . . 169
Table 206. Input parameters . . . 169
Table 207. I^2C DC characteristics . . . 170
Table 208. I^2C AC characteristics . . . 172
Table 209. GPO DC characteristics . . . 174
Table 210. GPO AC characteristics . . . 175
Table 211. RF characteristics . . . 175
Table 212. Operating conditions . . . 176
Table 213. SO8N – 8-lead plastic small outline, 150 mils body width, package mechanical data . . . 178
Table 214. TSSOP8 – 8-lead thin shrink small outline, 3 x 4.4 mm, 0.5 mm pitch, package mechanical data . . . 179
Table 215. UFDFN8 - 8-lead, 2 × 3 mm, 0.5 mm pitch ultra thin profile fine pitch dual flat package mechanical data . . . 181
Table 216. UFDFPN12 - 12-lead, 3x3 mm, 0.5 mm pitch ultra thin profile fine pitch dual flat package mechanical data . . . 183
Table 217. Ordering information scheme for packaged devices . . . 184
Table 218. ST25DVxxx Device select usage . . . 188
Table 219. Byte Write in user memory when write operation allowed . . . 188
Table 220. Polling during programming after byte writing in user memory . . . 189
Table 221. Byte Write in user memory when write operation is not allowed . . . 189
Table 222. Byte Write in Dynamic Register (if not Read Only) . . . 190
Table 223. Polling during programming after byte write in Dynamic Register . . . 190
Table 224. Byte Write in Dynamic Register if Read Only . . . 190
Table 225. Byte Write in mailbox when mailbox is free from RF message and Fast transfer Mode is activated . . . 191
Table 226. Byte Write in mailbox when mailbox is not free from RF message Fast transfer Mode is not activated . . . 192
Table 227. Byte Write in System memory if I2C security session is open and register is not RO . . . 192
Table 228. Polling during programing after byte write in System memory if I^2C security session is open and register is not RO . . . 193
Table 229. Byte Write in System memory if I^2C security session is closed or register is RO . . . 193
Table 230. Sequential write User memory when write operation allowed and all bytes belong to same area . . . 194
Table 231. Polling during programing after sequential write in User memory when write operation allowed and all bytes belong to same area . . . 194
Table 232. Sequential write in User memory when write operation allowed and crossing over area border . . . 195
Table 233. Polling during programing after sequential write in User memory when write operation allowed and crossing over area border . . . 196
Table 234. Sequential write in mailbox when mailbox is free from RF message and Fast transfer Mode is activated . . . 196

Table 235. Polling during programing after sequential write in mailbox 197
Table 236. Current byte Read in User memory if read operation allowed (depending on area protection and RF user security session) 197
Table 237. Current Read in User memory if read operation not allowed (depending on area protection and RF user security session) 197
Table 238. Random byte read in User memory if read operation allowed (depending on area protection and RF user security session) 198
Table 239. Random byte read in User memory if operation not allowed (depending on area protection and RF user security) 198
Table 240. Byte Read System memory (Static register or I2C Password after a valid Present I2C Password) 199
Table 241. Random byte read in Dynamic registers 199
Table 242. Sequential Read User memory if read operation allowed (depending on area protection and RF user security session) and all bytes belong to the same area 200
Table 243. Sequential Read User memory if read operation allowed (depending on area protection and RF user security session) but crossing area border 200
Table 244. Sequential Read User memory if read operation allowed (depending on area protection and RF user security session) 201
Table 245. Sequential in Read System memory (I^2C security session open if reading I2C_PWD) 202
Table 246. Sequential Read system memory when access is not granted (I^2C password I2C_PWD) 203
Table 247. Sequential read in dynamic register 203
Table 248. Sequential read in Dynamic register and mailbox continuously if Fast Transfer Mode is activated 204
Table 249. Sequential in mailbox if Fast Transfer Mode is activated 205
Table 250. Sequential read in mailbox if Fast Transfer Mode is not activated 206
Table 251. Write Password when I^2C security session is already open and Fast Transfer Mode is not activated 207
Table 252. Write Password when I^2C security session is not open or Fast Transfer Mode activated 208
Table 253. Document revision history 210

List of figures

Figure 1. ST25DVxxx block diagram . . . 17
Figure 2. ST25DVxxx 8-pin packages connections with Open drain Interruption Output . . . 18
Figure 3. ST25DVxxx 12-pin package connections with Cmos Interrupt Output (GPO) . . . 19
Figure 4. ST25DVxxx Power-Up sequence (No RF field, LPD pin tied to Vss or package without LPD pin) . . . 22
Figure 5. ST25DVxxx RF Power Up sequence (No DC supply) . . . 23
Figure 6. Memory organization . . . 25
Figure 7. ST25DVxxx user memory areas . . . 27
Figure 8. RF to I^2C fast transfer mode operation . . . 40
Figure 9. I^2C to RF fast transfer mode operation . . . 41
Figure 10. Fast Transfer Mode mailbox access management . . . 43
Figure 11. RF_USER chronogram . . . 45
Figure 12. RF_ACTIVITY chronogram . . . 46
Figure 13. RF_INTERRUPT chronogram . . . 47
Figure 14. FIELD_CHANGE chronogram . . . 48
Figure 15. RF_PUT_MSG chronogram . . . 49
Figure 16. RF_GET_MSG chronogram . . . 50
Figure 17. RF_WRITE chronogram . . . 52
Figure 18. EH delivery state diagram . . . 61
Figure 19. ST25DVxxx Energy Harvesting Delivery Sequence . . . 62
Figure 20. ST25DVxxx, Arbitration between RF and I^2C . . . 65
Figure 21. RF security sessions management . . . 77
Figure 22. I2C security sessions management . . . 78
Figure 23. I^2C bus protocol . . . 85
Figure 24. I²C timeout on Start condition . . . 87
Figure 25. Write mode sequences when write is not inhibited . . . 90
Figure 26. Write mode sequences when write is inhibited . . . 90
Figure 27. Write cycle polling flowchart using ACK . . . 91
Figure 28. Read mode sequences . . . 93
Figure 29. I^2C Present Password Sequence . . . 95
Figure 30. I^2C Write Password Sequence . . . 96
Figure 31. ST25DVxxx protocol timing . . . 99
Figure 32. ST25DVxxx state transition diagram . . . 100
Figure 33. Stay Quiet frame exchange between VCD and ST25DVxxx . . . 111
Figure 34. Read Single Block frame exchange between VCD and ST25DVxxx . . . 112
Figure 35. Extended Read Single Block frame exchange between VCD and ST25DVxxx . . . 113
Figure 36. Write Single Block frame exchange between VCD and ST25DVxxx . . . 115
Figure 37. Extended Write Single frame exchange between VCD and ST25DVxxx . . . 116
Figure 38. Lock single block frame exchange between VCD and ST25DVxxx . . . 118
Figure 39. Extended Lock block frame exchange between VCD and ST25DVxxx . . . 119
Figure 40. Read Multiple Block frame exchange between VCD and ST25DVxxx . . . 121
Figure 41. Extended Read Multiple Block frame exchange between VCD and ST25DVxxx . . . 122
Figure 42. Write Multiple Block frame exchange between VCD and ST25DVxxx . . . 124
Figure 43. Extended Write Multiple Block frame exchange between VCD and ST25DVxxx . . . 125
Figure 44. Select frame exchange between VCD and ST25DVxxx . . . 126
Figure 45. Reset to Ready frame exchange between VCD and ST25DVxxx . . . 127

Figure 46. Write AFI frame exchange between VCD and ST25DVxxx . . . 129
Figure 47. Lock AFI frame exchange between VCD and ST25DVxxx . . . 130
Figure 48. Write DSFID frame exchange between VCD and ST25DVxxx . . . 131
Figure 49. Lock DSFID frame exchange between VCD and ST25DVxxx . . . 132
Figure 50. Get System Info frame exchange between VCD and ST25DVxxx . . . 133
Figure 51. Extended Get System Info frame exchange between VCD and ST25DVxxx . . . 138
Figure 52. Get Multiple Block Security Status frame exchange between VCD and ST25DVxxx . . . 139
Figure 53. Extended Get Multiple Block Security Status frame exchange between VCD and ST25DVxxx . . . 141
Figure 54. Read Configuration frame exchange between VCD and ST25DVxxx . . . 142
Figure 55. Write Configuration frame exchange between VCD and ST25DVxxx . . . 143
Figure 56. Read Dynamic Configuration frame exchange between VCD and ST25DVxxx . . . 145
Figure 57. Write Dynamic Configuration frame exchange between VCD and ST25DVxxx . . . 146
Figure 58. ManageGPO frame exchange between VCD and ST25DVxxx . . . 147
Figure 59. Write Message frame exchange between VCD and ST25DVxxx . . . 149
Figure 60. Read Message Length frame exchange between VCD and ST25DVxxx . . . 150
Figure 61. Read Message frame exchange between VCD and ST25DVxxx . . . 151
Figure 62. Fast Read Message frame exchange between VCD and ST25DVxxx . . . 152
Figure 63. Write Password frame exchange between VCD and ST25DVxxx . . . 154
Figure 64. Present Password frame exchange between VCD and ST25DVxxx . . . 155
Figure 65. Fast Read Single Block frame exchange between VCD and ST25DVxxx . . . 156
Figure 66. Fast Extended Read Single Block frame exchange between VCD and ST25DVxxx . . . 158
Figure 67. Fast Read Multiple Block frame exchange between VCD and ST25DVxxx . . . 160
Figure 68. Fast Extended Read Multiple Block frame exchange between VCD and ST25DVxxx . . . 161
Figure 69. Fast Write Message frame exchange between VCD and ST25DVxxx . . . 163
Figure 70. Fast Read Message Length frame exchange between VCD and ST25DVxxx . . . 164
Figure 71. Fast Read Dynamic Configuration frame exchange between VCD and ST25DVxxx . . . 165
Figure 72. Fast Write Dynamic Configuration frame exchange between VCD and ST25DVxxx . . . 166
Figure 73. AC test measurement I/O waveform . . . 169
Figure 74. I^2C AC waveforms . . . 173
Figure 75. I^2C Fast mode (f_C = 1 MHz): maximum R_{bus} value versus bus parasitic capacitance (C_{bus}) . . . 174
Figure 76. ASK modulated signal . . . 177
Figure 77. SO8N – 8-lead plastic small outline, 150 mils body width, package outline . . . 178
Figure 78. TSSOP8 – 8-lead thin shrink small outline, 3 x 4.4 mm, 0.5 mm pitch, package outline . . . 179
Figure 79. UFDFN8 - 8-lead, 2 × 3 mm, 0.5 mm pitch ultra thin profile fine pitch dual flat package outline . . . 181
Figure 80. UFDFPN12 - 12-lead, 3x3 mm, 0.5 mm pitch ultra thin profile fine pitch dual flat package outline . . . 183
Figure 81. Logic 0, high data rate, fast commands . . . 185
Figure 82. Logic 1, high data rate, fast commands . . . 185
Figure 83. Logic 0, low data rate, fast commands . . . 185
Figure 84. Logic 1, low data rate, fast commands . . . 186

Figure 85. Start of frame, high data rate, one subcarrier, fast commands 186
Figure 86. Start of frame, low data rate, one subcarrier, fast commands 186
Figure 87. End of frame, high data rate, one subcarrier, fast commands 187
Figure 88. End of frame, low data rate, one subcarrier, fast commands 187

1 Description

The ST25DVxxx device is a NFC RFID Tag offering 4 Kbit, 16 Kbit, and 64 Kbit of electrically erasable programmable memory (EEPROM). ST25DVxxx offers two interfaces. The first one is an I^2C serial link and can be operated from a DC power supply. The second one is a RF link activated when ST25DVxxx acts as a contactless memory powered by the received carrier electromagnetic wave.

In I^2C mode, the ST25DVxxx user memory contains up to 8192 bytes, which could be split in 4 flexible and protectable areas.

In RF mode, following ISO/IEC 15693 or NFC forum type 5 recommendations, ST25DVxxx user memory contains up to 2048 blocks of 4 bytes which could be split in 4 flexible and protectable areas.

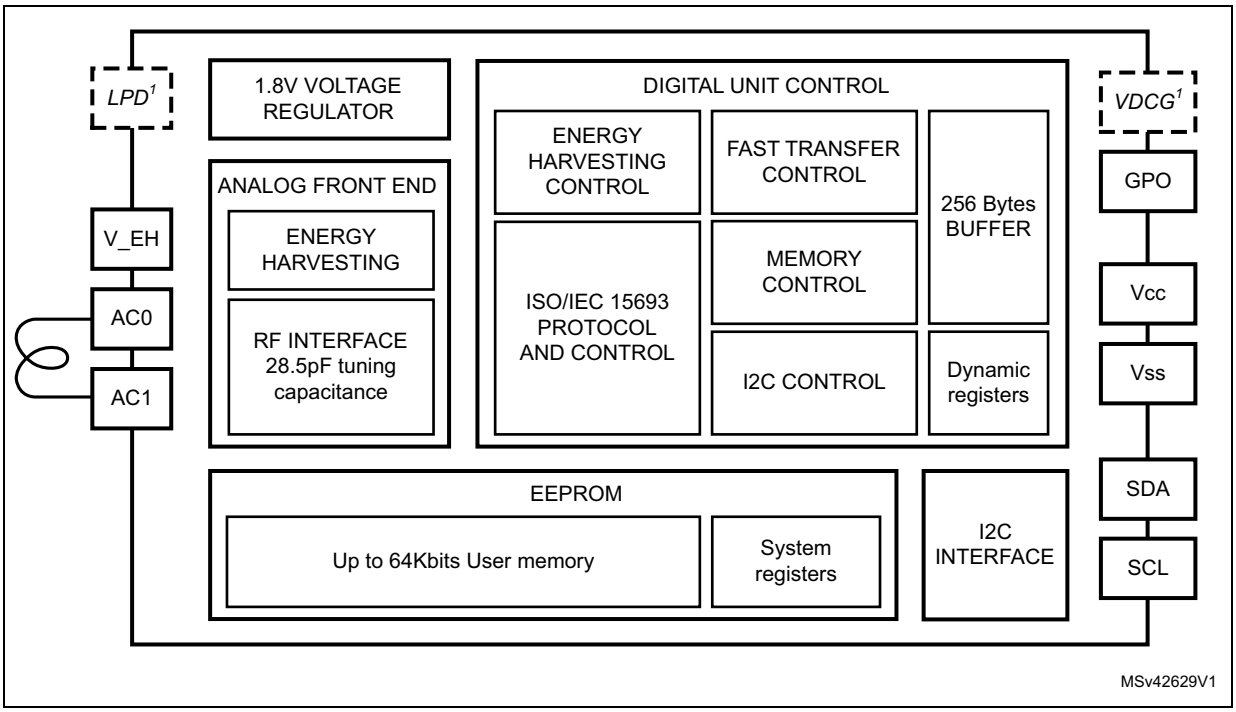
ST25DVxxx offers a fast transfer mode between the RF and contact worlds, thanks to a 256 bytes volatile buffer (also called Mailbox).

In addition, the GPO pin of the ST25DVxxx provides data informing the contact world about incoming events, like RF field detection, RF activity in progress or mailbox message availability.

An energy harvesting feature is also proposed when external conditions make it possible.

1.1 ST25DVxxx block diagram

Figure 1. ST25DVxxx block diagram



1. V_{DCG} and LPD are included in 12 pins package only

1.2 ST25DVxxx packaging

ST25DVxxx is provided in different packages:

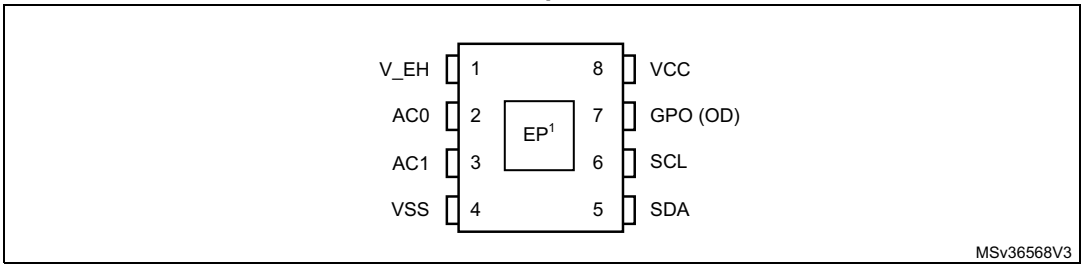
- 8 pins (S08 or TSSPOP8 or UFDFPN8) for the open drain version of Interrupt output
- 12 pins (UFDFPN12) for a CMOS interrupt output. This package includes an additional element that minimizes standby consumption.

Table 2. Signal names

Signal name	Function	Direction
V_EH	Energy Harvesting	Power output
GPO	Interrupt Output	Output
SDA	Serial Data	I/O
SCL	Serial Clock	Input
AC0, AC1	Antenna coils	
V_{CC}	Supply voltage	Power
V_{SS}	Ground	
LPD(1)	Low power down mode	Input
V_{DCG}(1)	Supply voltage for GPO driver	Power
NC	Not connected	Must be left floating
EP(2)	Exposed Pad	Must be left floating

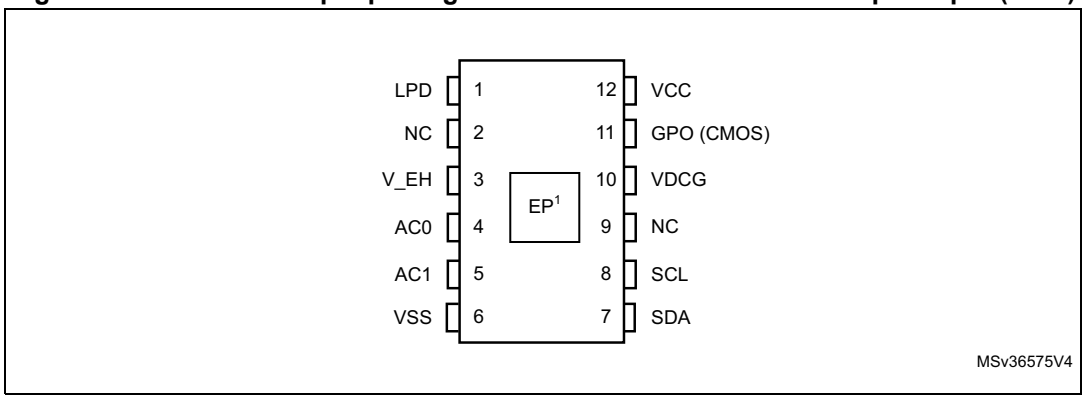
1. Available only on 12-pin package.
2. Available only on UFDPN8 and UFDFPN12 packages.

Figure 2. ST25DVxxx 8-pin packages connections with Open drain Interruption Output



1. Exposed Pad is only present on UFDFPN8 package.

Figure 3. ST25DVxxx 12-pin package connections with Cmos Interrupt Output (GPO)



1. Exposed Pad is only present on UFDFPN12 package.

2 Signal descriptions

2.1 Serial link (SCL, SDA)

2.1.1 Serial clock (SCL)

This input signal is used to strobe all data in and out of the ST25DVxxx. In applications where this signal is used by slave devices to synchronize the bus to a slower clock, the bus master must have an open drain output, and a pull-up resistor must be connected from Serial Clock (SCL) to V_{CC}. See *Section 9.2* to know how to calculate the value of this pull-up resistor

2.1.2 Serial data (SDA)

This bidirectional signal is used to transfer data in or out of the ST25DVxxx. It is an open drain output that may be wire-OR'ed with other open drain or open collector signals on the bus. A pull-up resistor must be connected from Serial Data (SDA) to V_{CC}. (*Figure 75* indicates how the value of the pull-up resistor can be calculated).

2.2 Power control (V_{CC}, LPD,V_{SS})

2.2.1 Supply voltage (V_{CC})

This pin can be connected to an external DC supply voltage.

Note: An internal voltage regulator allows the external voltage applied on V_{CC} to supply the ST25DVxxx, while preventing the internal power supply (rectified RF waveforms) to output a DC voltage on the V_{CC} pin.

2.2.2 Low Power Down (LPD)

This input signal is used to control an internal 1.8 V regulator delivering ST25DVxxx internal supply. When LPD is high, this regulator is shut off and its consumption is reduced below 1µA. This regulator has a turn on time in range of 100us, to be added to the boot duration, before the device becomes fully operational. This feature is only available on the 12-pin ST25DVxxx package.

2.2.3 Ground (V_{SS})

V_{SS} is the reference for the V_{CC} and V_{DCG} supply voltages and V_EH analog output voltage.

2.3 RF link (AC0 AC1)

2.3.1 Antenna coil (AC0, AC1)

These inputs are used to connect the ST25DVxxx device to an external coil exclusively. It is advised not to connect any other DC or AC path to AC0 or AC1.

When correctly tuned, the coil is used to power and access the device using the ISO/IEC 15693 and ISO 18000-3 mode 1 protocols.

2.4 Process control (V_{DCG}, GPO)

2.4.1 Driver Supply voltage (V_{DCG})

This pin, available only with ST25DVxx-JF version, can be connected to an external DC supply voltage. It only supplies the GPO driver block. ST25DVxxx cannot be powered by V_{DCG}. If V_{DCG} is left floating, no information will be available on GPO pin.

2.4.2 General purpose output (GPO)

The ST25DVxxx features a configurable output GPO pin used to provide RF activity information to an external device. ST25DVxx-IE offers a GPO open drain. This GPO pin must be connected to an external pull-up resistor (> 4.7 KΩ) to operate.

The interrupt consists in pulling the state to a low level or outputting a low-level pulse on GPO pin.

ST25DVxx-JF offers a GPO CMOS output, which requires to connect V_{DCG} pin to an external power supply. The interrupt consists in setting the state to a high level or outputting a positive pulse on the GPO pin.

GPO pin is a configurable output signal, and can mix several Interruption modes. By default, the GPO register sets the interruption mode as a RF Field Change detector. It is able to raise various events like RF Activity, Memory Write completion, or fast transfer actions. It can authorize the RF side to directly drive GPO pin using the Manage GPO command to set the output state or emit a single pulse (for example, to wake up an application.). See *Section 5.2: GPO* for details.

2.5 Energy harvesting analog output (V_EH)

This analog output pin is used to deliver the analog voltage V_EH available when the Energy harvesting mode is enabled and if the RF field strength is sufficient. When the Energy harvesting mode is disabled or the RF field strength is not sufficient, V_EH pin is in High-Z state (See *Section 5.3: Energy Harvesting (EH)* for details).

3 Power management

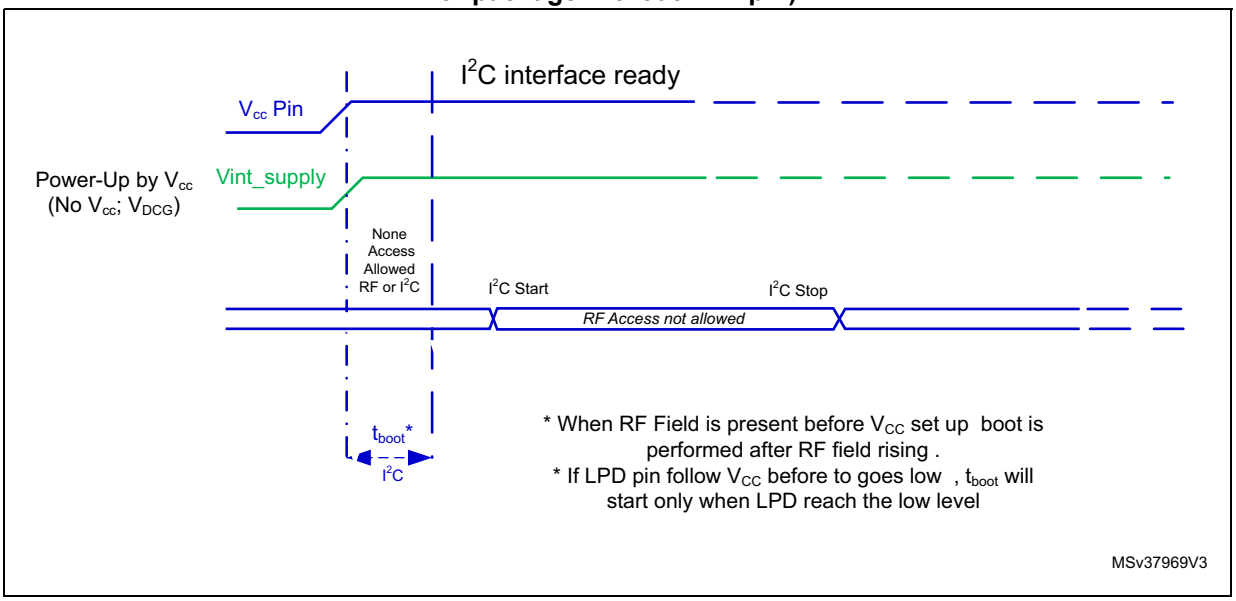
3.1 Wired interface

Operating supply voltage V_{CC}

In contact mode, prior to selecting the memory and issuing instructions to it, a valid and stable V_{CC} voltage within the specified [V_{CC}(min), V_{CC}(max)] range must be applied (see *Table 204: I²C operating conditions*). To maintain a stable DC supply voltage, it is recommended to decouple the V_{CC} line with a suitable capacitor (usually of the order of 10 nF and 100 pF) close to the V_{CC}/V_{SS} package pins.

This voltage must remain stable and valid until the end of the transmission of the instruction and, for a Write instruction, until the completion of the internal I²C write cycle (t_W). Instructions are not taken into account until completion of ST25DVxxx's boot sequence (see *Figure 4*).

Figure 4. ST25DVxxx Power-Up sequence (No RF field, LPD pin tied to V_{ss} or package without LPD pin)



Power-up conditions

When the power supply is turned on, V_{CC} rises from V_{SS} to V_{CC}. The V_{CC} rise time must not vary faster than 1V/µs.

Device reset in I²C mode

In order to prevent inadvertent write operations during power-up, a power-on reset (POR) circuit is included. At power-up (continuous rise of V_{CC}), the ST25DVxxx does not respond to any I²C instruction until V_{CC} has reached the power-on reset threshold voltage (this threshold is lower than the minimum V_{CC} operating voltage defined in *Table 204: I²C operating conditions*). When V_{CC} passes over the POR threshold, the device is reset and enters the Standby power mode. However, the device must not be accessed until V_{CC} has reached a valid and stable V_{CC} voltage within the specified [$V_{CC(min)}$, $V_{CC(max)}$] range and

t_boot time necessary to ST25DVxxx set-up has passed. In the version supporting LPD pin, the boot will take place only when LPD goes low.

In a similar way, during power-down (continuous decrease in V_{CC}), as soon as V_{CC} drops below the power-on reset threshold voltage, the device stops responding to any instruction sent to it, and I^2C address counter is reset.

Power-down mode

During power-down (continuous decay of V_{CC}), the device must be in Standby power mode (mode reached after decoding a Stop condition, assuming that there is no internal write cycle in progress).

3.2 Contactless interface

Device set in RF mode

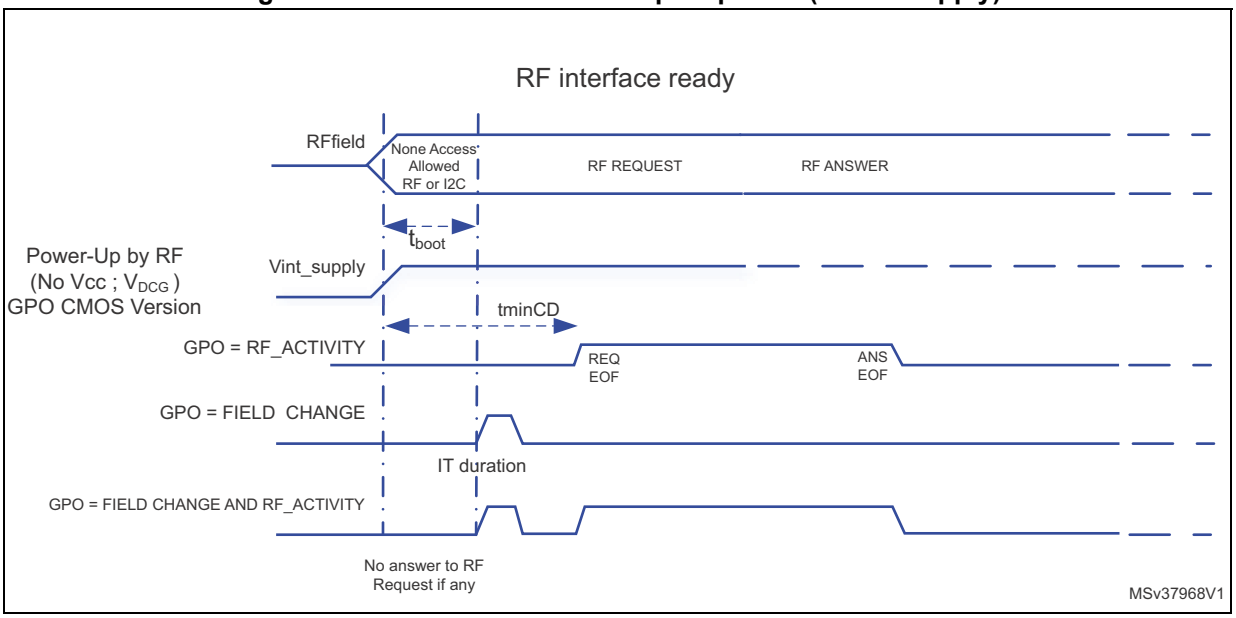
To ensure a proper boot of the RF circuitry, the RF field must be turned ON without any modulation for a minimum period of time t_{RF_ON}. Before this time, ST25DVxxx will ignore all received RF commands. (See *Figure 5: ST25DVxxx RF Power Up sequence (No DC supply)*).

Device reset in RF mode

To ensure a proper reset of the RF circuitry, the RF field must be turned off (100% modulation) for a minimum t_{RF_OFF} period of time.

The RF access can be temporarily or indefinitely disabled by setting the appropriate value in the RF disable register.

Figure 5. ST25DVxxx RF Power Up sequence (No DC supply)



4 Memory management

4.1 Memory organization overview

The ST25DVxxx memory is divided in four main memory areas:

- User memory
- Dynamic registers
- Fast Transfer Mode buffer
- System configuration area

The ST25DVxxx user memory can be divided into 4 flexible user areas. Each area can be individually read - and/or - write-protected with one out of three specific 64-bit password.

The ST25DVxxx dynamic registers are accessible by RF or I^2C host and provide dynamic activity status or allow temporary activation or deactivation of some ST25DVxxx features.

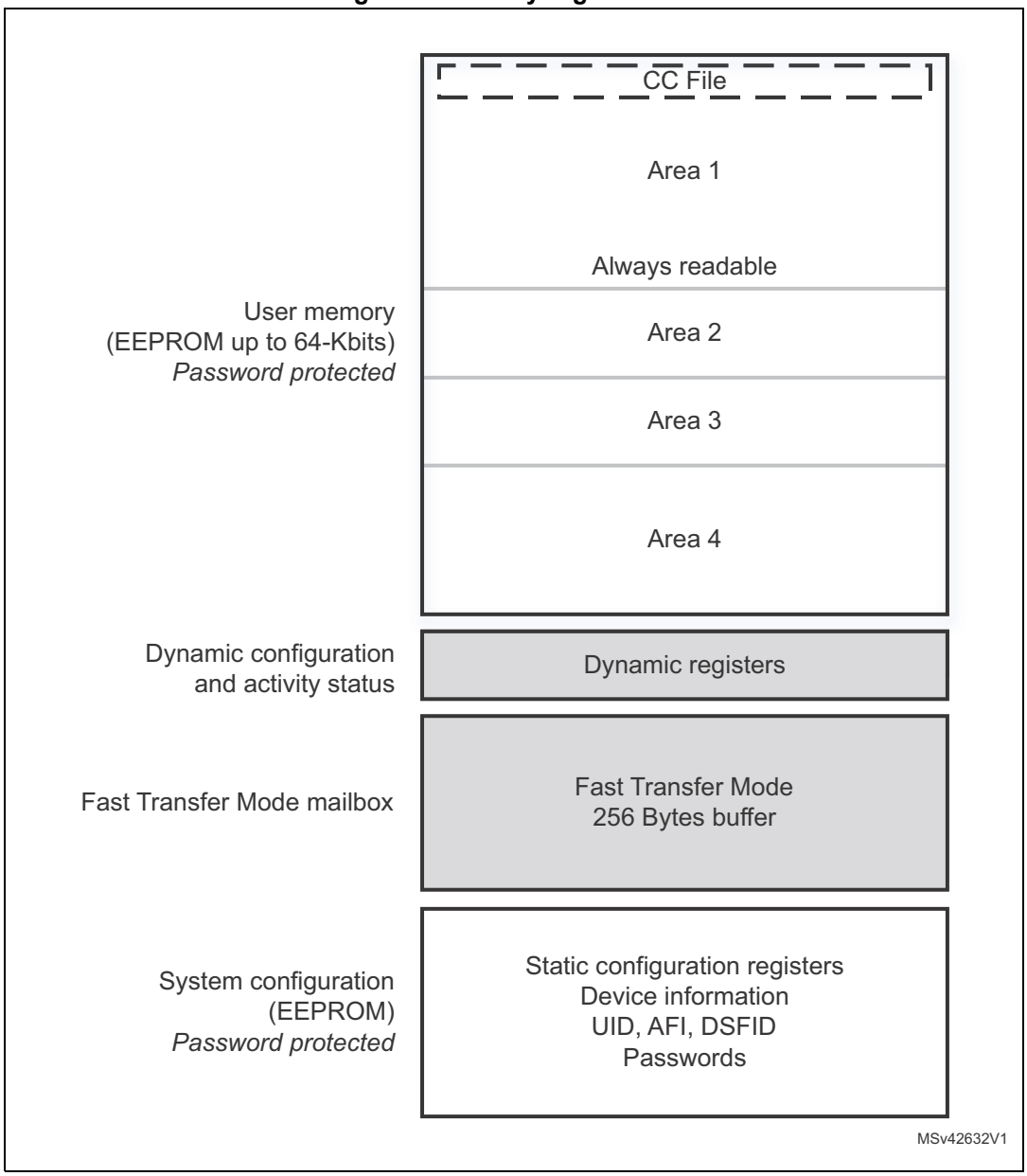
The ST25DVxxx also provides a 256 byte Fast Transfer Mode buffer, acting as a mailbox between RF and I^2C interface, allowing fast data transfer between contact and contactless worlds.

Finally, the ST25DVxxx system configuration area contains static registers to configure all ST25DVxxx features, which can be tuned by user. Its access is protected by a 64 bit configuration password.

This system configuration area also includes read only device information such as IC reference, memory size or IC revision, as well as a 64-bit block that is used to store the 64-bit unique identifier (UID), and the AFI (default 00h) and DSFID (default 00h) registers. The UID is compliant with the ISO 15693 description, and its value is used during the anticollision sequence (Inventory). The UID value is written by ST on the production line. The AFI register stores the application family identifier. The DSFID register stores the data storage family identifier used in the anticollision algorithm.

The system configuration area includes five additional 64-bit blocks that store an I^2C password plus three RF user area access passwords and a RF configuration password.

Figure 6. Memory organization



4.2 User memory

User memory is accessible from both RF contactless interface and I^2C wired interface.

From RF interface, user memory is addressed as Blocks of 4 bytes, starting at address 0. RF extended read and write commands can be used to address all ST25DVxxx memory blocks. Other read and write commands can only address up to block FFh.

From I^2C interface, user memory is addressed as Bytes, starting at address 0. Device select must set E2 = 0. User memory can be read in continuity. Unlike the RF interface, there is no roll-over when the requested address reaches the end of the memory capacity.

Table 3: User memory as seen by RF and by I2C shows how memory is seen from RF interface and from I^2C interface.

Table 3. User memory as seen by RF and by I^2C

RF command (block addressing)	User memory				I^2C command (byte addressing)
Read Single Block Read Multiple Blocks Fast Read Single Block Fast Read Multiple Blocks Write Single Block Write Multiple Blocks Ext Read Single Block Ext Read Multiple Blocks Fast Ext Read Single Block Fast Ext Read Multi. Blocks Ext Write Single Block Ext Write Multiple Blocks	RF block (00)00h				I^2C Read command I^2C Write command Device select E2 = 0
	I^2C byte 0003h	I^2C byte 0002h	I^2C byte 0001h	I^2C byte 0000h	
	RF block (00)01h				
	I^2C byte 0007h	I^2C byte 0006h	I^2C byte 0005h	I^2C byte 0004h	
	RF block (00)02h				
	I^2C byte 000Bh	I^2C byte 000Ah	I^2C byte 0009h	I^2C byte 0008h	
	....				
	RF block (00)7Fh(1)				
	I^2C byte 01FFh	I^2C byte 01FEh	I^2C byte 01FDh	I^2C byte 01FCh	
	....				
	RF block (00)FFh(2)				
	I^2C byte 03FFh	I^2C byte 03FEh	I^2C byte 03FDh	I^2C byte 03FCh	
Ext Read Single Block Ext Read Multiple Blocks Fast Ext Read Single Block Fast Ext Read Multi. Blocks Ext Write Single Block Ext Write Multiple Blocks	RF block 0100h				
	I^2C byte 0403h	I^2C byte 0402h	I^2C byte 0401h	I^2C byte 0400h	
	....				
	RF block 01FFh(3)				
	I^2C byte 07FFh	I^2C byte 07FEh	I^2C byte 07FDh	I^2C byte 07FCh	
	....				
	RF block 07FFh(4)				
	I^2C byte 1FFFh	I^2C byte 1FFEh	I^2C byte 1FFDh	I^2C byte 1FFCh	

1. Last block of user memory in ST25DV04K-XX.
2. Last block accessible with Read Single Block, Read Multiple Blocks, Fast Read Single Block, Fast Read Multiple Blocks, Write Single Block and Write Multiple Blocks RF commands.
3. Last block of user memory in ST25DV16K-XX.
4. Last block of user memory in ST25DV64K-XX.

4.2.1 User memory areas

The user memory can be split into different areas, each one with a distinct access privilege.

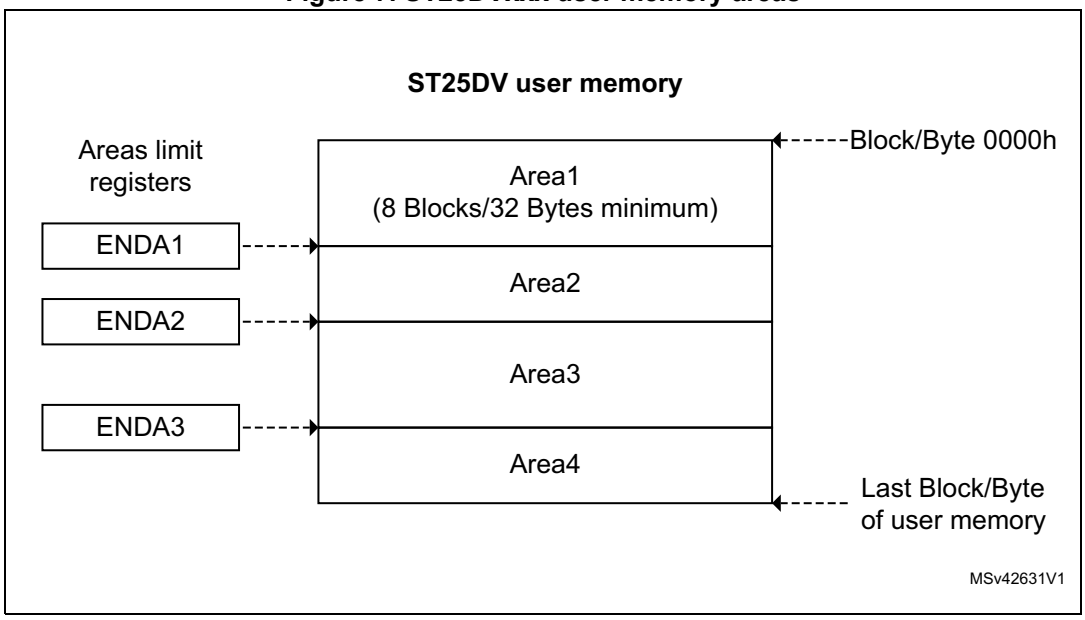
RF and I^2C read and write commands are legal only within a same zone:

- In RF, a multiple read or a multiple write command is not executed and returns the error code 0Fh if addresses cross the area borders.
- In I^2C, a read data always return FFh after crossing an area border. A write command is not acknowledged and not executed if the command crosses the area border.

Each user memory area is defined by its ending block address $ENDA_i$. The starting block address is defined by the end of the preceding area.

There are three $ENDA_i$ registers in the configuration system memory, used to define the end block addresses of Area 1, Area 2 and Area 3. The end of Area 4 is always the last block of memory and is not configurable.

Figure 7. ST25DVxxx user memory areas



On factory delivery all $ENDA_i$ are set to maximum value, only Area1 exists and includes the full user memory.

A granularity of 8 Blocks (32 Bytes) is offered to code area ending points.

An area's end limit is coded as followed in $ENDA_i$ registers:

- Last RF block address of area = 8 x $ENDA_i$ + 7 => $ENDA_i$ = int(Last $Area_i$ RF block address / 8)
- Last I^2C byte address of area = 32 * $ENDA_i$ + 31 => $ENDA_i$ = int(Last $Area_i$ I^2C byte address / 32)
- As a consequence, ENDA1 = 0 means size of Area 1 is 8 blocks (32 Bytes).

Table 4. Maximum user memory Block and Byte addresses and $ENDA_i$ value

Device	Last user memory block address seen by RF	Last user memory byte address seen by I^2C	Maximum $ENDA_i$ value
ST25DV04K-xx	007Fh	01FFh	0Fh
ST25DV16K-xx	01FFh	07FFh	3Fh
ST25DV64K-xx	07FFh	1FFFh	FFh

Table 5. Areas and limit calculation from $ENDA_i$ registers

Area	Seen from RF interface	Seen from I^2C interface
Area 1	Block 0000h … Block (ENDA1*8)+7	Byte 0000h … Byte (ENDA1*32)+31
Area 2	Block (ENDA1+1)*8 … Block (ENDA2*8)+7	Byte (ENDA1+1)*32 … Byte (ENDA2*32)+31
Area 3	Block (ENDA2+1)*8 … Block (ENDA3*8)+7	Byte (ENDA2+1)*32 … Byte (ENDA3*32)+31
Area 4	Block (ENDA3+1)*8 … Last memory Block	Byte (ENDA3+1)*32 … Last memory Byte

Organization of user memory in areas have the following characteristics:

- At least one area exists (Area1), starting at Block/Byte address 0000h and finishing at ENDA1, with ENDA1 = ENDA2 = ENDA3 = End of user memory (factory setting).
- Two Areas could be defined by setting ENDA1 < ENDA2 = ENDA3 = End of user memory.
- Three Areas may be defined by setting ENDA1 < ENDA2 < ENDA3 = End of user memory.
- A maximum of four areas may be defined by setting ENDA1 < ENDA2 < ENDA3 < End of user memory.
- Area 1 specificities
  - Start of Area1 is always Block/Byte address 0000h.
  - Area1 minimum size is 8 Blocks (32 Bytes) when ENDA1 = 00h.
  - Area1 is always readable.
- The last area always finishes on the last user memory Block/Byte address (ENDA4 doesn't exist).
- All areas are contiguous: end of Area(n) + one Block/Byte address is always start of Area(n+1).

Area size programming

RF user must first open the RF configuration security session to write ENDAi registers.

I^2C host must first open I^2C security session to write ENDAi registers.

When programming an ENDAi register, the following rule must be respected:

- ENDAi-1 < ENDAi ≤ ENDAi+1 = End of memory.

This means that prior to programming any ENDAi register, its successor (ENDAi+1) must first be programmed to the last Block/Byte of memory:

- Successful ENDA3 programming condition: ENDA2 < ENDA3 ≤ End of user memory
- Successful ENDA2 programming condition: ENDA1 < ENDA2 ≤ ENDA3 = End of user memory
- Successful ENDA1 programming condition: ENDA1 ≤ ENDA2 = ENDA 3 = End of user memory

If this rule is not respected, an error 0Fh is returned in RF, NoAck is returned in I2C, and programming is not done.

In order to respect this rule, the following procedure is recommended when programming Areas size (even for changing only one Area size):

1. Ends of Areas 3 and 2 must first be set to the end of memory while respecting the following order:
   a) If ENDA3 ≠ end of user memory, then set ENDA3 = end of memory; else, do not write ENDA3.
   b) If ENDA2 ≠ end of user memory, then set ENDA2 = end of memory; else, do not write ENDA2.
2. Then, desired area limits can be set respecting the following order:
   a) Set new ENDA1 value.
   b) Set new ENDA2 value, with ENDA2 > ENDA1
   c) Set new ENDA3 value, with ENDA3 > ENDA2

Example of successive user memory area setting (for a ST25DV64K-xx):

1. Initial state, 2 Areas are defined:
   a) ENDA1 = 10h (Last block of Area 1: (10h x 8) + 7 = 0087h)
   b) ENDA2 = FFh (Last block of Area 2: (FFh x 8) + 7 = 07FFh)
   c) ENDA3 = FFh (No Area 3)
      – Area 1 from Block 0000h to 0087h (136 Blocks)
      – Area 2 from Block 0088h to 07FFh (1912 Blocks)
      – There is no Area 3.
      – There is no Area 4.

2. Split of user memory in four areas:
   a) ENDA3 is not updated as it is already set to end of memory.
   b) ENDA2 is not updated as it is already set to end of memory.
   c) Set ENDA1 = 3Fh (Last block of Area 1: (3Fh x 8) + 7 = 01FFh)
   d) Set ENDA2 = 5Fh (Last block of Area 1: (5Fh x 8) + 7 = 02FFh)
   e) Set ENDA3 = BFh (Last block of Area 1: (BFh x 8) + 7 = 05FFh)
      – Area1 from Block 0000h to 01FFh (512 Blocks)
      – Area2 from Block 0200h to 02FFh (256 Blocks)
      – Area3 from Block 0300h to 05FFh (768 Blocks)
      – Area4 from Block 0600h to 07FFh (512 Blocks).

3. Return to a split in two equal areas:
   a) Set ENDA3 = FFh
   b) Set ENDA2 = FFh
   c) Set ENDA1 = 7Fh (Last block of Area 1: (7Fh x 8) + 7 = 03FFh)
      – Area1 from Block 0000h to 03FFh (1024 Blocks)
      – Area2 from Block 0400h to 07FFh (1024 Blocks)
      – There is no Area3.
      – There is no Area4.

Programming ENDA3 to FFh in step 2.a would have resulted in into an error, since rule ENDAi-1 < ENDAi would not been respected (ENDA2 = ENDA3 in that case).

Registers for user memory area configuration

Table 6. ENDA1(1)

RF	**Command**	**Read Configuration (cmd code A0h) @05h** **Write Configuration (cmd code A1h) @05h**	
	Type	**R always, W if RF configuration security session is open and configuration not locked**	
I^2C	**Address**	**E2 = 1, 0005h**	
	Type	**R always, W if I^2C security session is open**	
Bit	**Name**	**Function**	**Factory Value**
b7-b0	ENDA1	End Area 1 = 8*ENDA1+7 when expressed in blocks (RF) End Area 1 = 32*ENDA1+31 when expressed in bytes (I^2C)	ST25DV04K-XX: 0Fh ST25DV16K-XX: 3Fh ST25DV64K-XX: FFh

1. Refer to *Table 9: System configuration memory map* for the ENDA1 register.

Table 7. ENDA2(1)

RF	**Command**	**Read Configuration (cmd code A0h) @07h Write Configuration (cmd code A1h) @07h**	
	Type	**R always, W if RF configuration security session is open and configuration not locked**	
I^2C	**Address**	**E2 = 1, 0007h**	
	Type	**R always, W if I^2C security session is open**	
Bit	**Name**	**Function**	**Factory Value**
b7-b0	ENDA2	End Area 2 = 8 x ENDA2 + 7 when expressed in blocks (RF) End Area 2 = 32*ENDA2 + 31 when expressed in bytes (I^2C)	ST25DV04K-XX: 0Fh ST25DV16K-XX: 3Fh ST25DV64K-XX: FFh

1. Refer to *Table 9: System configuration memory map* for the ENDA2 register.

Table 8. ENDA3(1)

RF	**Command**	**Read Configuration (cmd code A0h) @09h Write Configuration (cmd code A1h) @09h**	
	Type	**R always, W if RF configuration security session is open and configuration not locked**	
I^2C	**Address**	**E2 = 1, 0009h**	
	Type	**R always, W if I^2C security session is open**	
Bit	**Name**	**Function**	**Factory Value**
b7-b0	ENDA3	End Area 3 = 8 x ENDA3 + 7 when expressed in blocks (RF) End Area 3 = 32 x ENDA3 + 31 when expressed in bytes (I^2C)	ST25DV04K-XX: 0Fh ST25DV16K-XX: 3Fh ST25DV64K-XX: FFh

1. Refer to *Table 9: System configuration memory map* for the ENDA3 register.

4.3 System configuration area

In addition to EEPROM user memory, ST25DVxxx includes a set of static registers located in the system configuration area memory (EEPROM nonvolatile registers). Those registers are set during device configuration (i.e.: area extension), or by the application (i.e.: area protection). Static registers content is read during the boot sequence and define basic ST25DVxxx behavior.

In RF, the static registers located in the system configuration area can be accessed via dedicated Read Configuration and Write Configuration commands, with a pointer acting as the register address.

The RF configuration security session must first be open, by presenting a valid RF configuration password, to grant write access to system configuration registers.

The system configuration area write access by RF can also be deactivated by I^2C host.

In I^2C static registers located in the system configuration area can be accessed with I^2C read and write commands with device select E2=1. Readable system areas could be read in continuity.

I^2C security session must first be open, by presenting a valid I^2C password, to grant write access to system configuration registers.

Table 9 shows the complete map of the system configuration area, as seen by RF and I^2C interface.

Table 9. System configuration memory map

RF access		Static Register		I^2C access		
Address	**Type**	**Name**	**Function**	**Device select**	**Address**	**Type**
00h	RW(1)	*Table 19: GPO*	Enable/disable ITs on GPO	E2=1	0000h	RW(2)
01h	RW(1)	*Table 20: IT_TIME*	Interruption pulse duration	E2=1	0001h	RW(2)
02h	RW(1)	*Table 24: EH_MODE*	Energy Harvesting default strategy after Power ON	E2=1	0002h	RW(2)
03h	RW(1)	*Table 27: RF_MNGT*	RF interface state after Power ON	E2=1	0003h	RW(2)
04h	RW(1)	*Table 29: RFA1SS*	Area1 RF access protection	E2=1	0004h	RW(2)
05h	RW(1)	*Table 6: ENDA1*	Area 1 ending point	E2=1	0005h	RW(2)
06h	RW(1)	*Table 30: RFA2SS*	Area2 RF access protection	E2=1	0006h	RW(2)
07h	RW(1)	*Table 7: ENDA2*	Area 2 ending point	E2=1	0007h	RW(2)
08h	RW(1)	*Table 31: RFA3SS*	Area3 RF access protection	E2=1	0008h	RW(2)
09h	RW(1)	*Table 8: ENDA3*	Area 3 ending point	E2=1	0009h	RW(2)
0Ah	RW(1)	*Table 32: RFA4SS*	Area4 RF access protection	E2=1	000Ah	RW(2)
No access		*Table 33: I2CSS*	Area 1 to 4 I^2C access protection	E2=1	000Bh	RW(2)
N/A	R(3)W(4)	*Table 34: LOCK_CCFILE*	Blocks 0 and 1 RF Write protection	E2=1	000Ch	RW(2)
0Dh	RW(1)	*Table 12: MB_MODE*	Fast Transfer Mode state after power ON	E2=1	000Dh	RW(2)
0Eh	RW(1)	*Table 13: MB_WDG*	Maximum time before the message is automatically released	E2=1	000Eh	RW(2)
0Fh	RW(1)	*Table 35: LOCK_CFG*	Protect RF Write to system configuration registers	E2=1	000Fh	RW(2)
N/A	WO(5)	*Table 43: LOCK_DSFID*	DSFID lock status	E2=1	0010h	RO
NA	WO(6)	*Table 44: LOCK_AFI*	AFI lock status	E2=1	0011h	RO
N/A	RW(5)	*Table 45: DSFID*	DSFID value	E2=1	0012h	RO

Table 9. System configuration memory map (continued)

RF access		Static Register		I^2C access		
Address	**Type**	**Name**	**Function**	**Device select**	**Address**	**Type**
N/A	RW(6)	*Table 46: AFI*	AFI value	E2=1	0013h	RO
N/A	RO	*Table 47: MEM_SIZE*	Memory size value in blocks, 2 bytes	E2=1	0014h to 0015h	RO
	RO	*Table 48: BLK_SIZE*	Block size value in bytes	E2=1	0016h	RO
N/A	RO	*Table 49: IC_REF*	IC reference value	E2=1	0017h	RO
NA	RO	*Table 50: UID*	Unique identifier, 8 bytes	E2=1	0018h to 001Fh	RO
No access		*Table 51: IC_REV*	IC revision	E2=1	0020h	RO
		-	ST Reserved	E2=1	0021h	RO
		-	ST Reserved	E2=1	0022h	RO
		-	ST Reserved	E2=1	0023h	RO
		Table 36: I2C_PWD	I^2C security session password, 8 bytes	E2=1	0900h to 0907h	R(7)/ W(8)
N/A	WO(9)	*Table 37: RF_PWD_0*	RF configuration security session password, 8 bytes	No access		
N/A	WO(9)	*Table 38: RF_PWD_1*	RF user security session password 1, 8 bytes			
N/A	WO(9)	*Table 39: RF_PWD_2*	RF user security session password 2, 8 bytes			
N/A	WO(9)	*Table 40: RF_PWD_3*	RF user security session password 3, 8 bytes			

1. Write access is granted if RF configuration security session is open and configuration is not locked (LOCK_CFG register equals to 0).
2. Write access if I^2C security session is open.
3. LOCK_CCFILE content is only readable through reading the Block Security Status of blocks 00h and 001h (see *Section 5.6.3: User memory protection*)
4. Write access to bit 0 if Block 00h is not already locked and to bit 1 if Block 01h is not already locked.
5. Write access if DSFID is not locked
6. Write access if AFI is not locked.
7. Read access is granted if I^2C security session is open.
8. Write access with I^2C Write Password command, only after presenting a correct I^2C password.
9. Write access only if corresponding RF security session is open.

4.4 Dynamic configuration

ST25DV has a set of dynamic registers that allow temporary modification of its behavior or report on its activity. Dynamic registers are volatile and not restored to their previous values after POR.

Some static registers have an image in dynamic registers: dynamic register value is initialized with the static register value and may be updated by the application to modify the device behavior temporarily (i.e.: set reset of Energy Harvesting). When a valid change occurs in a static register, in RF or I^2C, the corresponding dynamic register is automatically updated.

Other, dynamic registers, automatically updated, contain indication on ST25DV activity. (i.e.: IT_STS_Dyn gives the interruption's status or MB_CTRL_Dyn gives the Fast Transfer Mode mailbox control).

In RF, dynamic registers can be accessed via dedicated (Fast) Read Dynamic Configuration and (Fast) Write Dynamic Configuration commands, with a pointer acting as the register address. No password is needed to access dynamic registers.

In I^2C, dynamic registers can be accessed with I^2C read and write commands with device select E2=0. Dynamic registers can be read in continuity. Dynamic registers and Fast transfer Mode mailbox can be read in continuity, but not written in continuity. No password is needed to access dynamic registers.

Table 10 shows the complete map of dynamic registers, as seen by RF interface and by I^2C interface.

Table 10. Dynamic registers memory map

RF access		Dynamic Registers		I^2C access		
Address	**Type**	**Name**	**Function**	**Device select**	**Address**	**Type**
00h	RO	*Table 21: GPO_CTRL_Dyn*	GPO control	E2 = 0	2000h	R/W
No access		-	ST Reserved	E2 = 0	2001h	RO
02h	R/W	*Table 25: EH_CTRL_Dyn*	Energy Harvesting management & usage status	E2 = 0	2002h	R/W
No access		*Table 28: RF_MNGT_Dyn*	RF interface usage management	E2 = 0	2003h	R/W
		Table 41: I2C_SSO_Dyn	I^2C security session status	E2 = 0	2004h	RO
		Table 22: IT_STS_Dyn	Interruptions Status	E2 = 0	2005h	RO
0Dh	R/W	*Table 14: MB_CTRL_Dyn*	Fast Transfer Mode control and status	E2 = 0	2006h	R/W
NA	RO	*Table 15: MB_LEN_Dyn*	length of Fast Transfer Mode message	E2 = 0	2007h	RO

4.5 Fast Transfer Mode mailbox

ST25DVxxx Fast Transfer Mode uses a dedicated mailbox buffer for transferring messages between RF and I^2C worlds. This mailbox contains up to 256 Bytes of data which are filled from the first byte.

Fast Transfer Mode mailbox is accessed in bytes from both RF and I^2C.

In RF, mailbox is read via a dedicated (Fast) Read Message command. Read can start from any address value inside the mailbox, between 00h and FFh. Writing in the mailbox is done via the (Fast) Write Message command in one shot, always starting at mailbox address 00h. No password is needed to access mailbox from RF, but Fast Transfer Mode must be enabled.

In I^2C, mailbox read can start from any address value between 2008h and 2107h. Write mailbox MUST start from address 2008h to a max of 2107h. No password is needed to access mailbox from I^2C, but Fast Transfer Mode must be enabled.

Table 11 shows the map of fast transfer mode mailbox, as seen by RF interface and by I^2C interface.

Table 11. 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

5 ST25DVxxx specific features

ST25DVxxx offers the following features:

- A Fast Transfer Mode (FTM), to achieve a fast link between RF and contact worlds, via a 256 byte buffer called Mailbox. This mailbox dynamic buffer of 256 byte can be filled or emptied via either RF or I^2C.
- A GPO pin, which indicates incoming event to the contact side, like RF Field changes, RF activity in progress, RF writing completion or Mailbox message availability.
- An Energy Harvesting element to deliver µW of power when external conditions make it possible.
- RF management, which allows ST25DVxxx to ignore RF requests.

All these features can be programmed by setting static and/or dynamic registers of the ST25DVxxx. ST25DVxxx can be partially customized using configuration registers located in the E^2 system area.

These registers are:

- dedicated to Data Memory organization and protection $ENDA_i$, I2CSS, RFAiSS, LOCK_CCFILE.
- dedicated to Fast Transfer Mode MB_WDG, MB_MODE
- dedicated to observation, GPO, IT_TIME
- dedicated to RF , RF_MNGT, EH_MODE
- dedicated the device's structure LOCK_CFG

A set of additional registers allows to identify and customize the product (DSFID, AFI, IC_REF, etc.).

In I²C,

Read accesses to the static configuration register is always allowed, except for passwords. For dedicated registers, write access is granted after prior successful presentation of the I^2C password. Configuration register are located from @00h to 0Fh in the system area (device code 111)

In RF

Dedicated commands Read Configuration and Write Configuration must be used to access the static configuration registers. Update is only possible when the access right was granted by presenting the RF configuration password (RF_PWD_0), and if the system configuration was not previously locked by the I^2C host (LOCK_CFG=1), which acts as security master.

After any valid write access to the static configuration registers, the new configuration is immediately applied.

Some of the static registers have a dynamic image (notice _Dyn) preset with the static register value: GPO_CTRL_Dyn, EH_CTRL_Dyn, RF_MNGT_Dyn and MB_CTRL_Dyn.

When it exists, ST25DVxxx uses the dynamic configuration register to manage its processes. A dynamic configuration register updated by the application will recover its default static value after a Power On Reset (POR).

Other dynamic registers are dedicated to process monitoring:

- I2C_SSO_Dyn is dedicated to data memory protection
- MB_LEN_Dyn, MB_CTRL_Dyn are dedicated to Fast Transfer Mode
- IT_STS_Dyn is dedicated to interrupt

In I^2C, read and write of the Dynamic registers is done using usual I^2C read & write command at dedicated address. (E2 =0 in device select).

In RF read or write accesses to the Dynamic registers are associated to the dedicated commands, Read Dynamic Configuration, Write Dynamic Configuration and Read Message Length.

5.1 Fast transfer mode (FTM)

5.1.1 Fast Transfer Mode registers

Static Registers

Table 12. MB_MODE(1)

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**	
Bit	**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

1. Refer to *Table 9: System configuration memory map* for the MB_MODE register.

Table 13. MB_WDG(1)

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**	
Bit	**Name**	**Function**	**Factory Value**
b2-b0	MB_WDG	Watch dog duration = $2^{(MB_WDG - 1)} \times 30ms \pm 6$ If MB_WDG = 0, then watchdog duration is infinite	111b
b7-b3	RFU	-	00000b

1. Refer to *Table 9: System configuration memory map* for the MB_WDG register.

Dynamic Registers

Table 14. MB_CTRL_Dyn(1)

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(2)	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 message in FTM mailbox	0b

Table 14. MB_CTRL_Dyn(1) (continued)

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**	
I²C	**Address**	**E2 = 0, 2006h**	
	Type	**b0: R always, W - b7 - b1: RO**	
Bit	**Name**	**Function**	**Factory Value**
b3	RFU	-	0b
b4	HOST_MISS_MSG	0: No message missed by I²C 1: I²C 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 I²C 1: Current Message in FTM mailbox comes from I²C	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. Refer to *Table 10: Dynamic registers memory map* for the MB_CTRL_Dyn register.
2. MB_EN bit is automatically reset to 0 if MB_MODE register is reset to 0.

Table 15. MB_LEN_Dyn(1)

RF	**Command**	**Read Message Lenght (cmd code ABh) Fast Read Message Lenght (cmd code CBh)**	
	Type	**RO**	
I²C	**Address**	**E2 = 0, 2007h**	
	Type	**RO**	
Bit	**Name**	**Function**	**Factory Value**
b7-b0	MB_LEN	Size in byte of message contained in FTM mailbox (automatically set by ST25DVxxx)	0h

1. Refer to *Table 10: Dynamic registers memory map* for the MB_LEN_Dyn register.

5.1.2 Fast Transfer Mode usage

ST25DV acts as mailbox between RF (reader, smartphone, ...) and an I²C host (microcontroller...). Each interface can send a message containing up to 256 bytes of data to the other interface through that mailbox.

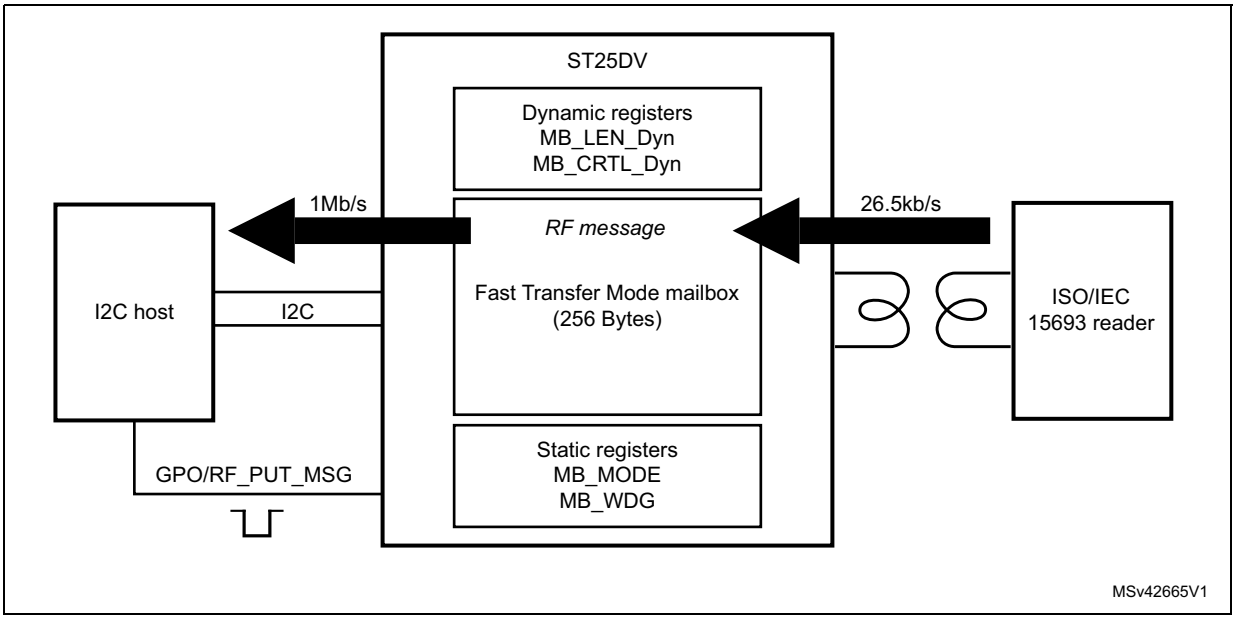
To send data from RF reader to I²C host, the RF user first writes the message containing data in the mailbox.

I^{2}C host is then informed (by interruption on GPO output or polling on MB_CTRL_Dyn register) that a message from RF is present in the mailbox.

Once read, mailbox is cleared and RF user can write the next data message.

The RF user is informed that the message has been read by the I^{2}C host by polling on MB_CTRL_Dyn register.

Figure 8. RF to I^{2}C fast transfer mode operation



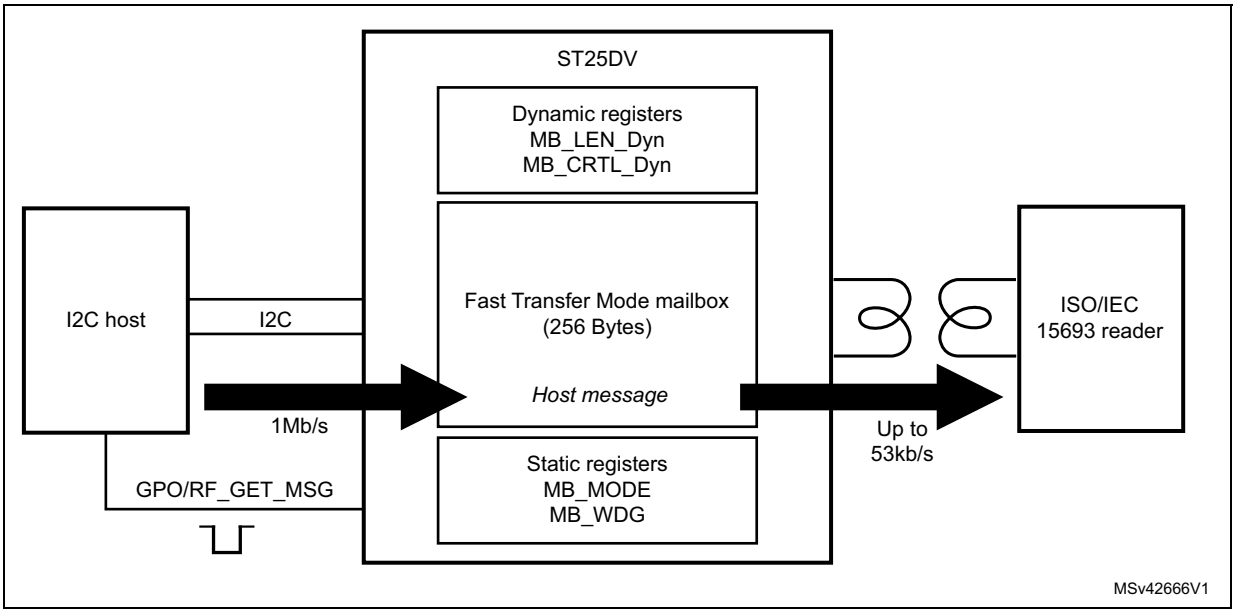
To send data from the I^{2}C host to the RF reader, the I^{2}C host first writes the message containing data in the mailbox.

The RF user must poll on MB_CTRL_Dyn register to check for the presence of a message from I2C in the mailbox.

Once read, the mailbox is cleared and the I^{2}C host can write the next data message.

The I^{2}C host is informed that message has been read by RF user through a GPO interruption or by polling on the MB_CTRL_Dyn register.

Figure 9. I²C to RF fast transfer mode operation



V_{CC} supply source is mandatory to activate this feature.

No precedence rule is applied: the first request is served first.

Adding a message is only possible when Fast Transfer Mode is enabled (MB_EN=1) and mailbox is free (HOST_PUT_MSG and RF_PUT_MSG cleared, which is the case after POR or after complete reading of I²C message by RF, and complete reading of RF message by I²C).

A watchdog limits the message availability in time: when a time-out occurs, the mailbox is considered free, and the HOST_MISS_MSG or RF_MISS_MSG bits is set into MB_CTRL_Dyn register. The message is stored in a buffer (256 Bytes), and the write operation is done immediately.

Caution: The data written in user or system memory (EEPROM), either from I²C or from RF, transits via the 256-Bytes Fast Transfer Mode's buffer. Consequently Fast Transfer Mode must be deactivated (MB_EN=0) before starting any write operation in user or system memory, otherwise command will be NotACK for I²C or get an answer 0Fh for RF and programming is not done.

I²C access to mailbox

The access by I²C can be done by dedicated address mapping to mailbox (2008h to 2107h) with device identifier E2 = 0.

I²C reading operation does not support rollover. Therefore data out is set to FFh when the counter reaches the message end.

The RF_PUT_MSG is cleared after reaching the STOP consecutive to reading the last message byte.

A I²C reading operation will never clear HOST_PUT_MSG, and the message remains available for RF.

A I^2C write operation must start from the first mailbox location, at address 2008h. After reaching the Mailbox border at address 2107h all bytes are NACK and the command is not executed (rollover feature not supported).

At the end of a successful message write, the message length is automatically set into MB_LEN_Dyn register, and HOST_PUT_MSG bit is set into MB_CTRL_Dyn register.

RF access to mailbox

The RF Control & Access to mailbox is possible using dedicated custom commands:

- Read Dynamic Configuration and Fast Read Dynamic Configuration to check availability of mailbox.
- Write Dynamic Configuration and Fast Write Dynamic configuration to enable or disable Fast Transfer Mode.
- Read Message Length and Fast Read Message Length to get the length of the contained message,
- Read Message and Fast Read Message to download the content of the mailbox,
- Write Message and Fast Write Message to put a new message in mailbox. (New length is automatically updated after completion of a successful Write Message or Fast Write Message command).

HOST_PUT_MSG is cleared following a valid reading of the last message byte.

A RF reading operation will never clear RF_PUT_MSG , the message will remain available for I^2C.

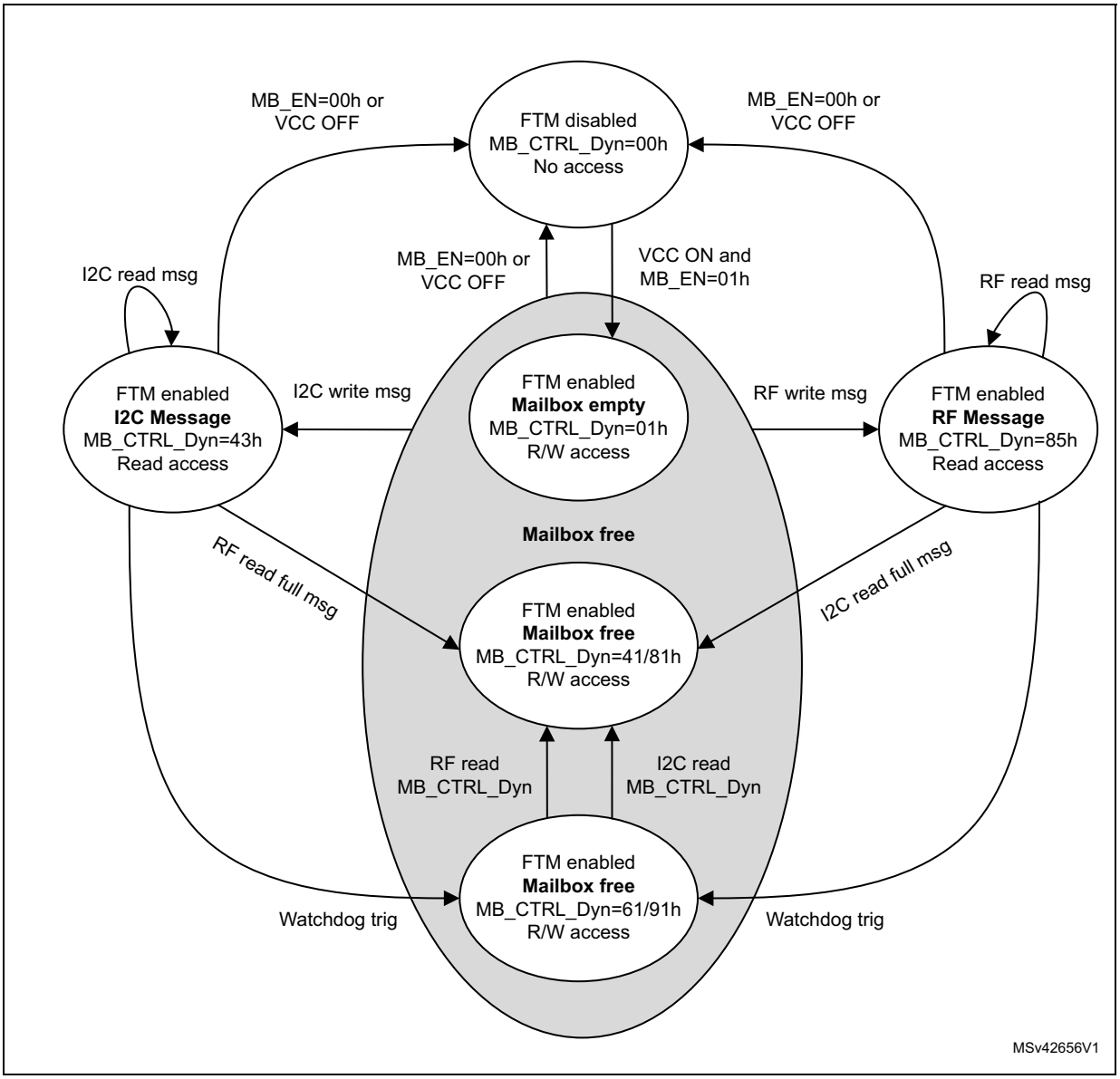
To get more details about sequences to prepare and initiate a Fast Transfer, to detect progress of a fast transfer or to control and execute a fast transfer, please refer to AN4910(

How to exchange data between wired (I^2C) and wireless world (RF ISO15693) using fast transfer mode supported by ST25DVxxx).

At the end of a successful message write, the message length is automatically set in MB_LEN_Dyn register, and RF_PUT_MSG bit is set in MB_CTRL_Dyn register.

the presence of a DC supply is mandatory to get RF access to the mailbox. VCC_ON can be checked reading the dynamic register EH_CTRL_Dyn.

Figure 10. Fast Transfer Mode mailbox access management.



Note: Assuming MB_MODE=01h

Assuming no error occurred

5.2 GPO

GPO signal is used to alert the I^2C host of external RF events or ST25DVxxx processes activity. Several causes could be used to request a host interruption. RF user can also directly drive GPO pin level using a dedicated RF command.

5.2.1 ST25DVxxx interrupt capabilities on RF events

ST25DVxxx supports multi interruption mode and can report several events occurring through RF interface.

In this chapter, all drawings are referring to the Open Drain version of GPO output (ST25DVxxK-IE).

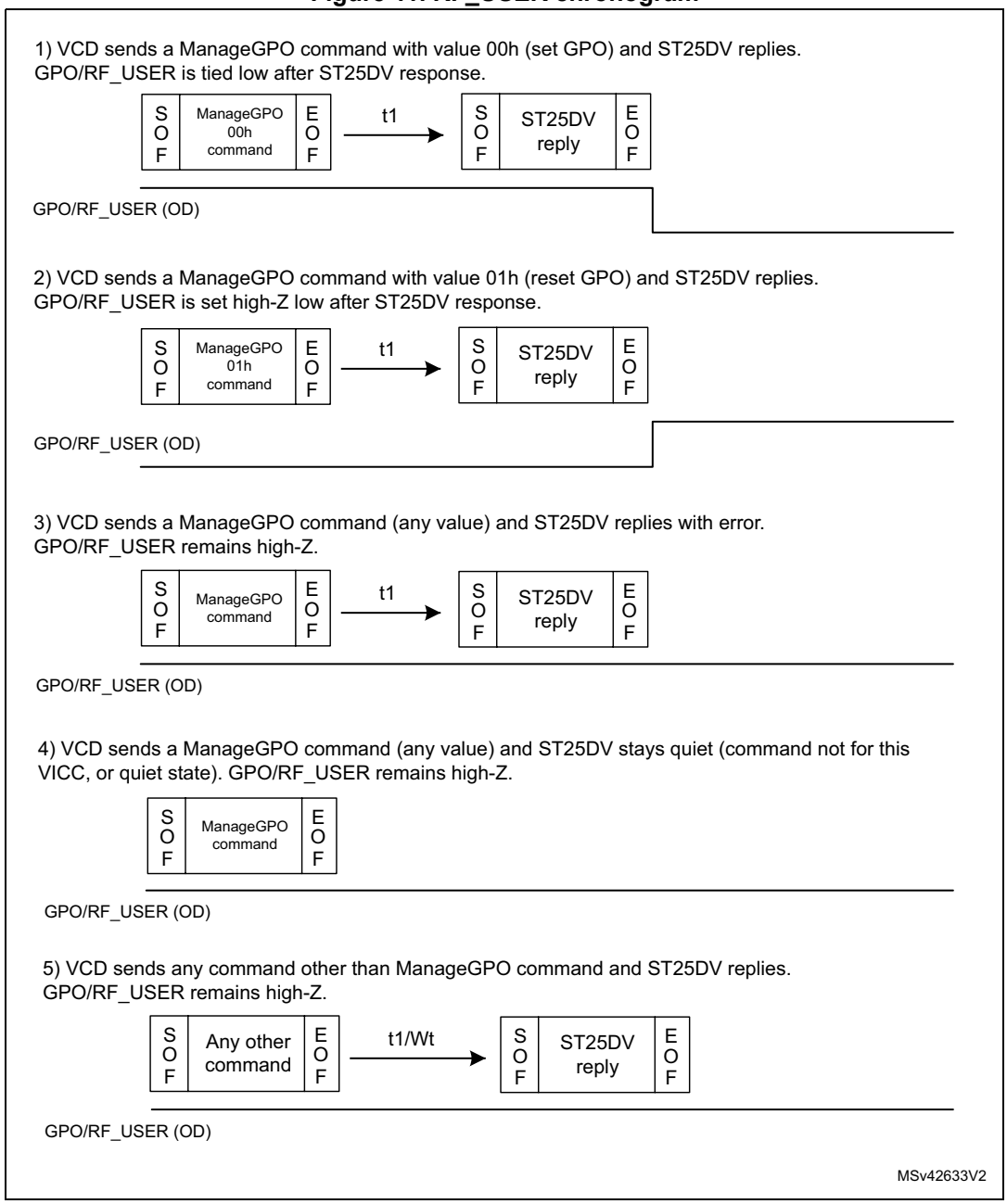
The reader can retrieve the behavior of CMOS version (ST25DVxxK-JF) by inverting the GPO curve polarity and replace in text “released” or “high-Z” by “pull to ground”.

Supported RF events is listed hereafter:

RF_USER:

- GPO output level is controlled by Manage GPO command (set or reset)
- When RF_USER is activated, GPO level is changed after EOF of ST25DV response to a Manage GPO set or reset command (see *Section 7.6.30: Manage GPO*).
- RF_USER is prevalent over all other GPO events when set by Manage GPO command (other interrupts are still visible in IT_STS_Dyn status register, but do not change GPO output level).

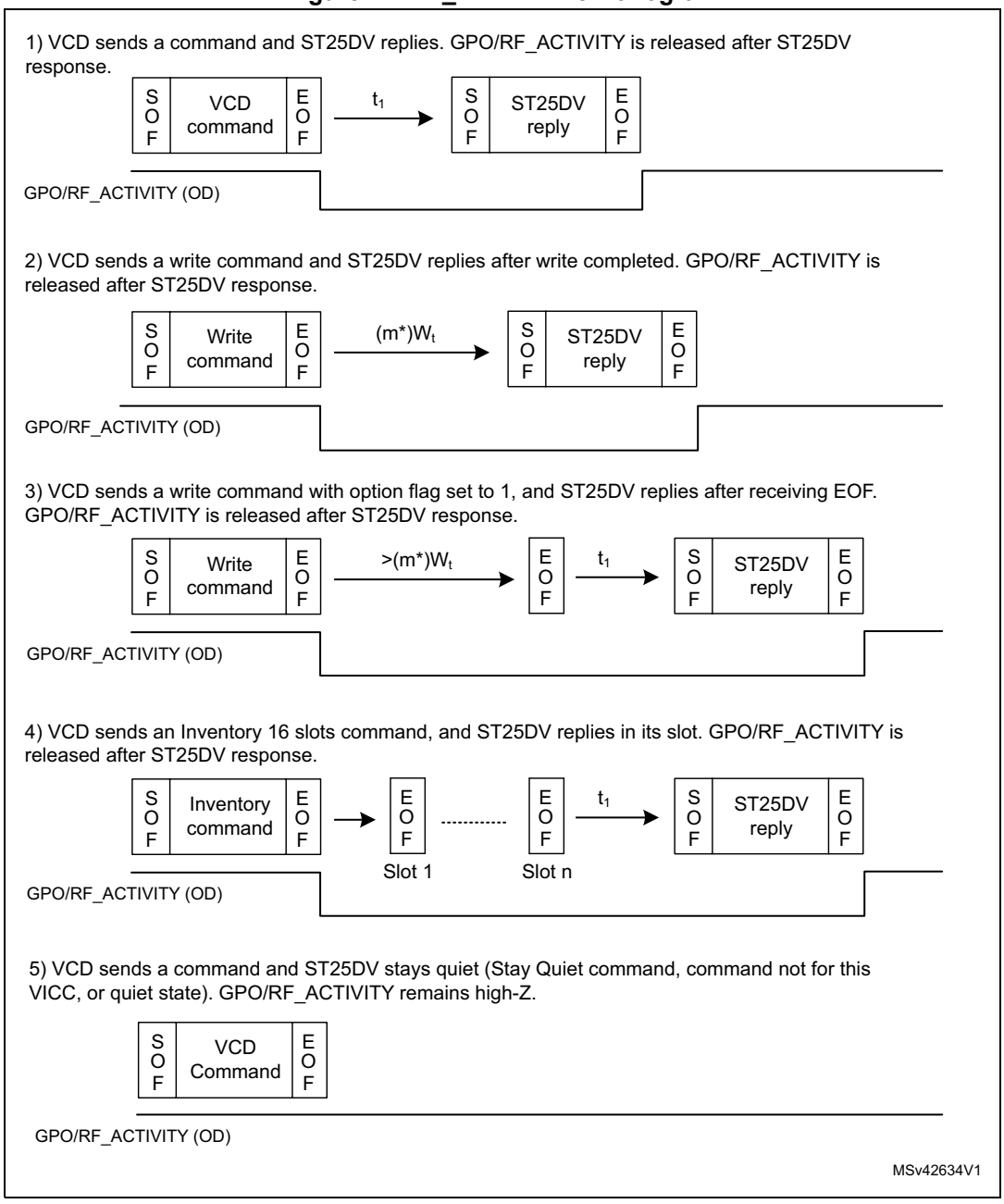
Figure 11. RF_USER chronogram



RF_ACTIVITY:

- GPO output level reflects the RF activity.
- When RF_ACTIVITY is activated, a GPO output level change from RF command EOF to ST25DV response EOF.

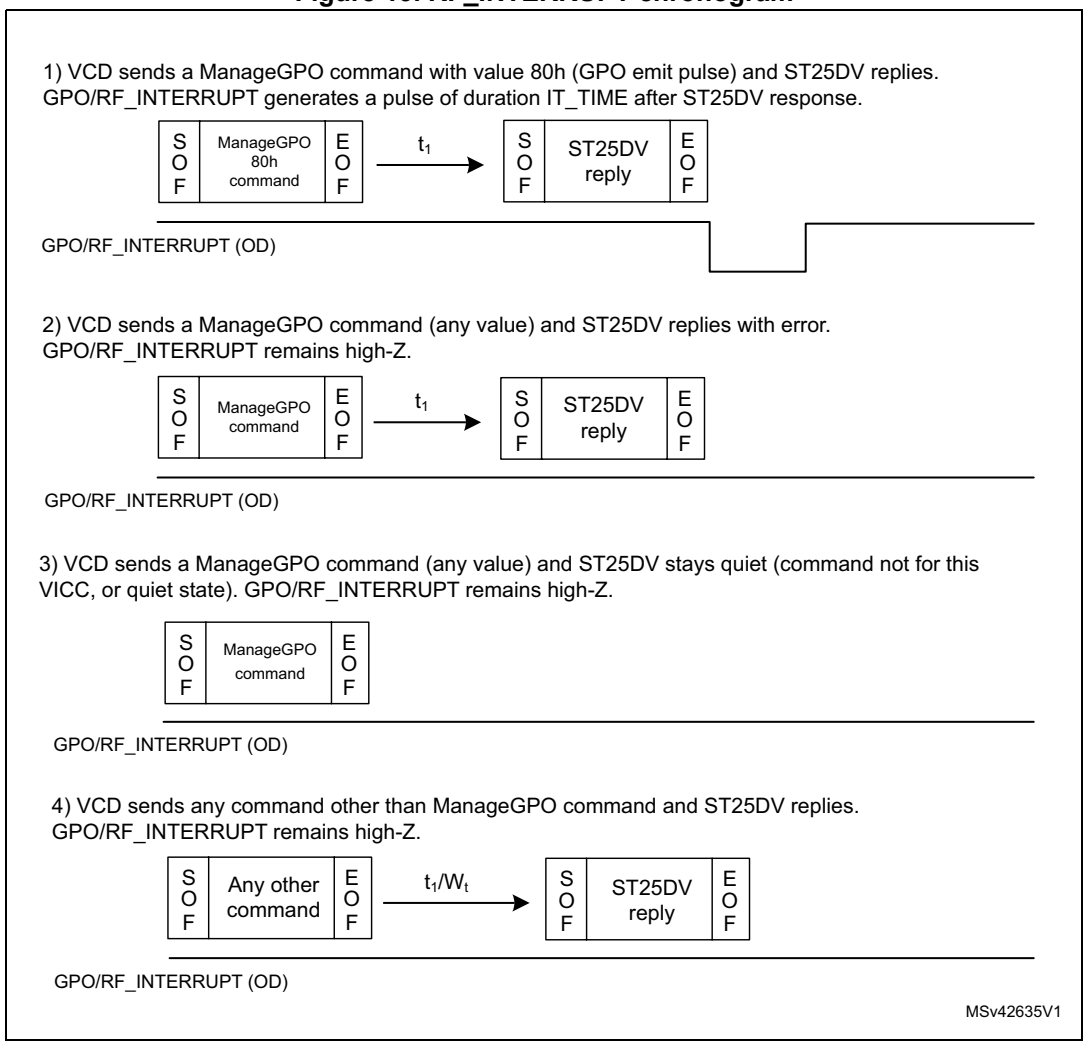
Figure 12. RF_ACTIVITY chronogram



RF_INTERRUPT:

- A pulse is emitted on GPO by Manage GPO command (interrupt).
- When RF_INTERRUPT is activated, a pulse of duration IT_TIME is emitted after EOF of ST25DV response to a Manage GPO interrupt command (see *Section 7.6.30: Manage GPO*).

Figure 13. RF_INTERRUPT chronogram



FIELD_CHANGE:

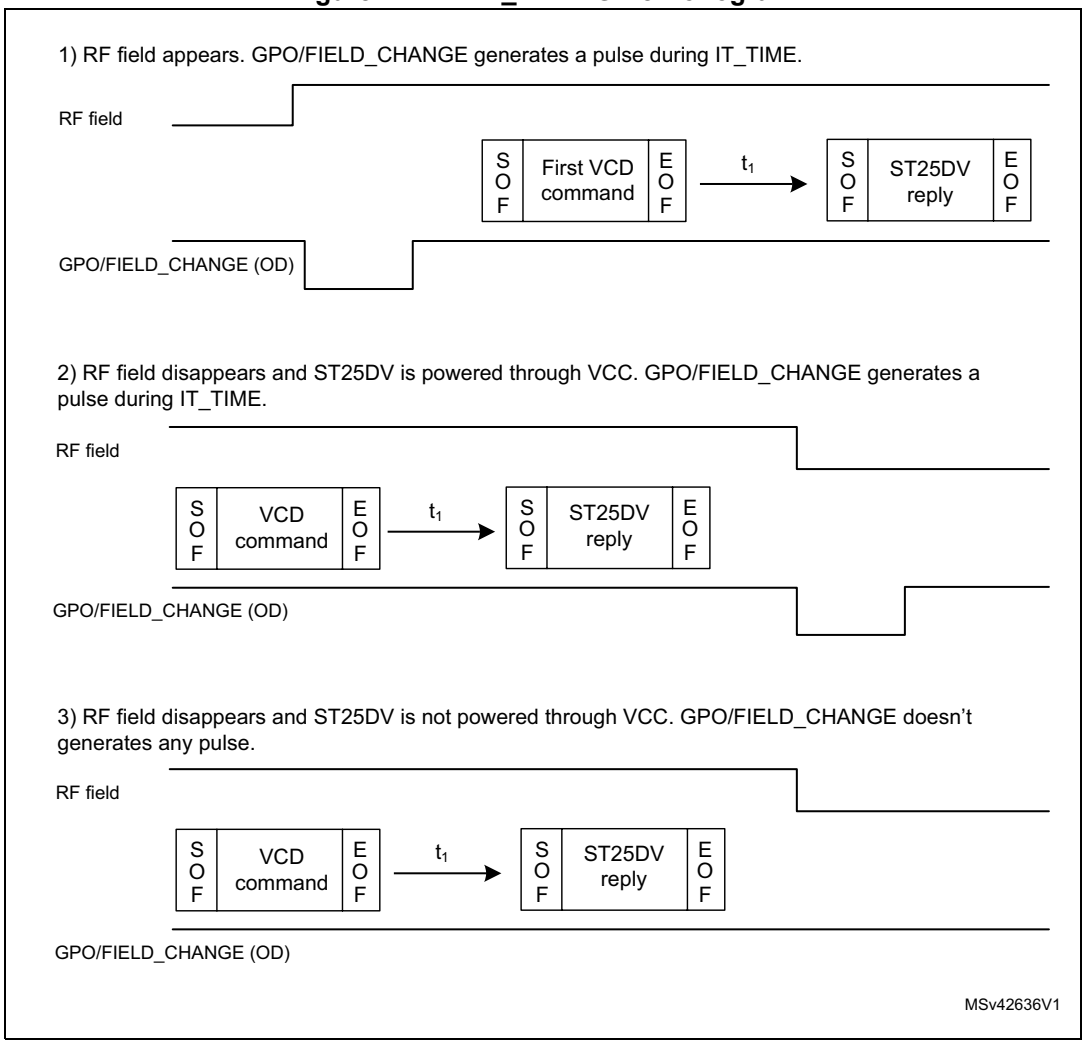
- A pulse is emitted on GPO to signal a change in RF field state.
- When FIELD_CHANGE is activated, and when RF field appear or disappear, GPO emits a pulse of duration IT_TIME.
- In case of RF field disappear, the pulse is emitted only if V_{CC} power supply is present.
- If RF is configured in RF_SLEEP mode, field change are not reported on GPO, even if FIELD_CHANGE event is activated, as shown in *Table 16*.

Table 16. FIELD_CHANGE when RF is disabled or in sleep mode

RF_DISABLE	RF_SLEEP	GPO behavior when FIELD_DETECT is enabled
0	0	A pulse is emitted on GPO if RF field appears or disappears(1)
1	0	
X	1	GPO remains High-Z (OD) or tied low (CMOS) IT_STS_Dyn register is not updated.
X	1	

1. assuming that GPO output is enabled (GPO_EN = 1).

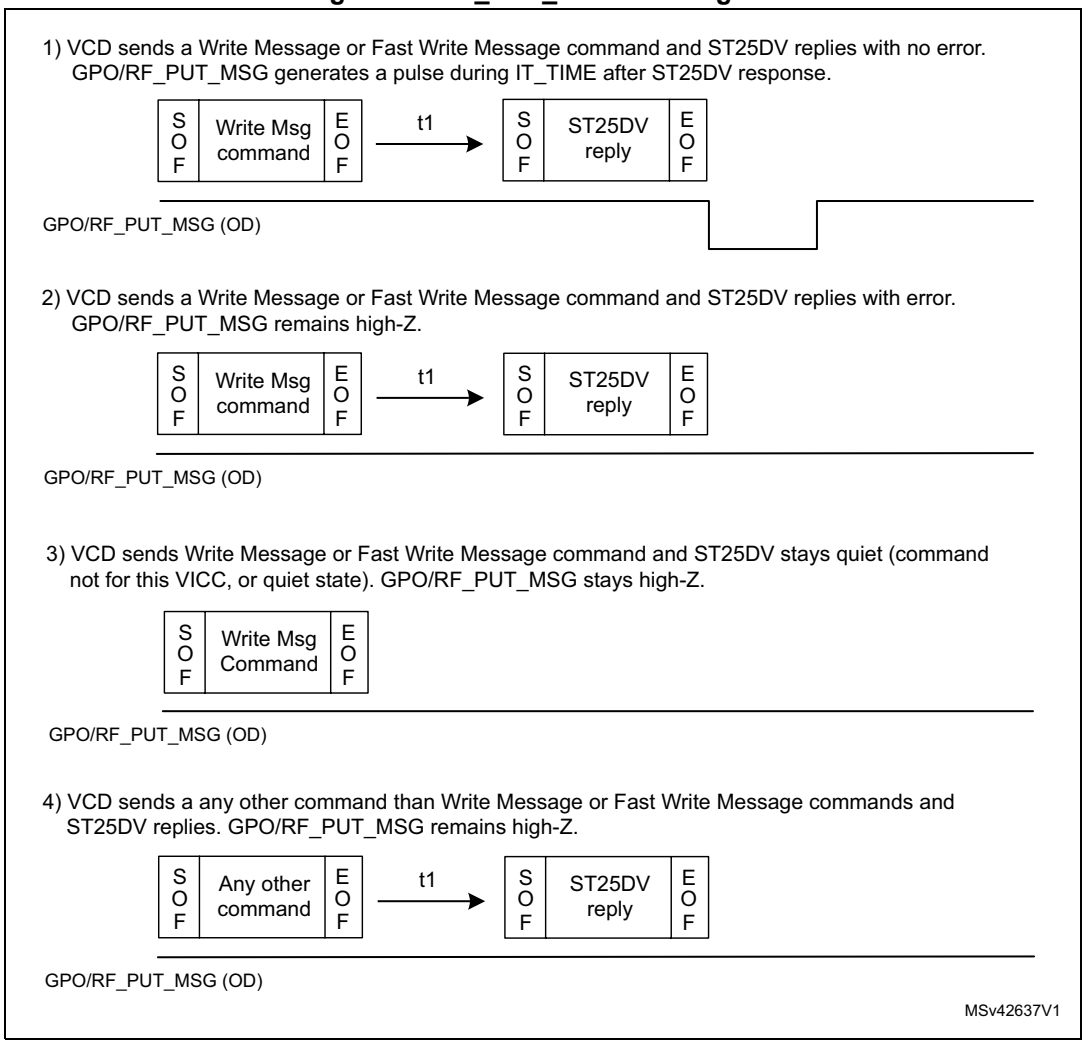
Figure 14. FIELD_CHANGE chronogram



RF_PUT_MSG:

- A pulse is emitted on GPO when a message is successfully written by RF in Fast transfer mode mailbox.
- When RF_PUT_MSG is activated, a pulse of duration IT_TIME is emitted on GPO at completion of valid Write Message or Fast Write Message commands (after EOF of ST25DV response).

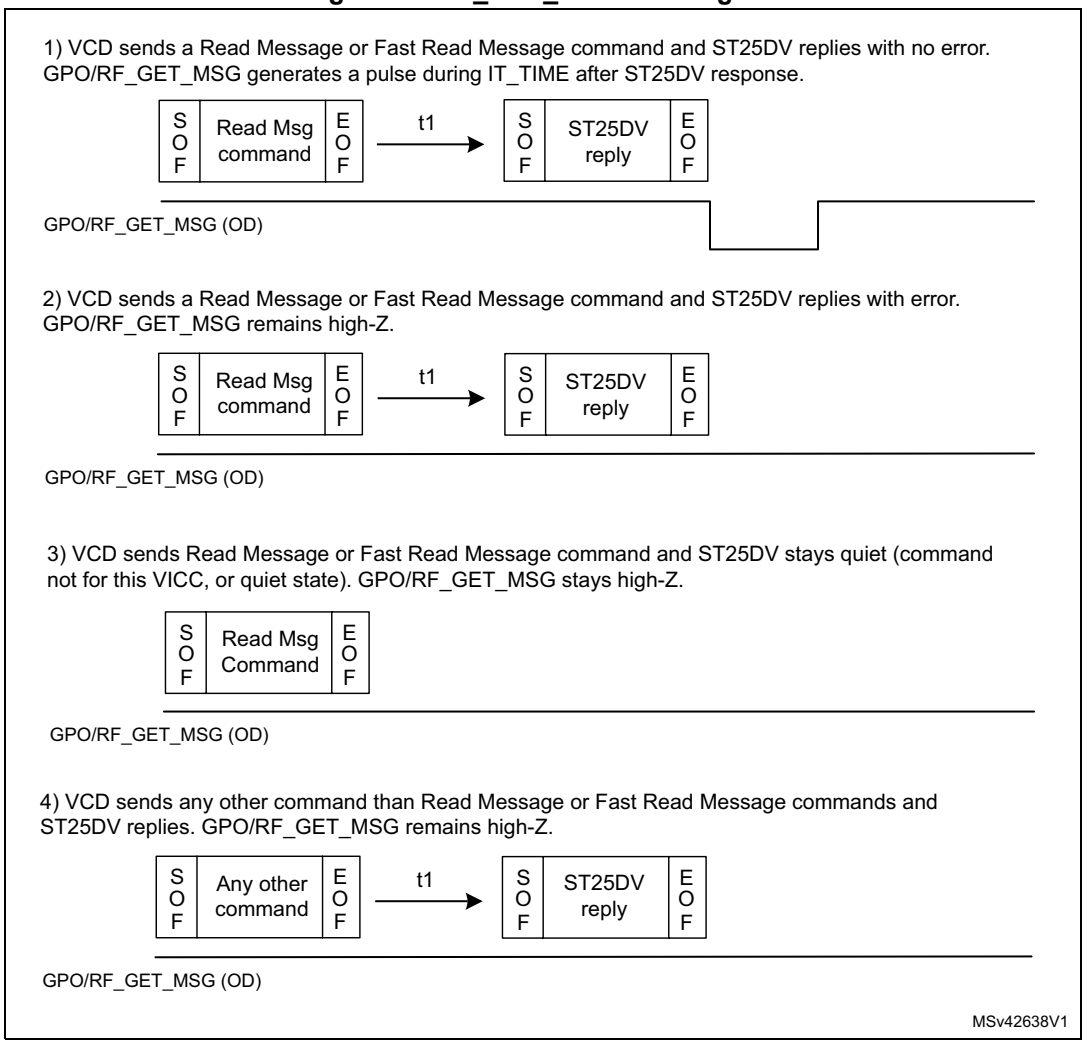
Figure 15. RF_PUT_MSG chronogram



RF_GET_MSG:

- A pulse is emitted on GPO when RF has successfully read a message, up to its last byte, in Fast transfer mode mailbox.
- When RF_GET_MSG is activated, a pulse of duration IT_TIME is emitted on GPO at completion of valid Read Message or Fast Read Message commands (after EOF of ST25DV response), and end of message has been reached.

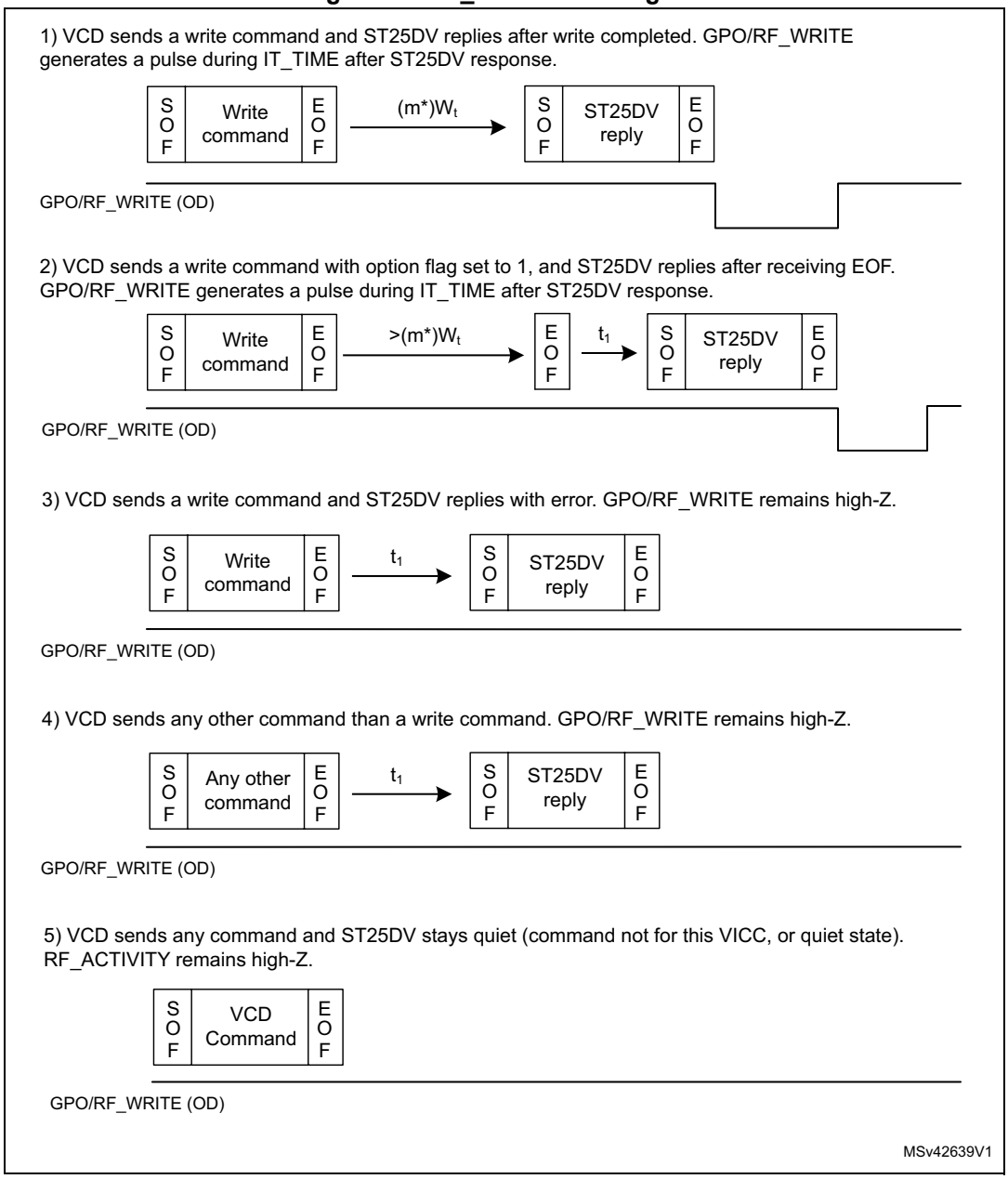
Figure 16. RF_GET_MSG chronogram



RF_WRITE:

- When RF_WRITE is activated, a pulse of duration IT_TIME is emitted at completion of a valid RF write operation in EEPROM (after EOF of ST25DV response).
- Following commands trigger the RF_WRITE interrupt after a valid write operation in EEPROM:
  - Write Single Block
  - Extended Write Single Block
  - Write Multiple Block
  - Extended Write Multiple Block
  - Lock Block
  - Extended Lock Block
  - Write AFI
  - Lock AFI
  - Write DSFID
  - Lock DSFID
  - Write Configuration
  - Write Password
- Note that writing in dynamic registers or Fast transfer mode mailbox does not trigger RF_WRITE interrupt (no write operation in EEPROM).

Figure 17. RF_WRITE chronogram



5.2.2 GPO and power supply

When at the same time RF field is present and V_{CC} is ON, GPO is acting as configured in GPO, GPO_CTRL_Dyn and IT_TIME registers.

When the RF field disappears, the GPO state is reset and the output level is set to high-Z (Open Drain) or tied low (CMOS). Interruption status in IT_STS_Dyn register is maintained until next I^2C read or V_{CC} power off.

Table 17. GPO interrupt capabilities in function of RF field

RF field on	RF field off
GPO state is function of RF events(1)	GPO remains High-Z (OD) or tied low (CMOS)

1. If pull-up resistor is powered (Open Drain-IE version), and V_{DCG} is powered (CMOS –JF version).

When V_{CC} is not present, or ST25DVxxx is in low power mode, all events are available on GPO pin, assuming pull-up resistor is supplied with correct voltage (Open Drain-IE version) or V_{DCG} is powered (CMOS-JF version). Host can be waken up using GPO interrupt in any power condition.

Exception is FIELD_CHANGE when RF field is falling, which can't be reported on GPO output if V_{CC} is off (no power supply on ST25DVxxx)

Table 18. GPO interrupt capabilities in function of V_{CC} power supply

GPO events	V_{CC} OFF	V_{CC} ON and LPD high(1) (low power mode)	V_{CC} ON and LPD low(1)
FIELD_CHANGE if RF field disappears	GPO remains High-Z (OD) or tied low (CMOS)	Pulse emitted on GPO(2)	Pulse emitted on GPO
Any other activated RF event	GPO state is function of RF events(2)	GPO state is function of RF events(2)	GPO state is function of RF events(2)

1. For STM25DVxxK-JF only.
2. If pull-up resistor is powered (Open Drain-IE version) and V_{DCG} is powered (CMOS-JF version).

5.2.3 GPO registers

Four registers are dedicated to this feature:

- Two static registers in system configuration
- Two dynamic registers

Table 19. GPO(1)

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**	
I²C	**Address**	**E2=1, 0000h**	
	Type	**R always, W if I²C 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
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

1. Refer to *Table 9: System configuration memory map* for the GPO register.

- Enables the interruption source, and enable GPO output.
- Several interruption sources can be enabled simultaneously.
- The updated value is valid for the next command (except for the RF_WRITE interrupt, which is valid right after EOF of the Write Configuration command if enabled through RF).
- The GPO_EN bit (b7) allows to disable GPO output (High-Z for Open Drain version, driven low for CMOS version). Interruptions are still reported in IT_STS_Dyn register.
- RF configuration security session (present RF password 0) or I²C security session (present I²C password) must be open in order to write the GPO register.

Table 20. IT_TIME(1)

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**	
I²C	**Address**	**E2=1, 0001h**	
	Type	**R always, W if I²C security session is open**	
Bit	**Name**	**Function**	**Factory Value**
b2-b0	IT_TIME	Pulse duration = 301 us - IT_TIME x 37.65 us ± 2 us	011b
b7-b3	RFU	-	00000b

1. Refer to *Table 9: System configuration memory map* for the IT_TIME register.

- Defines interrupt pulse duration on GPO pin for the flowing events: RF_INTERRUPT, FIELD_CHANGE, RF_PUT_MSG, RF_GET_MSG and RF_WRITE.
- See *IT pulse duration equation:* for interrupt duration calculation.
- RF configuration security session (present RF password 0) or I²C security session (present I²C password) must be open in order to write IT_TIME register.

Table 21. GPO_CTRL_Dyn(1)

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

Table 21. GPO_CTRL_Dyn(1) (continued)

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**
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

1. Refer to *Table 10: Dynamic registers memory map* for the GPO_CTRL_Dyn register.

- Allows I^2C host to dynamically enable or disable GPO output by writing in GPO_EN bit (b7).
- GPO_EN bit of GPO_CTRL_Dyn register is prevalent over GPO_EN bit of GPO register.
- At power up, and each time GPO register is updated, GPO_CTRL_Dyn content is copied from GPO register.
- GPO_CTRL_Dyn is a volatile register. Value is maintained only if at least one of the two power sources is present (RF field or V_{CC}).
- GPO_CTRL_Dyn bit 7 (GPO_EN) can be written even if I^2C security session is closed (I^2C password not presented) but is read only for RF user.
- Modifying GPO_CTRL_Dyn, the bit 7 GPO_EN does not affect the value of GPO register bit 7 GPO_EN

Table 22. IT_STS_Dyn(1)

RF	Command	No access	
	Type		
I^2C	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

1. Refer to *Table 10: Dynamic registers memory map* for the IT_STS_Dyn register.

- Cumulates all events which generate interruptions. It should be checked by I^2C host to know which event triggered an interrupt on GPO pin.
- When enabled, RF events are reported in IT_STS_Dyn register even if GPO output is disabled though the GPO_EN bit.
- Once read the ITSTS_Dyn register is cleared (set to 00h).
- At power up, IT_STS_Dyn content is cleared (set to 00h).
- IT_STS_Dyn is a volatile register. Value is maintained only if at least one of the two power sources is present (RF field or V_{CC}).

5.2.4 Configuring GPO

GPO and interruption pulse duration can be configured by RF user or by I^2C host. One or more interrupts can be enabled at same time.

RF user can use Read Configuration and Write Configuration commands to set accordingly the GPO and IT_TIME registers, after presenting a valid RF configuration password to open RF configuration security session.

I²C host can write GPO and IT_TIME registers, after presenting a valid I²C password to open I²C security session.

Enabling or disabling GPO output:

- RF user and I²C host can disable or enable GPO output at power up time by writing in GPO_EN bit 7 of GPO register (if write access is granted).
- I²C host can temporarily enable or disable GPO output at any time by toggling GPO_EN bit 7 of GPO_CTRL_Dyn register. No password is required to write into GPO_CTRL_Dyn register.
- Disabling GPO output by writing in GPO_EN bit (either in GPO or in GPO_CTRL_Dyn registers) does not disable interruption report in IT_STS_Dyn status register.

Table 23. Enabling or disabling GPO interruptions

GPO bit 7: GPO_EN	GPO_CTRL_Dyn bit 7: GPO_EN	GPO output
0	0	GPO remains High-Z (OD) or tied low (CMOS)
1	0	GPO remains High-Z (OD) or tied low (CMOS)
0	1	Activated RF events are reported on GPO output(1)
1	1	Activated RF events are reported on GPO output(1)

1. If pull-up resistor is powered (Open Drain -IE version), and V_{DCG} is powered (CMOS –JF version).

Interruption pulse duration configuration:

- Interrupt pulse duration is configured by writing pulse duration value in IT_TIME register.
- Pulse duration is calculated with the following equation

IT pulse duration equation:

$$\text{IT pulse duration} = 301\mu s - \text{IT_TIME} \times 37.65\mu s \pm 2\mu s$$

5.3 Energy Harvesting (EH)

5.3.1 Energy harvesting registers

Table 24. EH_MODE(1)

RF	**Command**	**Read Configuration (cmd code A0h) @02h** **Write Configuration (cmd code A1h) @02h**	
	Type	**R always, W if RF configuration security session is open and configuration not locked**	
I²C	**Address**	**E2 = 1, 0002h**	
	Type	**R always, W if I²C security session is open**	
Bit	**Name**	**Function**	**Factory Value**
b0	EH_MODE	0: EH forced after boot 1: EH on demand only	1b
b7-b1	RFU	-	0000000b

1. Refer to *Table 9: System configuration memory map* for the EH_MODE register.

Table 25. EH_CTRL_Dyn(1)

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**	
Bit	**Name**	**Function**	**Factory Value**
b0	EH_EN	0: Disable EH feature 1: Enable EH feature	0b
b1	EN_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 ST25DVxxx may communicate in RF	Depending of power source
b3	VCC_ON	0: No DC supply detected on V_{CC} pin or Low Power Down mode is forced (LPD is high) 1: V_{CC} supply is present and Low Power Down mode is not forced (LPD is low)	Depending of power source
b7-b4	RFU	-	0b

1. Refer to *Table 10: Dynamic registers memory map* for the EH_CTRL_Dyn register.

5.3.2 Energy harvesting feature description

The usage of Energy Harvesting element can be defined in configuration register EH_MODE. When the Energy harvesting mode is disabled or the RF field strength is not sufficient, the energy harvesting analog voltage output V_EH is in High-Z state.

EH_MODE Static Register is used to define the Energy Harvesting default strategy after boot.

At boot EH_EN (in EH_CTRL_Dyn register) is set depending EH_MODE value as shown in table below:

Table 26. Energy harvesting at power-up

EH_MODE	EH_EN (at boot)	Energy harvesting at power-up
0	1	EH enabled after boot (when possible)
1	0	EH disabled initially, EH delivered on demand (when possible)

Writing 0 in EH_MODE at any time after boot will automatically set EH_EN bit to 1, and thus activate energy harvesting.

Writing 1 in EH_MODE at any time after boot will not modify EH_EN bit (until next reboot) and thus will not modify energy harvesting current state.

EH_CTRL_Dyn allows to activate or deactivate on the fly the Energy harvesting (EH_EN) and bring information on actual state of EH and state of power supplies :

- EH_ON set reflects the EH_EN bit value
- FIELD_ON is set in presence of a RF field
- VCC_ON is set when Host power supply is on, and low power-down mode is not forced.

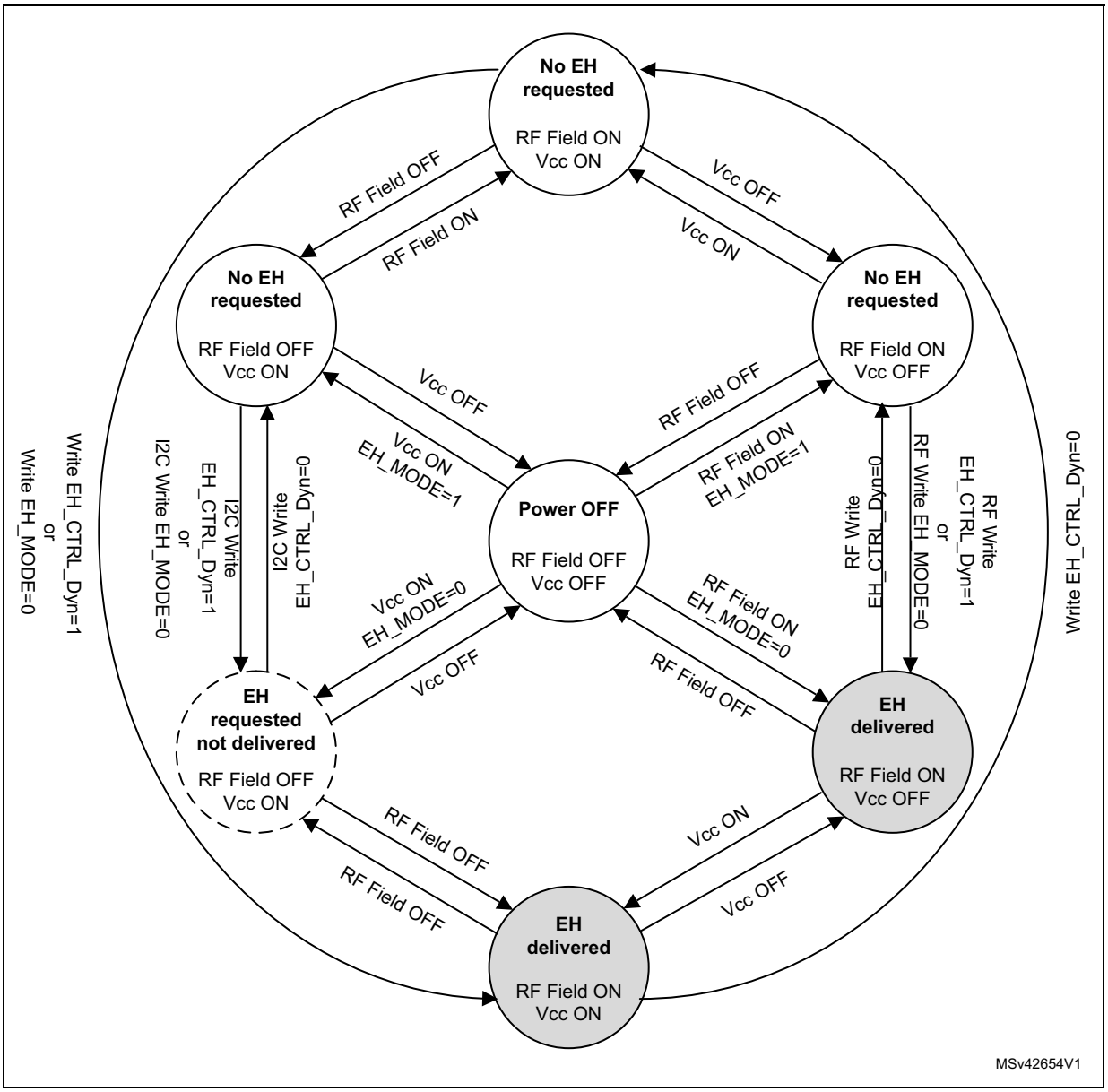
During boot, EH is not delivered to avoid alteration in device configuration.

Caution: Communication is not guaranteed during EH delivery.

Energy harvesting can be set even if ST25DVxxx is in RF disabled or RF Sleep mode, or in Low power mode. In all these cases, ST25DVxxx will deliver power on V_EH pin if RF field is present.

5.3.3 EH delivery state diagram

Figure 18. EH delivery state diagram

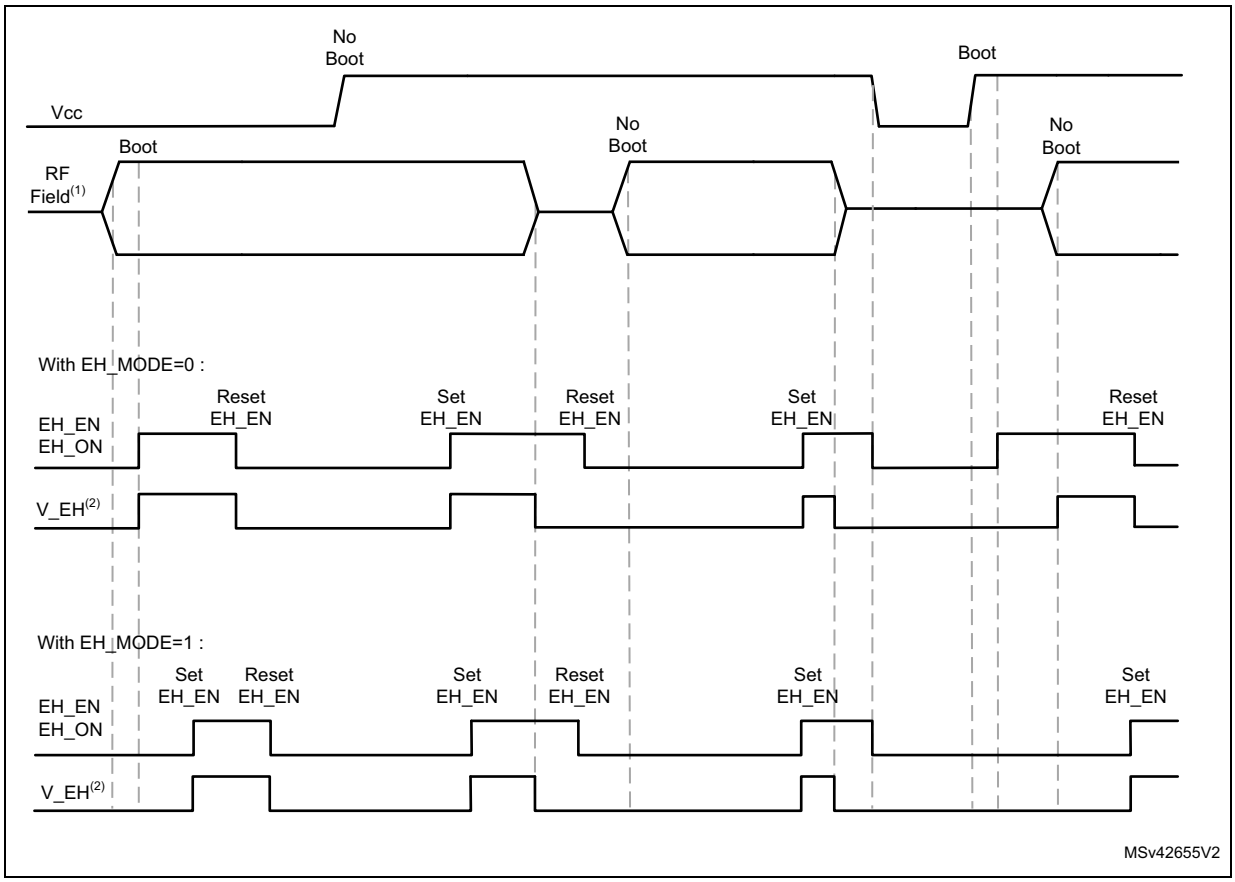


Note: Power is delivered on V_EH only if harvested energy is sufficient to supply ST25DV and leave over power.

Grey color indicates the states where power is delivered on V_EH pin.

5.3.4 EH delivery sequence

Figure 19. ST25DVxxx Energy Harvesting Delivery Sequence



1. We suppose that the captured RF power is sufficient to trig EH delivery.
2. V_EH = 1 means some µW are available on V_EH pin.
   V_EH = 0 means V_EH pin is in high-Z.

5.4 RF management feature

5.4.1 RF management registers

Table 27. RF_MNGT(1)

RF	**Command**	**Read Configuration (cmd code A0h) @03h** **Write Configuration (cmd code A1h) @03h**	
	Type	**R always, W if RF configuration security session is open and configuration not locked**	
I^2C	**Address**	**E2 = 1, 0003h**	
	Type	**R always, W if I^2C security session is open**	
Bit	**Name**	**Function**	**Factory Value**
b0	RF_DISABLE	0: RF commands executed 1: RF commands not executed (error 0Fh returned)	0b
b1	RF_SLEEP	0: RF communication enabled 1: RF communication disabled (ST25DV remains silent)	0b
b7-b2	RFU	-	0000000b

1. Refer to *Table 9: System configuration memory map* for the RF_MNGT register.

Table 28. RF_MNGT_Dyn(1)

RF	**Command**	**No access**	
	Type		
I^2C	**Address**	**E2 = 0, 2003h**	
	Type	**R always, W always**	
Bit	**Name**	**Function**	**Factory Value**
b0	RF_DISABLE	0: RF commands executed 1: RF commands not executed (error 0Fh returned)	0b
b1	RF_SLEEP	0: RF communication enabled 1: RF communication disabled (ST25DV remains silent)	0b
b7-b2	RFU	-	0000000b

1. Refer to *Table 10: Dynamic registers memory map* for the RF_MNGT register.

5.4.2 RF management feature description

RF_MNGT Register is used to control the RF communication between ST25DVxxx and a RF reader.

At boot time, and each time RF_MNGT register it is updated, content of RF_MNGT_Dyn register is copied from RF_MNGT register. The content of RF_MNGT_Dyn register is used during application to set ST25DVxxx behavior.

Content of this dynamic register RF_MNGT_Dyn can be updated on the fly, to temporarily modify the behavior of ST25DVxxx without affecting the static value of RF_MNGT which will be recovered at next POR.

RF_MNGT register is composed of two bits (see *Table 28: RF_MNGT_Dyn*): RF_DISABLE and RF_SLEEP

For a normal usage of RF interface, bits RF_SLEEP and RF_DISABLE must be set to 0.

For RF are offered three modes:

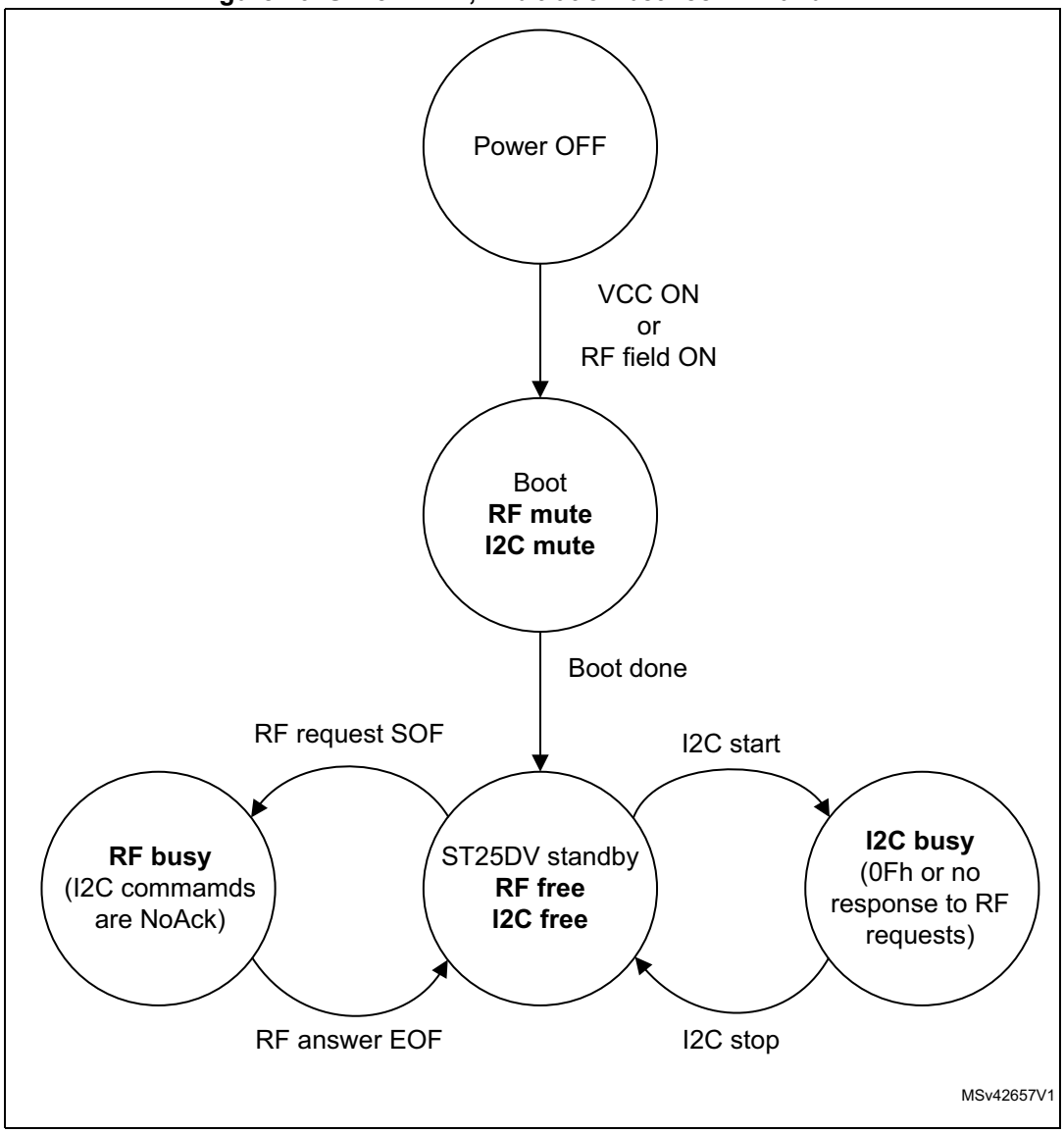
- RF sleep mode:
  - When RF_SLEEP is set to 1, all RF communications are disabled, RF interface doesn't interpret commands, but minimizes consumption of RF interface.
- RF disable mode:
  - When RF_SLEEP is set to 0 and RF_DISABLE is set to 1, RF commands are interpreted but not executed. In case of a valid command, ST25DV will respond after t1 with the error code 0Fh. Inventory and Stay Quiet commands are not answered.
- RF normal mode:
  - In normal usage, RF_SLEEP and RF_DISABLE are set to 0, ST25DVxxx will process the request and respond accordingly when I^2C is not accessing ST25DVxxx. If I^2C is busy, ST25DV will respond to RF request with the error code 0Fh.

Whatever RF_MNGT register value, the Field detection remains available leaving to master the possibility to temporarily open RF communication by overwriting value in RF_MNGT_Dyn dynamic register.

5.5 Interface Arbitration

ST25DVxxx automatically arbitrates the exclusive usage of RF and I^2C interfaces. Arbitration scheme obeys to “first talk first served” basic law. (see *Figure 20*).

Figure 20. ST25DVxxx, Arbitration between RF and I^2C



1. If no response, RF is considered busy from Request SOF to Response EOF or Request EOF.
2. I^2C is considered busy from Start (Select) to Stop (Deselect) or when I^2C timeout occurs (Start_out, Clock low out or Clock High out).

When RF is busy, I^2C interface answers by NoAck on any I^2C command.

When I^2C is busy, RF commands receive no response (Inventory, Stay quiet, addressed commands) or error code 0Fh for any other command.

5.6 Data Protection

ST25DVxxx provides a special data protection mechanism based on passwords that unlock security sessions.

User memory can be protected for read and/or write access and system configuration can be protected from write access, both from RF and I^2C assess.

5.6.1 Data protection registers

Table 29. RFA1SS(1)

RF	**Command**	**Read Configuration (cmd code A0h) @04h** **Write Configuration (cmd code A1h) @04h**	
	Type	**R always, W if RF configuration security session is open and configuration not locked**	
I^2C	**Address**	**E2 = 1, 0004h**	
	Type	**R always, W if I^2C security session is open**	
Bit	**Name**	**Function**	**Factory Value**
b1-b0	PWD_CTRL_A1	00: Area 1 RF user security session can't be open by password 01: Area 1 RF user security session is open by RF_PWD_1 10: Area 1 RF user security session is open by RF_PWD_2 11: Area 1 RF user security session is open by RF_PWD_3	00b
b3-b2	RW_PROTECTION_A1	00: Area 1 RF access: Read always allowed / Write always allowed 01: Area 1 RF access: Read always allowed, Write allowed if RF user security session is open 10: Area 1 RF access: Read always allowed, Write allowed if RF user security session is open 11: Area 1 RF access: Read always allowed, Write always forbidden	00b
b7-b4	RFU	-	0000b

1. Refer to *Table 9: System configuration memory map* for the RFA1SS register.

Table 30. RFA2SS(1)

RF	**Command**	**Read Configuration (cmd code A0h) @06h** **Write Configuration (cmd code A1h) @06h**	
	Type	**R always, W if RF configuration security session is open and configuration not locked**	
I^2C	**Address**	**E2 = 1, 0006h**	
	Type	**R always, W if I^2C security session is open**	
Bit	**Name**	**Function**	**Factory Value**
b1-b0	PWD_CTRL_A2	00: Area 2 RF user security session can't be open by password 01: Area 2 RF user security session is open by RF_PWD_1 10: Area 2 RF user security session is open by RF_PWD_2 11: Area 2 RF user security session is open by RF_PWD_3	00b
b3-b2	RW_PROTECTION_A2	00: Area 2 RF access: Read always allowed, Write always allowed 01: Area 2 RF access: Read always allowed, Write allowed if RF user security session is open 10: Area 2 RF access: Read allowed if RF user security session is open, Write allowed if RF user security session is open 11: Area 2 RF access: Read allowed if RF user security session is open, Write always forbidden	00b
b7-b4	RFU	-	0000b

1. Refer to *Table 9: System configuration memory map* for the RFA2SS register.

Table 31. RFA3SS(1)

RF	**Command**	**Read Configuration (cmd code A0h) @08h** **Write Configuration (cmd code A1h) @08h**	
	Type	**R always, W if RF configuration security session is open and configuration not locked**	
I^2C	**Address**	**E2 = 1, 0008h**	
	Type	**R always, W if I^2C security session is open**	
Bit	**Name**	**Function**	**Factory Value**
b1-b0	PWD_CTRL_A3	00: Area 3 RF user security session can't be open by password 01: Area 3 RF user security session is open by RF_PWD_1 10: Area 3 RF user security session is open by RF_PWD_2 11: Area 3 RF user security session is open by RF_PWD_3	00b
b3-b2	RW_PROTECTION_A3	00: Area 3 RF access: Read always allowed / Write always allowed 01: Area 3 RF access: Read always allowed, Write allowed if RF user security session is open 10: Area 3 RF access: Read allowed if RF user security session is open, Write allowed if RF user security session is open 11: Area 3 RF access: Read allowed if RF user security session is open, Write always forbidden	00b
b7-b4	RFU	-	0000b

1. Refer to *Table 9: System configuration memory map* for the RFA3SS register.

Table 32. RFA4SS[1]

RF	**Command**	**Read Configuration (cmd code A0h) @0Ah** **Write Configuration (cmd code A1h) @0Ah**	
	Type	**R always, W if RF configuration security session is open and configuration not locked**	
I^2C	**Address**	**E2 = 1, 000Ah**	
	Type	**R always, W if I^2C security session is open**	
Bit	**Name**	**Function**	**Factory Value**
b1-b0	PWD_CTRL_A4	00: Area 4RF user security session can't be open by password 01: Area 4 RF user security session is open by RF_PWD_1 10: Area 4 RF user security session is open by RF_PWD_2 11: Area 4 RF user security session is open by RF_PWD_3	00b
b3-b2	RW_PROTECTION_A4	00: Area 4 RF access: Read always allowed, Write always allowed 01: Area 4 RF access: Read always allowed, Write allowed if RF user security session is open 10: Area 4 RF access: Read allowed if RF user security session is open, Write allowed if RF user security session is open 11: Area 4 RF access: Read allowed if RF user security session is open, Write always forbidden	00b
b7-b4	RFU	-	0000b

1. Refer to *Table 9: System configuration memory map* for the RFA4SS register.

Table 33. I2CSS(1)

RF	Command	No access	
	Type		
I^2C	Address	E2 = 1, 000Bh	
	Type	R always, W if I^2C security session is open	
Bit	**Name**	**Function**	**Factory Value**
b1-b0	RW_PROTECTION_A1	00: Area 1 I^2C access: Read always allowed, Write always allowed 01: Area 1 I^2C access: Read always allowed, Write allowed if I^2C user security session is open 10: Area 1 I^2C access: Read always allowed, Write always allowed 11: Area 1 I^2C access: Read always allowed, Write allowed if I^2C user security session is open	00b
b3-b2	RW_PROTECTION_A2	00: Area 2 I^2C access: Read always allowed, Write always allowed 01: Area 2 I^2C access: Read always allowed, Write allowed if I^2C user security session is open 10: Area 2 I^2C access: Read allowed if I^2C user security session is open, Write always allowed 11: Area 2 I^2C access: Read allowed if I^2C security session is open, Write allowed if I^2C security session is open	00b
b5-b4	RW_PROTECTION_A3	00: Area 3 I^2C access: Read always allowed, Write always allowed 01: Area 3 I^2C access: Read always allowed, Write allowed if I^2C user security session is open 10: Area 3 I^2C access: Read allowed if I^2C user security session is open, Write always allowed 11: Area 3 I^2C access: Read allowed if I^2C security session is open, Write allowed if I^2C security session is open	00b
b7-b6	RW_PROTECTION_A4	00: Area 4 I^2C access: Read always allowed, Write always allowed 01: Area 4 I^2C access: Read always allowed, Write allowed if I^2C user security session is open 10: Area 4 I^2C access: Read allowed if I^2C user security session is open, Write always allowed 11: Area 4 I^2C access: Read allowed if I^2C security session is open, Write allowed if I^2C security session is open	00b

1. Refer to *Table 9: System configuration memory map* for the I2CSS register.

Table 34. LOCK_CCFILE(1)

RF	**Command**	**Lock Block (cmd code 22h) @00h/01h** **Ext Lock Block (cmd code 32h) @00h/01h** **Read Block(2) (cmd code 20h) @00h/01h** **Fast Read Block(2) (cmd code C0h) @00h/01h** **Ext Read Block(2) (cmd code 30h) @00h/01h** **Fast Ext Read Block(2) (cmd code C4h) @00h/01h** **Read Multi Block(2) (cmd code 23h) @00h/01h** **Ext Read Multi Block(2) (cmd code 33h) @00h/01h** **Fast Read Multi Block(2) (cmd code C3h) @00h/01h** **Fast Ext Read Multi Block(2) (cmd code C5h) @00h/01h** **Get Multi Block SS (cmd code 2Ch) @00h/01h** **Ext Get Multi Block SS (cmd code 3Ch) @00h/01h**	
	Type	**R always** **b0: W if Block 00h is not already locked,** **b1: W if Block 01h is not already locked.**	
I^2C	**Address**	**E2 = 1, 000Ch**	
	Type	**R always, W if I^2C security session is open**	
Bit	**Name**	**Function**	**Factory Value**
b0	LCKBCK0	0: Block @ 00h is not Write locked 1: Block @ 00h is Write locked	0b
b1	LCKBCK1	0: Block @ 01h is not Write locked 1: Block @ 01h is Write locked	0b
b7-b2	RFU	-	000000b

1. Refer to *Table 9: System configuration memory map* for the LOCK_CCFILE register.
2. With option flag set to 1.

Table 35. LOCK_CFG(1)

RF	**Command**	**Read Configuration (cmd code A0h) @0Fh** **Write Configuration (cmd code A1h) @0Fh**	
	Type	**R always, W if RF configuration security session is open and configuration not locked**	
I²C	**Address**	**E2 = 1, 000Fh**	
	Type	**R always, W if I²C security session is open**	
Bit	**Name**	**Function**	**Factory Value**
b0	LCK_CFG	0: Configuration is unlocked 1: Configuration is locked	0b
b7-b1	RFU	-	0000000b

1. Refer to *Table 9: System configuration memory map* for the LOCK_CFG register.

Table 36. I2C_PWD(1)

RF		**Command**	**No access**	
		Type		
I²C		**Address**	**E2 = 1, 0900h to 0907h, Present/Write password command format.**	
		Type	**R if I²C security session is open, W if I²C security session is open**	
I²C Address	**Bit**	**Name**	**Function**	**Factory Value**
0900h	b7-b0	I2C_PWD	Byte 7 (MSB) of password for I²C security session	00h
0901h	b7-b0		Byte 6 of password for I²C security session	00h
0902h	b7-b0		Byte 5 of password for I²C security session	00h
0903h	b7-b0		Byte 4 of password for I²C security session	00h
0904h	b7-b0		Byte 3 of password for I²C security session	00h
0905h	b7-b0		Byte 2 of password for I²C security session	00h
0906h	b7-b0		Byte 1 of password for I²C security session	00h
0907h	b7-b0		Byte 0 (LSB) of password for I²C security session	00h

1. Refer to *Table 9: System configuration memory map* for the I2C_PWD register.

Table 37. RF_PWD_0(1)

RF	Command	Present Password (cmd code B3h) Write Password (cmd code B1h)	
	Type	WO if RF configuration security session is open	
I²C	Address	No access	
	Type		
Bit	**Name**	**Function**	**Factory Value**
b7-b0	RF_PWD_0	Byte 0 (LSB) of password for RF configuration security session	00h
b7-b0		Byte 1 of password for RF configuration security session	00h
b7-b0		Byte 2 of password for RF configuration security session	00h
b7-b0		Byte 3 of password for RF configuration security session	00h
b7-b0		Byte 4 of password for RF configuration security session	00h
b7-b0		Byte 5 of password for RF configuration security session	00h
b7-b0		Byte 6 of password for RF configuration security session	00h
b7-b0		Byte 7 (MSB) of password for RF configuration security session	00h

1. Refer to *Table 9: System configuration memory map* for the RF_PWD_0 register.

Table 38. RF_PWD_1(1)

RF	Command	Present Password (cmd code B3h) Write Password (cmd code B1h)	
	Type	WO if RF configuration security session is open with RF password 1	
I²C	Address	No access	
	Type		
Bit	**Name**	**Function**	**Factory Value**
b7-b0	RF_PWD_1	Byte 0 (LSB) of password 1 for RF user security session	00h
b7-b0		Byte 1 of password 1 for RF user security session	00h
b7-b0		Byte 2 of password 1 for RF user security session	00h
b7-b0		Byte 3 of password 1 for RF user security session	00h
b7-b0		Byte 4 of password 1 for RF user security session	00h
b7-b0		Byte 5 of password 1 for RF user security session	00h
b7-b0		Byte 6 of password 1 for RF user security session	00h
b7-b0		Byte 7 (MSB) of password 1 for RF user security session	00h

1. Refer to *Table 9: System configuration memory map* for the RF_PWD_1 register.

Table 39. RF_PWD_2(1)

RF	**Command**	**Present Password (cmd code B3h) Write Password (cmd code B1h)**	
	Type	**WO if RF user security session is open with RF password 2**	
I²C	**Address**	**No access**	
	Type		
Bit	**Name**	**Function**	**Factory Value**
b7-b0	RF_PWD_2	Byte 0 (LSB) of password 2 for RF user security session	00h
b7-b0		Byte 1 of password 2 for RF user security session	00h
b7-b0		Byte 2 of password 2 for RF user security session	00h
b7-b0		Byte 3 of password 2 for RF user security session	00h
b7-b0		Byte 4 of password 2 for RF user security session	00h
b7-b0		Byte 5 of password 2 for RF user security session	00h
b7-b0		Byte 6 of password 2 for RF user security session	00h
b7-b0		Byte 7 (MSB) of password 2 for RF user security session	00h

1. Refer to *Table 9: System configuration memory map* for the RF_PWD_2 register.

Table 40. RF_PWD_3(1)

RF	**Command**	**Present Password (cmd code B3h) Write Password (cmd code B1h)**	
	Type	**WO if RF user security session is open with RF password 3**	
I²C	**Address**	**No access**	
	Type		
Bit	**Name**	**Function**	**Factory Value**
b7-b0	RF_PWD_3	Byte 0 (LSB) of password 3for RF user security session	00h
b7-b0		Byte 1 of password 3 for RF user security session	00h
b7-b0		Byte 2 of password 3 for RF user security session	00h
b7-b0		Byte 3 of password 3 for RF user security session	00h
b7-b0		Byte 4 of password 3 for RF user security session	00h
b7-b0		Byte 5 of password 3 for RF user security session	00h
b7-b0		Byte 6 of password 3 for RF user security session	00h
b7-b0		Byte 7 (MSB) of password 3 for RF user security session	00h

1. Refer to *Table 9: System configuration memory map* for the RF_PWD_3 register.

Table 41. I2C_SSO_Dyn(1)

RF	Command	No access	
	Type		
I^2C	Address	E2 = 0, 2004h	
	Type	RO	
Bit	**Name**	**Function**	**Factory Value**
b0	I2C_SSO	0: I^2C security session close 1: I^2C security session open (Set or reset via I^2C Present password command)	0b
b7-b1	RFU	-	0b

1. Refer to *Table 10: Dynamic registers memory map* for the I2C_SS0_Dyn register.

5.6.2 Passwords and security sessions

ST25DVxxx provides protection of user memory and system configuration static registers. RF user and I^2C host can access those protected data by opening security sessions with the help of passwords. Access rights is more restricted when security sessions are closed, and less restricted when security sessions are open.

Dynamic registers and Fast transfer Mode mailbox are not protected by any security session.

There is three type of security sessions, as shown in *Table 42*:

Table 42. Security session type

Security session	Open by presenting	Right granted when security session is open, and until it is closed
RF user	RF password 1, 2 or 3(1) (RF_PWD_1, RF_PWD_2, RF_PWD_3)	RF user access to protected user memory as defined in RFA_iSS registers RF user write access to RF password 1, 2 or 3(2)
RF configuration	RF password 0 (RF_PWD_0)	RF user write access to configuration static registers RF user write access to RF password 0
I^2C	I^2C password (I2C_PWD)	I^2C host access to protected user memory as defined in I2CSS register I^2C host write access to configuration static registers I^2C host write access to I^2C password

1. Password number must be the same as the one selected for protection.
2. Write access to the password number corresponding to the password number presented.

All passwords are 64-bits long, and default factory passwords value is 0000000000000000h.

The ST25DVxxx passwords management is organized around RF and I^2C dedicated set of commands to access the dedicated registers in system configuration area where password values are stored.

The dedicated password commands in RF mode are:

- Write Password command (code B1h): see *Section 7.6.35: Write Password*.
- Present Password command (code B3h): see *Section 7.6.36: Present Password*.

RF user possible actions for security sessions are:

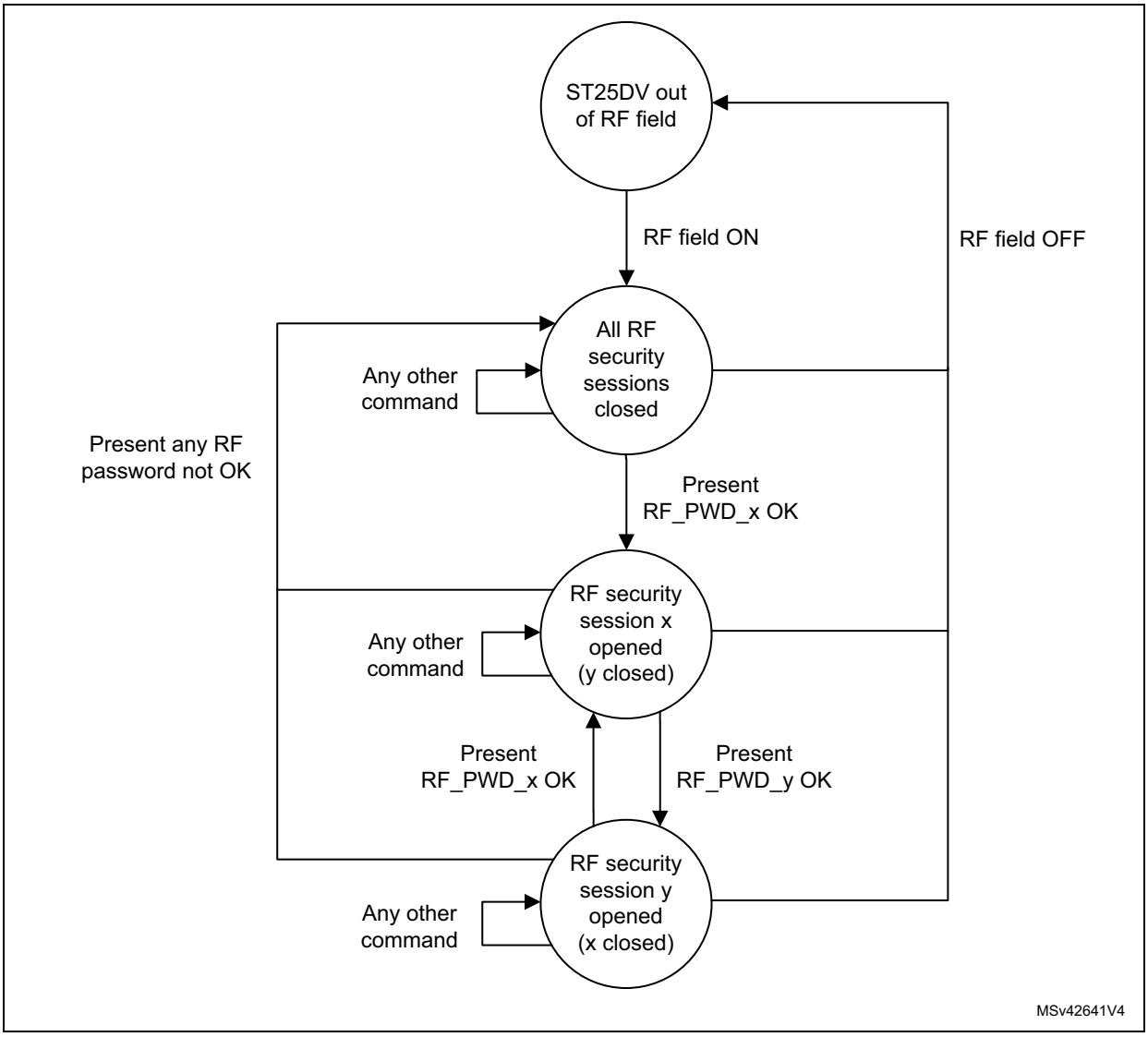
- **Open RF user security session**: Present Password command, with password number 1, 2 or 3 and the valid corresponding password
- **Write RF password**: Present Password command, with password number (0, 1, 2 or 3) and the current valid corresponding password. Then Write Password command, with same password number (0, 1, 2 or 3) and the new corresponding password.
- **Close RF user security session**: Present Password command, with a different password number than the one used to open session or any wrong password. Or remove tag from RF field (POR).
- **Open RF configuration security session**: Present Password command, with password number 0 and the valid password 0.
- **Close RF configuration security session**: Present Password command, with a password number different than 0, or password number 0 and wrong password 0. Or remove tag from RF field (POR).

Opening any new RF security session (user or configuration) automatically close the previously open one (even if it fails).

There is no interaction between I^2C and RF security sessions. Both are independent, and can run in parallel.

Caution: If ST25DVxxx is powered through V_{CC}, removing V_{CC} or setting LPD high during a RF command can abort the command. As a consequence, before writing a new password, RF user should check if V_{CC} is ON, by reading EH_CTRL_Dyn register bit 3 (VCC_ON), and eventually ask host to maintain or to shut down V_{CC}, and not change voltage applied on LPD while issuing the Write Password command in order to avoid password corruption.

Figure 21. RF security sessions management



The dedicated password commands in I^2C mode are:

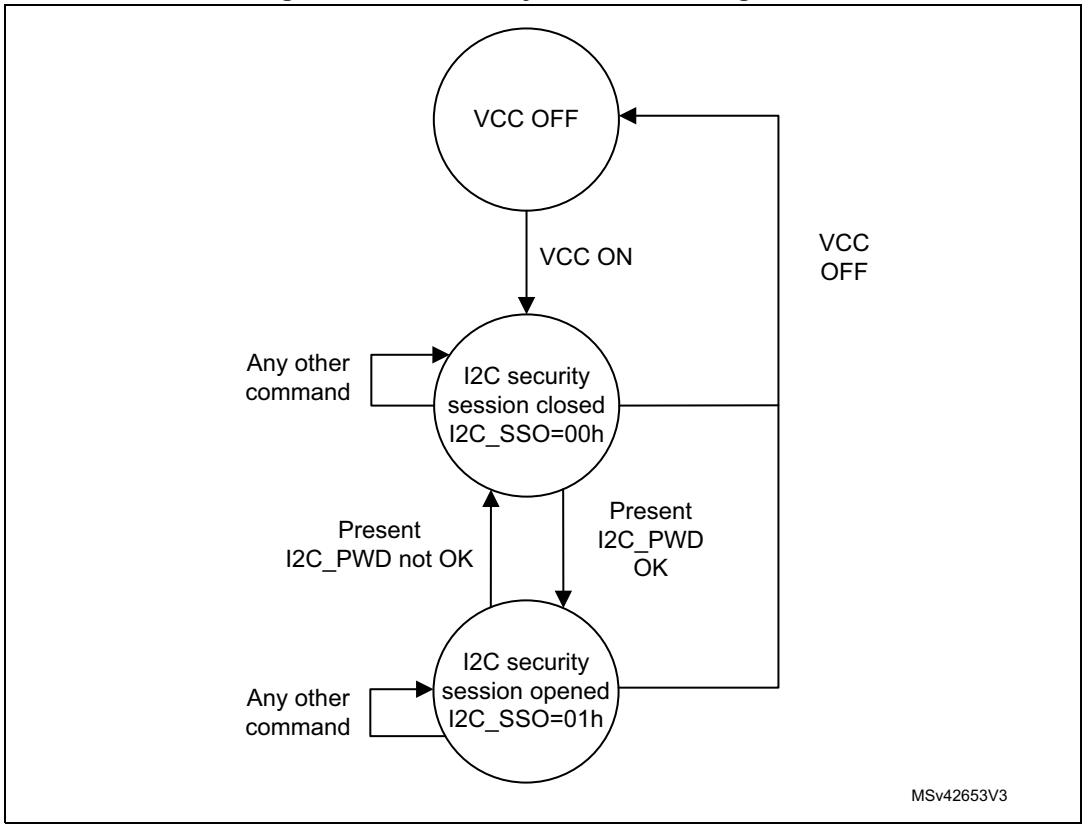
- I^2C Write Password command: see *Section 6.6.2: I^2C write password command description*.
- I^2C Present Password command: see *Section 6.6.2: I^2C write password command description*.

I^2C host possible actions for security sessions are:

- **Open I^2C security session**: I^2C Present Password command with valid I^2C password.
- **Write I^2C password**: I^2C Present Password command with valid I^2C password. Then I^2C Write Password command with new I^2C password.
- **Close I^2C security session**: I^2C Present Password command with wrong I^2C password. Or remove tag V_{CC} power supply (POR).
- **Check if I^2C security session is open**: I^2C host can read the current status (open or closed) of I^2C security session by reading the I2C_SSO_Dyn register.

There is no interaction between I^2C and RF security sessions. Both are independent and can run in parallel.

Figure 22. I^2C security sessions management



5.6.3 User memory protection

On factory delivery, areas are not protected.

Each area can be individually protected in read and/or write access from RF and I^2C.

Area 1 is always readable (from RF and I^2C).

Furthermore, RF blocks 0 and 1 (I^2C bytes 0000h to 00007h) can be independently write locked.

User memory protection from RF access

In RF mode, each memory area of the ST25DVxxx can be individually protected by one out of three available passwords (RF password 1, 2 or 3), and each area can also have individual Read/Write access conditions.

For each area, an RFA_iSS register is used to:

- Select the RF password that unlock the RF user security session for this area
- Select the protection against read and write operations for this area

(See *Table 29: RFA1SS*, *Table 30: RFA2SS*, *Table 31: RFA3SS* and *Table 32: RFA4SS* for details about available read and write protections).

Note: Setting 00b in PWD_CTRL_A$_i$ field means that RF user security session cannot be open by any password for the corresponding area.

When updating RFA$_i$SS registers, the new protection value is effective immediately after the register write completion.

- Rf blocks 0 and 1 are exceptions to this protection mechanism:
  - RF blocks 0 and 1 can be individually write locked by issuing a (Ext) Lock Single Block RF command. Once locked, they cannot be unlock through RF. LOCK_CCFILE register is automatically updated when using (Ext) Lock Single Block command.
  - A RF user needs no password to lock blocks 0 and/or 1.
  - Locking blocks 0 and/or 1 is possible even if the configuration is locked (LOCK_CFG=1).
  - Locking blocks 0 and/or 1 is possible even if the area is write locked.
  - Unlocking area1 (through RFA1SS register) does not unlock blocks 0 and 1 if they have been locked though (Ext) Lock Block command.
  - Once locked, the RF user cannot unlock blocks 0 and/or 1 (can be done by I^{2}C host).

Note: When areas size are modified (ENDAi registers), RFAiSS registers are not modified.

User memory protection from I^{2}C access

In I^{2}C mode, each area can also have individual Read/Write access conditions, but only one I^{2}C password is used to unlock I^{2}C security session for all areas.

The I2CSS register is used to set protection against read and write operation for each area (see *Table 33: I2CSS* for details about available read and write protections).

When updating I2CSS registers, the new protection value is effective immediately after the register write completion.

I^{2}C user memory Bytes 0000h to 0003h (RF Block 0) and 0004h to 0007h (RF Block 1) can be individually locked and unlocked by writing in the LOCK_CCFILE register (by group of 4 Bytes), independently of Area 1 protection. Unlocking Area 1 (through I2CSS register) does not unlock those bytes if they have been locked though the LOCK_CCFILE register.

Note: When areas size are modified (ENDAi registers), I2CSS register is not modified.

Retrieve the security status of a user memory block or byte

RF user can read a block security status by issuing following RF commands:

- (Ext) Get Multiple Blocks Security Status command.
- (Ext) (Fast) Read Single Block with option flag set to 1.
- (Ext) (Fast) Read Multiple Blocks with option flag set to 1.

ST25DV will respond with a Block security status containing a Lock_bit flag as specified in ISO 15693 standard. This lock_bit flag is set to one if block is locked against write.

Lock_bit flag value may vary if corresponding RF user security session is open or closed.

I^{2}C host can retrieve a block security status by reading the I2CSS register to get security status of the corresponding area and by reading the I2C_SSO_Dyn register to know if I^{2}C security session is open or closed.

For blocks 0 and 1 (Bytes 0000h to 0007h in I^2C user memory), lock status can also be read in the LOCK_CCFILE register.

5.6.4 System memory protection

By default, system memory (static registers) is write protected, both in RF and I^2C.

I^2C host must open the I^2C security session (by presenting a valid I^2C password) to enable write access to system configuration static registers.

I^2C host doesn't have read or write access to RF passwords.

By default, I^2C host can read all system configuration static registers (except RF passwords)

In RF, to enable write access to system configuration static registers, RF user must open the RF configuration security session (by presenting a valid RF password 0) and system configuration must not be locked (LOCK_CFG=00h).

RF doesn't have read or write access to I^2C password.

By default, RF user can read all system configuration static registers, except all passwords, LOCK_CCFILE, LOCK_DSFID and LOCK_AFI.

RF configuration lock:

- RF write access to system configuration static registers can be locked by writing 01h in the LOCK_CFG register (by RF or I^2C).
- RF user cannot unlock system configuration if LOCK_CFG=01h, even after opening RF configuration security session (only I^2C host can unlock system configuration).

Device identification registers:

- AFI and DFSID registers can be independently locked by RF user, issuing respectively a Lock AFI and a Lock DSFID command. Lock is definitive: once locked, AFI and DSFID registers cannot be unlocked (either by RF or I^2C).
- Other device identification registers (MEM_SIZE, BLK_SIZE, IC_REF, UID, IC_REV) are read only registers for both RF and I^2C.

5.7 Device Parameter Registers

Table 43. LOCK_DSFID(1)

RF	**Command**	**Lock DSFID (cmd code 2Ah)**	
	Type	**WO if DSFID not locked**	
I^2C	**Address**	**E2 = 1, 0010h**	
	Type	**RO**	
Bit	**Name**	**Function**	**Factory Value**
b0	LOCK_DSFID	0: DSFID is not locked 1: DSFID is locked	0b
b7-b1	RFU	-	0000000b

1. Refer to *Table 9: System configuration memory map* for the LOCK_DSFID register.

Table 44. LOCK_AFI(1)

RF	**Command**	**Lock AFI (cmd code 28h)**	
	Type	**WO if AFI not locked**	
I²C	**Address**	**E2 = 1, 0011h**	
	Type	**RO**	
Bit	**Name**	**Function**	**Factory Value**
b0	LOCK_AFI	0: AFI is not locked 1: AFI is locked	0b
b7-b1	RFU	-	0000000b

1. Refer to *Table 9: System configuration memory map* for the LOCK_AFI register.

Table 45. DSFID(1)

RF	**Command**	**Inventory (cmd code 01h)** **Get System Info (cmd code 2Bh)** **Ext Get System Info (cmd code 3Bh)** **Write DSFID (cmd code 28h)**	
	Type	**R always, W if DSFID not locked**	
I²C	**Address**	**E2 = 1, 0012h**	
	Type	**RO**	
Bit	**Name**	**Function**	**Factory Value**
b7-b0	DSFID	ISO/IEC 15693 Data Storage Format Identifier	00h

1. Refer to *Table 9: System configuration memory map* for the DSFID register.

Table 46. AFI(1)

RF	**Command**	**Inventory (cmd code 01h)** **Get System Info (cmd code 2Bh)** **Ext Get System Info (cmd code 3Bh)** **Write AFI (cmd code 27h)**	
	Type	**R always, W if AFI not locked**	
I²C	**Address**	**E2 = 1, 0013h**	
	Type	**RO**	
Bit	**Name**	**Function**	**Factory Value**
b7-b0	AFI	ISO/IEC 15693 Application Family Identifier	00h

1. Refer to *Table 9: System configuration memory map* for the AFI register.

Table 47. MEM_SIZE(1)

RF		**Command**	**Get System Info(2) (cmd code 2Bh)** **Ext Get System Info (cmd code 3Bh)**	
		Type	**RO**	
I²C		**Address**	**E2=1, 0014h to 0015h**	
		Type	**RO**	
I²C Address	**Bit**	**Name**	**Function**	**Factory Value**
0014h	b7-b0	MEM_SIZE	Address 0015h: LSB byte of the memory size expressed in RF blocks	ST25DV04K-XX: 7Fh ST25DV16K-XX: FFh ST25DV64K-XX: FFh
0015h	b7-b0		Address 0014h: MSB byte of the memory size expressed in RF blocks	ST25DV04K-XX: 00h ST25DV16K-XX: 01h ST25DV64K-XX: 07h

1. Refer to *Table 9: System configuration memory map* for the MEM_SIZE register.
2. Only ST25DV04K-IE and ST25DV04K-JF

Table 48. BLK_SIZE(1)

RF	**Command**	**Get System Info(2) (cmd code 2Bh)** **Ext Get System Info (cmd code 3Bh)**	
	Type	**RO**	
I²C	**Address**	**E2 = 1, 0016h**	
	Type	**RO**	
Bit	**Name**	**Function**	**Factory Value**
b7-b0	BLK_SIZE	RF user memory block size	03h

1. Refer to *Table 9: System configuration memory map* for the BLK_SIZE register.
2. Only ST25DV04K-IE and ST25DV04K-JF

Table 49. IC_REF(1)

RF	**Command**	**Get System Info (cmd code 2Bh)** **Ext Get System Info (cmd code 3Bh)**	
	Type	**RO**	
I²C	**Address**	**E2 = 1, 0017h**	
	Type	**RO**	
Bit	**Name**	**Function**	**Factory Value**
b7-b0	IC_REF	ISO/IEC 15693 IC Reference	ST25DV04K-IE: 24h ST25DV16K-IE: 26h ST25DV64K-IE: 26h ST25DV04K-JF: 24h ST25DV16K-JF: 26h ST25DV64K-JF: 26h

1. Refer to *Table 9: System configuration memory map* for the IC_REF register.

Table 50. UID(1)

RF		**Command**	**Inventory (cmd code 01h)** **Get System Info (cmd code 2Bh)** **Ext Get System Info (cmd code 3Bh)**	
		Type	**RO**	
I²C		**Address**	**E2=1, 0018h to 001Fh**	
		Type	**RO**	
I²C Address	**Bit**	**Name**	**Function**	**Factory Value**
0018h	b7-b0	UID	ISO/IEC 15693 UID byte 0 (LSB)	IC manufacturer serial number
0019h	b7-b0		ISO/IEC 15693 UID byte 1	
001Ah	b7-b0		ISO/IEC 15693 UID byte 2	
001Bh	b7-b0		ISO/IEC 15693 UID byte 3	
001Ch	b7-b0		ISO/IEC 15693 UID byte 4	
001Dh	b7-b0		ISO/IEC 15693 UID byte 5: ST Product code	ST25DV04K-IE: 24h ST25DV16K-IE: 26h ST25DV64K-IE: 26h ST25DV04K-JF: 25h ST25DV16K-JF: 27h ST25DV64K-JF: 27h
001Eh	b7-b0		ISO/IEC 15693 UID byte 6: IC Mfg code	02h
001Fh	b7-b0		ISO/IEC 15693 UID byte 7 (MSB)	E0h

1. Refer to *Table 9: System configuration memory map* for the UID register.

Table 51. IC_REV(1)

RF	**Command**	**No access**	
	Type		
I^2C	**Address**	**E2 = 1, 0020h**	
	Type	**RO**	
Bit	**Name**	**Function**	**Factory Value**
b7-b0	IC_REV	IC revision	Depending on revision

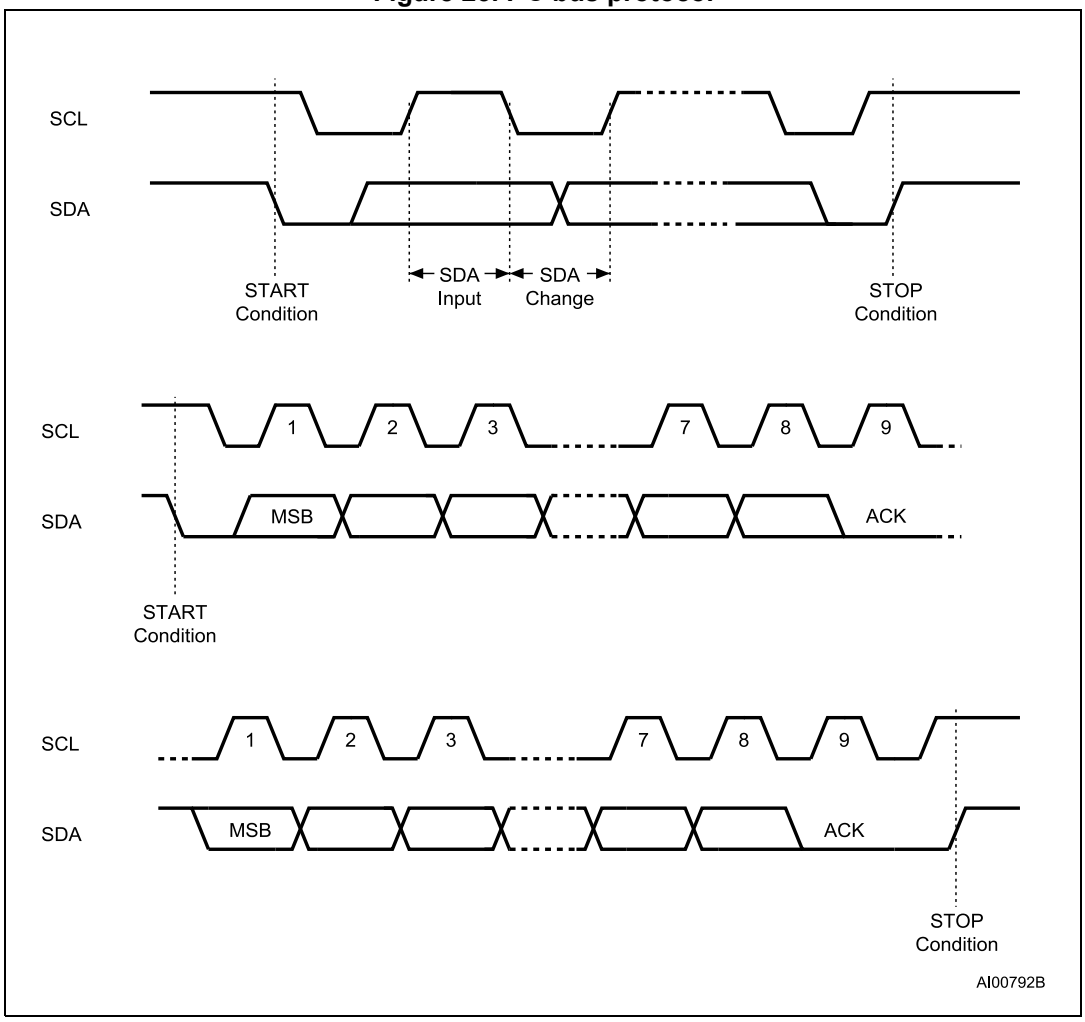
1. Refer to *Table 9: System configuration memory map* for the IC_REV register.

6 I²C operation

6.1 I²C protocol

The device supports the I²C protocol. This is summarized in *Figure 23: I²C bus protocol*. Any device that sends data to the bus is defined as a transmitter, and any device that reads data is defined as a receiver. The device that controls the data transfer is known as the bus master, and the other as the slave device. A data transfer can only be initiated by the bus master, which also provides the serial clock for synchronization. The ST25DVxxx device is a slave in all communications.

Figure 23. I²C bus protocol



6.1.1 Start condition

Start is identified by a falling edge of serial data (SDA) while the serial clock (SCL) is stable in the high state. A Start condition must precede any data transfer command. The device continuously monitors (except during a write cycle) the SDA and the SCL for a Start condition, and does not respond unless one is given.

6.1.2 Stop condition

Stop is identified by a rising edge of serial data (SDA) while the serial clock (SCL) is stable and driven high. A Stop condition terminates communication between the device and the bus master. A Read command that is followed by NoAck can be followed by a Stop condition to force the device into the Standby mode. A Stop condition at the end of a Write command triggers the internal write cycle.

6.1.3 Acknowledge bit (ACK)

The acknowledge bit is used to indicate a successful byte transfer. The bus transmitter, whether a bus master or a slave device, releases the serial data (SDA) after sending eight bits of data. During the 9th clock pulse period, the receiver pulls the SDA low to acknowledge the receipt of the eight data bits.

6.1.4 Data input

During data input, the device samples serial data (SDA) on the rising edge of the serial clock (SCL). For correct device operation, the SDA must be stable during the rising edge of the SCL, and the SDA signal must change *only* when the SCL is driven low.

6.2 I²C timeout

During the execution of an I²C operation, RF communications are not possible.

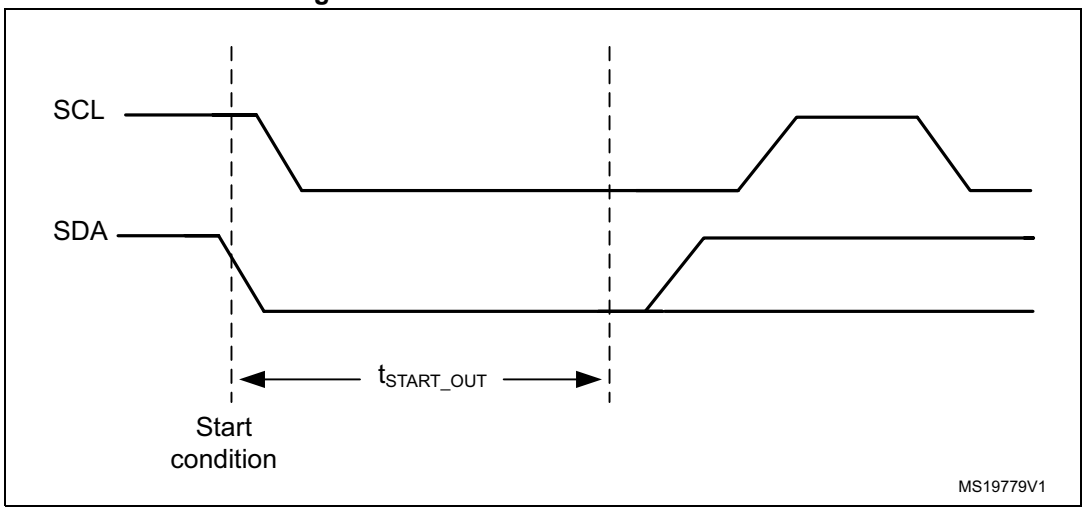
To prevent RF communication freezing due to inadvertent unterminated instructions sent to the I²C bus, the ST25DVxxx features a timeout mechanism that automatically resets the I²C logic block.

6.2.1 I²C timeout on Start condition

I²C communication with the ST25DVxxx starts with a valid Start condition, followed by a device select code.

If the delay between the Start condition and the following rising edge of the Serial Clock (SCL) that samples the most significant of the Device Select exceeds the t_{START_OUT} time (see *Table 208: I²C AC characteristics*), the I²C logic block is reset and further incoming data transfer is ignored until the next valid Start condition.

Figure 24. I²C timeout on Start condition



6.2.2 I²C timeout on clock period

During data transfer on the I²C bus, if the serial clock pulse width high (t_{CHCL}) or serial clock pulse width low (t_{CLCH}) exceeds the maximum value specified in *Table 208: I²C AC characteristics*, the I²C logic block is reset and any further incoming data transfer is ignored until the next valid Start condition.

6.3 Device addressing

To start a communication between the bus master and the slave device, the bus master must initiate a Start condition. Following this, the bus master sends the device select code, shown in *Table 52: Device select code* (on Serial Data (SDA), the most significant bit first).

The device select code consists of a 4-bit device type identifier and a 3-bit Chip Enable "Address" (E2,1,1). To address the memory array, the 4-bit device type identifier is 1010b. Refer to *Table 52: Device select code*.

The eighth bit is the Read/$\overline{\text{Write}}$ bit ($\overline{\text{RW}}$). It is set to 1 for Read and to 0 for Write operations.

Table 52. Device select code

	Device type identifier(1)				Chip Enable address			$\overline{\text{RW}}$
	b7	b6	b5	b4	b3	b2	b1	b0
Device select code	1	0	1	0	E2(2)	1	1	$\overline{\text{RW}}$

1. The most significant bit, b7, is sent first.
2. E2 is not connected to any external pin. It is however used to address the ST25DVxxx as described in *Section 4: Memory management*
   E2 = 0, access to user memory, Dynamic registers or Mailbox.
   E2 =1, access to system area.

If a match occurs on the device select code, the corresponding device gives an acknowledgment on serial data (SDA) during the ninth bit time. If the device does not match the device select code, it deselects itself from the bus, and goes into Standby mode.

Table 53. Operating modes

Mode	$\overline{RW}$ bit	Bytes	Initial sequence
Current address read	1	1	Start, device select, $\overline{RW}$ = 1
Random address read	0	1	Start, device select, $\overline{RW}$ = 0, address
	1		reStart, device select, $\overline{RW}$ = 1
Sequential read	1	≥ 1	Similar to current or random address read
Byte write	0	1	Start, device select, $\overline{RW}$ = 0
Sequential write	0	≤ 256 byte	Start, device select, $\overline{RW}$ = 0

6.4 I²C Write operations

Following a Start condition, the bus master sends a device select code with the Read/$\overline{Write}$ bit ($\overline{RW}$) reset to 0. The device acknowledges this, and waits for two address bytes. The device responds to each address byte with an acknowledge bit, and then waits for the data byte.

Each data byte in the memory has a 16-bit (two-byte wide) address. The most significant byte (see *Table 54: Address most significant byte*) is sent first, followed by the least significant byte (see *Table 55: Address least significant byte*). Bits b15 to b0 form the address of the byte in memory.

Table 54. Address most significant byte

b15	b14	b13	b12	b11	b10	b9	b8

Table 55. Address least significant byte

b7	b6	b5	b4	b3	b2	b1	b0

When the bus master generates a Stop condition immediately after the Ack bit (in the tenth-bit time slot), either at the end of a byte write or a sequential write, the internal write cycle is triggered. A Stop condition at any other time slot does not trigger the internal write cycle.

After the Stop condition, the delay t_W, and the successful completion of a Write operation, the device's internal address counter is incremented automatically, to point to the next byte address after the last one that was modified.

After an unsuccessful write operation, ST25DVxxx enters in I2C dead state: internal address counter is not incremented, and ST25DVxxx is waiting for a full new I2C instruction.

During the internal write cycle, the serial data (SDA) signal is disabled internally, and the device does not respond to any requests.

Caution: I²C Writing data in user or system memory (EEPROM), transit via the 256-Bytes Fast Transfer Mode's buffer. Consequently Fast Transfer Mode must be deactivated before starting any write operation in user or system memory, otherwise command will be NotACK, programming is not done and device goes in standby mode.

6.4.1 I²C Byte write

After the device select code and the address bytes, the bus master sends one data byte.

If byte write is not inhibited, the device replies with Ack. The bus master terminates the transfer by generating a Stop condition and byte location is modified (see *Figure 25: Write mode sequences when write is not inhibited*)

If byte write is inhibited, the device replies with NoAck. The bus master terminates the transfer by generating a Stop condition and byte location not is modified (see *Figure 26: Write mode sequences when write is inhibited*).

Byte write is inhibited if byte complies with one of the following conditions:

- Byte is in user memory and is write protected with LOCK_CCFILE register.
- Byte is in user memory and is write protected with I2CSS register, and I²C security session is closed.
- Byte is in user memory and Fast Transfer Mode is activated.
- Byte is in system memory and is a Read Only register.
- Byte is in system memory and I²C security session is closed.
- Byte is in Fast Transfer Mode's mailbox and is not the first Byte of mailbox.
- Byte is in Fast Transfer Mode's mailbox and mailbox is busy.
- Byte is in Fast Transfer Mode's mailbox and Fast Transfer Mode is not activated.
- Byte is in dynamic registers area and is a Read Only register.

6.4.2 I²C Sequential write

The I²C sequential write allows up to 256 bytes to be written in one command, provided they are all located in the same user memory area or are all located in writable addresses.

After each byte is transferred, the internal byte address counter is incremented.

For each byte sent by the bus master:

- If byte write is not inhibited, the device replies with Ack.
- If byte write is inhibited, the device replies with NoAck.

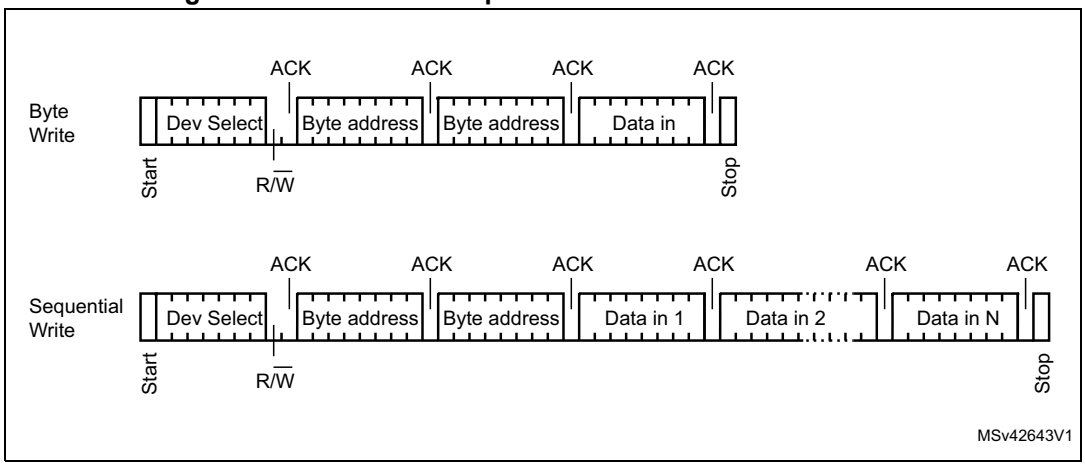
The transfer is terminated by the bus master generating a Stop condition:

- If all bytes have been Ack'ed, internal programming of all bytes is done.
- If some bytes have been NotAck'ed, no internal programming is done (0 byte written).

Byte write is inhibited if byte complies with conditions described in *Section 6.4.1: I²C Byte write*, in addition:

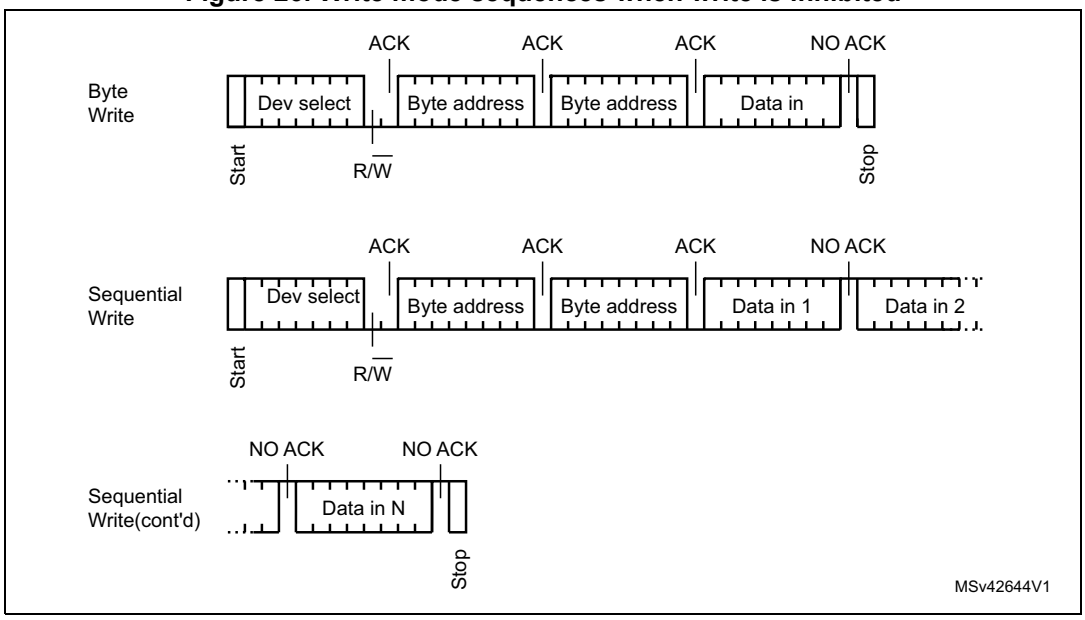
- Byte is in user memory but does not belong to same area than previous received byte (area border crossing is forbidden).
- 256 write occurrence have already been reached in the same sequential write.

Figure 25. Write mode sequences when write is not inhibited



Note: N ≤ 256

Figure 26. Write mode sequences when write is inhibited



Note: N ≤ 256

6.4.3 Minimizing system delays by polling on ACK

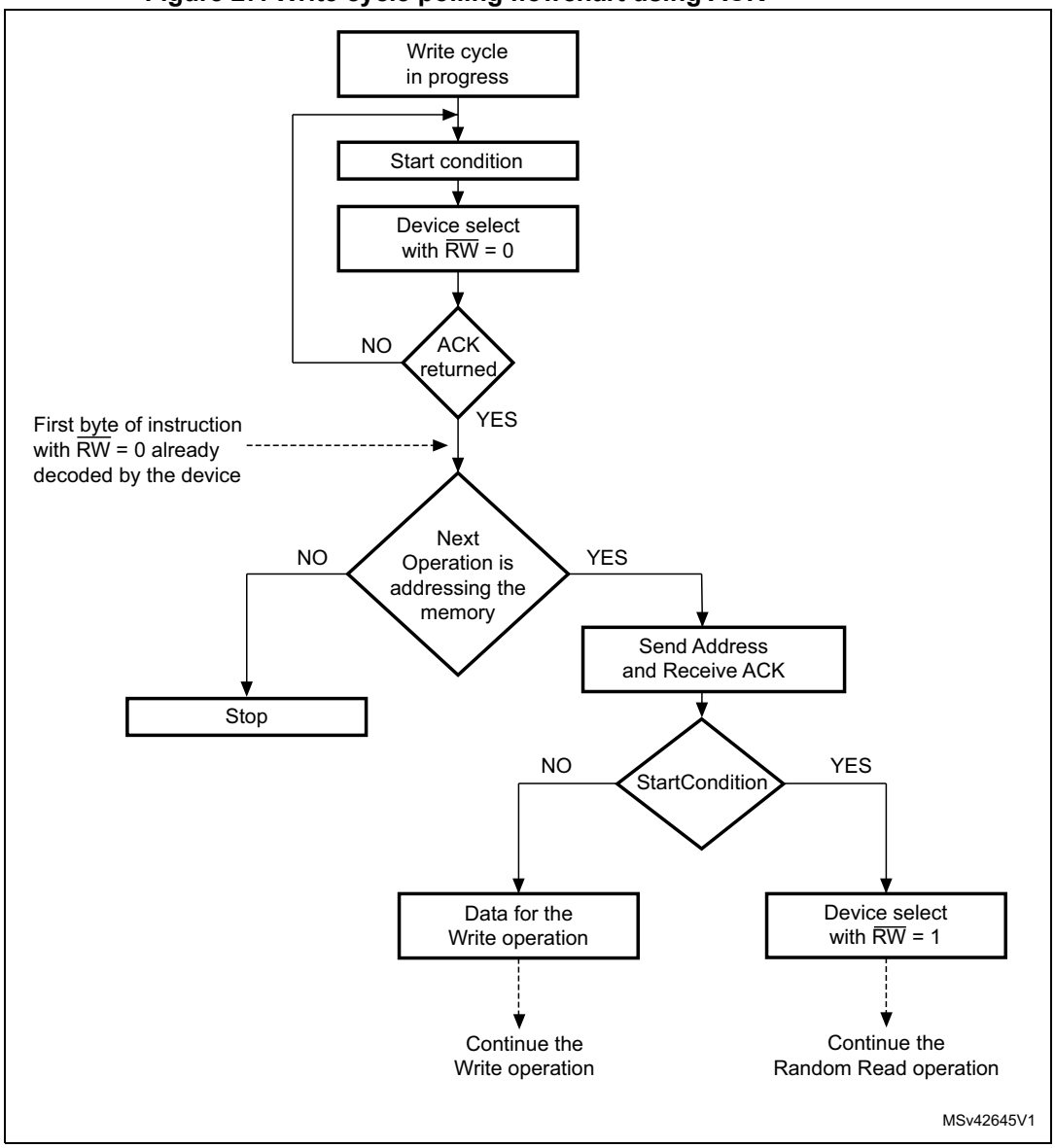
During the internal write cycle, the device disconnects itself from the bus, and writes a copy of the data from its internal latches to the memory cells. The maximum I²C write time (t_w) is shown in *Table 208: I²C AC characteristics*, but the typical time is shorter. To make use of this, a polling sequence can be used by the bus master.

The sequence, as shown in *Figure 27: Write cycle polling flowchart using ACK*, is:

- Initial condition: a write cycle is in progress.
- Step 1: the bus master issues a Start condition followed by a device select code (the first byte of the new instruction).
- Step 2: if the device is busy with the internal write cycle, no Ack is returned and the bus master goes back to Step 1. If the device has terminated the internal write cycle, it responds with an Ack, indicating that the device is ready to receive the second part of the instruction (the first byte of this instruction having been sent during Step 1).

Note: There is no need of polling when writing in dynamic registers or in mailbox, since programming time is null.

Figure 27. Write cycle polling flowchart using ACK



6.5 I²C read operations

Read operation in user memory is performed successfully only if:

- Area to which the byte belongs is not read protected by I2CSS register.
- Area to which the byte belongs is read protected by I2CSS register, but I²C security session is open.

Read operations in system memory and dynamic registers are done independently of any protection mechanism, except I2C_PWD register which needs I²C security session to be open first.

Read operation in fast Transfer Mode's mailbox is performed successfully only if Fast Transfer Mode is activated.

If read is not successful, ST25DVxxx releases the bus and I²C host reads byte value FFh.

After the successful completion of a read operation, the device's internal address counter is incremented by one, to point to the next byte address.

After an unsuccessful read operation, ST25DVxxx enters in I²C dead state: internal address counter is not incremented, and ST25DVxxx is waiting for a full new I²C instruction.

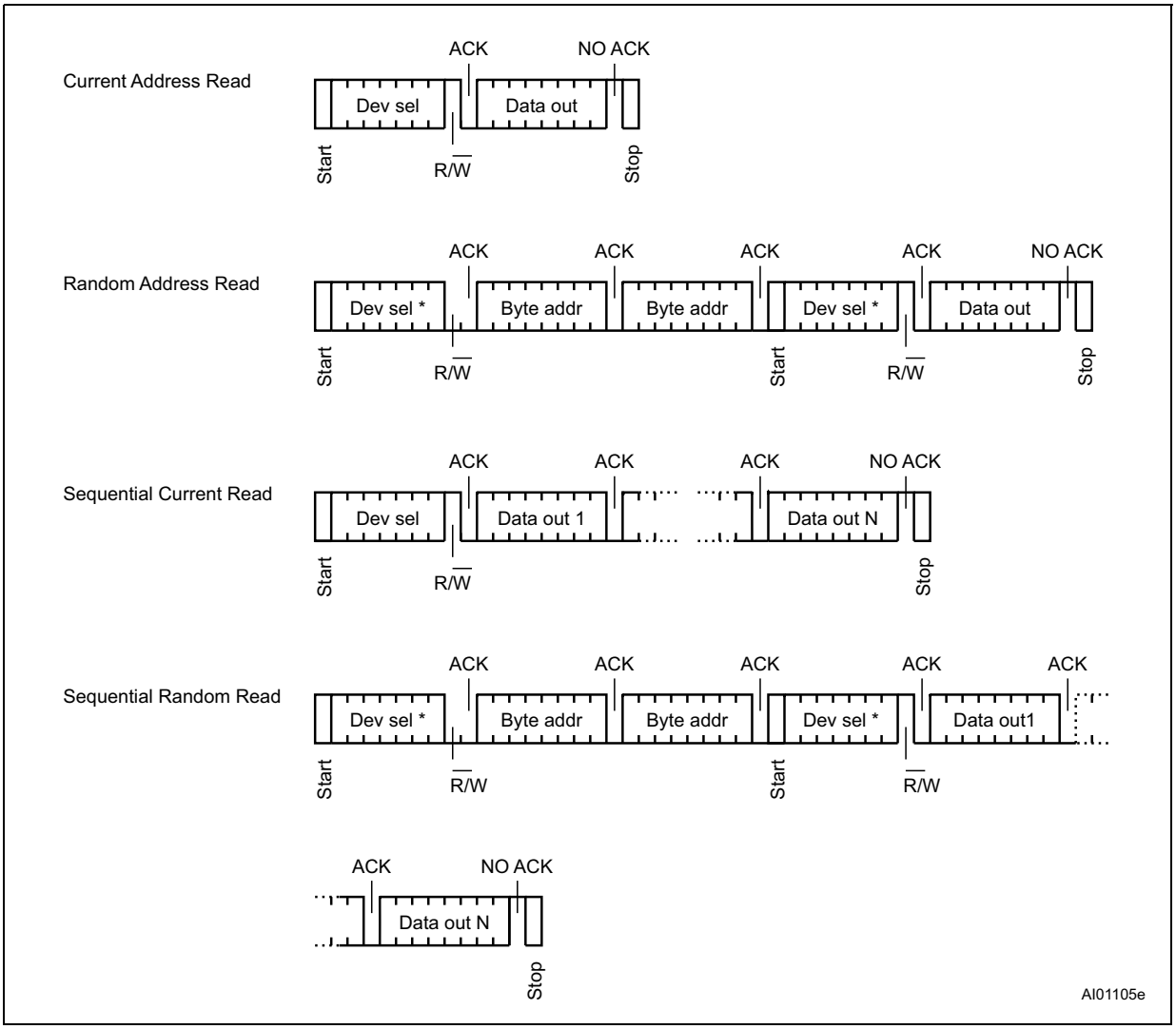
6.5.1 Random Address Read

A dummy write is first performed to load the address into this address counter (as shown in *Figure 28: Read mode sequences*) but *without* sending a Stop condition. Then, the bus master sends another Start condition, and repeats the device select code, with the Read/Write bit (RW) set to 1. The device acknowledges this, and outputs the contents of the addressed byte. The bus master must *not* acknowledge the byte, and terminates the transfer with a Stop condition.

6.5.2 Current Address Read

For the Current Address Read operation, following a Start condition, the bus master only sends a device select code with the Read/Write bit (RW) set to 1. The device acknowledges this, and outputs the byte addressed by the internal address counter. The counter is then incremented. The bus master terminates the transfer with a Stop condition, as shown in *Figure 28: Read mode sequences*, without acknowledging the byte.

Figure 28. Read mode sequences



6.5.3 Sequential Read access

This operation can be used after a Current Address Read or a Random Address Read. The bus master *does* acknowledge the data byte output, and sends additional clock pulses so that the device continues to output the next byte in sequence. To terminate the stream of bytes, the bus master must *not* acknowledge the last byte, and *must* generate a Stop condition, as shown in *Figure 28: Read mode sequences*.

The output data comes from consecutive addresses, with the internal address counter automatically incremented after each byte output.

Sequential read in user memory:

- Sequential read cannot cross area borders. After reaching area border, device continues to output FFh
- There is no roll over inside area or at the end of user memory (ST25DVxxx returns only FFh after last user memory byte address).

Sequential read in system memory:

- There is no roll over after reaching end of system memory (ST25DV returns only FFh after last system memory byte address).
- Sequential read in dynamic registers:
- It is possible to read sequentially dynamic registers and Fast Transfer Mode's mailbox (contiguous I²C addresses).

Sequential read in dynamic registers:

- There is no roll over at the end of the mailbox (ST25DV returns only FFh after last system memory byte address).

6.5.4 Acknowledge in Read mode

For all Read commands, the device waits, after each byte read, for an acknowledgment during the ninth bit time. If the bus master does not drive Serial Data (SDA) low during this time, the device terminates the data transfer and switches to its Standby mode.

6.6 I²C password management

The ST25DVxxx controls I²C security session using an I²C 64-bit password. This I²C password is managed with two I²C dedicated commands: I²C present password and I²C write password.

6.6.1 I²C present password command description

The I²C present password command is used in I²C mode to present the password to the ST25DVxxx. This is used to open I²C security session or to allow I²C password modification (see *Section 5.6: Data Protection* for detailed explanation about password usage).

Following a Start condition, the bus master sends a device select code with the Read/Write bit (R/W) reset to 0 and the Chip Enable bit E2 at 1. The device acknowledges this, as shown in *Figure 29: I²C Present Password Sequence*, and waits for two I²C password address bytes, 09h and 00h. The device responds to each address byte with an acknowledge bit, and then waits for the eight password data bytes, the validation code, 09h, and a resend of the eight password data bytes. The most significant byte of the password is sent first, followed by the least significant bytes.

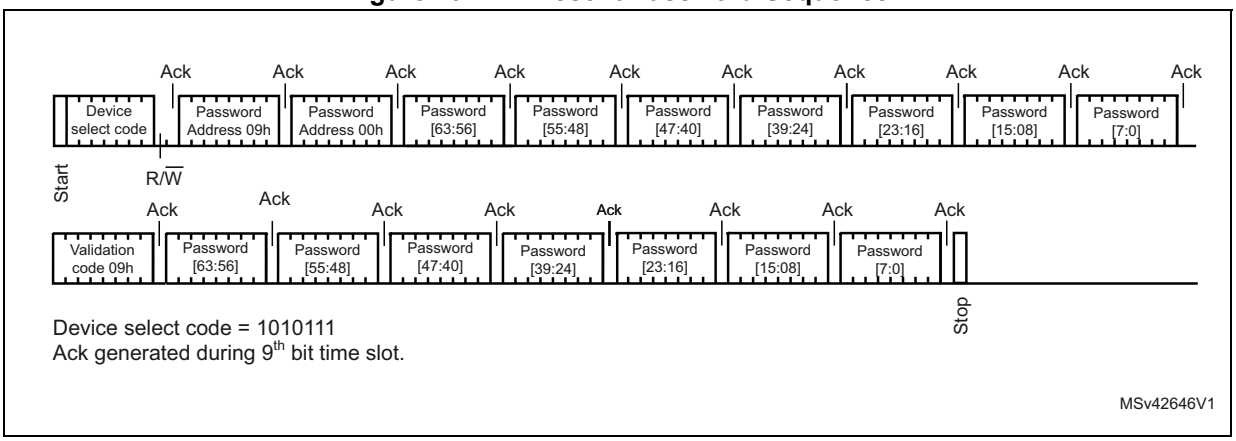
It is necessary to send the 64-bit password twice to prevent any data corruption during the sequence. If the two 64-bit passwords sent are not exactly the same, the ST25DVxxx does not start the internal comparison.

When the bus master generates a Stop condition immediately after the Ack bit (during the tenth bit time slot), an internal delay equivalent to the write cycle time is triggered. A Stop condition at any other time does not trigger the internal delay. During that delay, the ST25DVxxx compares the 64 received data bits with the 64 bits of the stored I²C password. If the values match, the I²C security session is open after the internal delay, and the I2C_SSO_Dyn register is set to 01h. If the values do not match, the I²C security session is closed and I2C_SSO_dyn register is set to 00h.

During the internal delay, the serial data (SDA) signal is disabled internally, and the device does not respond to any requests.

I2C_SSO_Dyn is a Dynamic register, it can be checked via I²C host to know If I²C security session is open.

Figure 29. I²C Present Password Sequence



6.6.2 I²C write password command description

The I²C write password command is used to update the I²C password value (register I2C_PWD). It cannot be used to update any of the RF passwords. After the write cycle, the new I²C password value is automatically activated. The I²C password value can only be modified after issuing a valid I²C present password command.

Following a Start condition, the bus master sends a device select code with the Read/$\overline{\text{Write}}$ bit ($\overline{\text{RW}}$) reset to 0 and the Chip Enable bit E2 at 1. The device acknowledges this, as shown in *Figure 30: I²C Write Password Sequence*, and waits for the two I²C password address bytes, 09h and 00h. The device responds to each address byte with an acknowledge bit, and then waits for the four password data bytes, the validation code, 07h, and a resend of the eight password data bytes. The most significant byte of the password is sent first, followed by the least significant bytes.

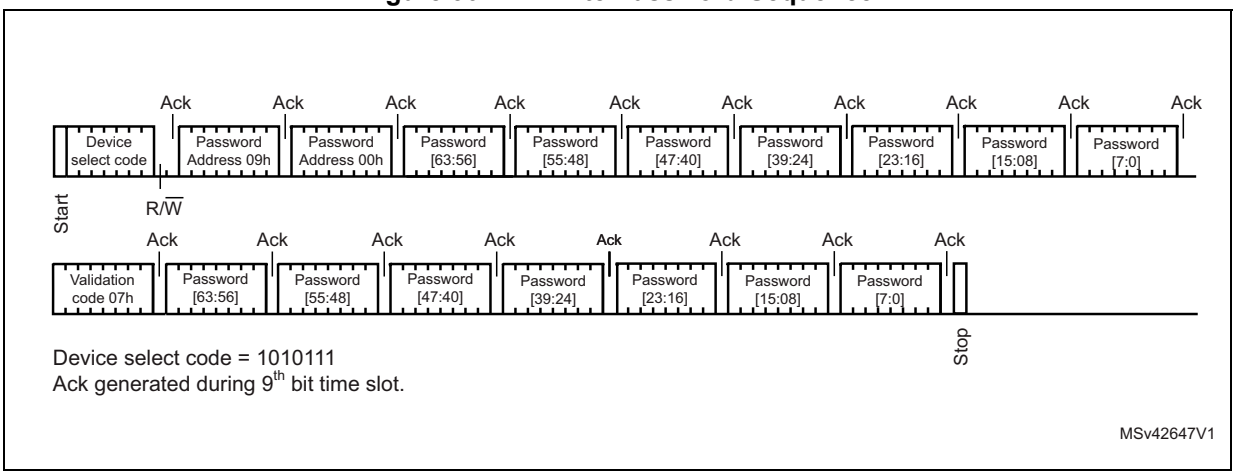
It is necessary to send twice the 64-bit password to prevent any data corruption during the write sequence. If the two 64-bit passwords sent are not exactly the same, the ST25DVxxx does not modify the I²C password value.

When the bus master generates a Stop condition immediately after the Ack bit (during the tenth bit time slot), the internal write cycle is triggered. A Stop condition at any other time does not trigger the internal write cycle.

During the internal write cycle, the serial data (SDA) signal is disabled internally, and the device does not respond to any requests.

Caution: I²C write password command data transits via the 256-Bytes Fast Transfer Mode's buffer. Consequently Fast Transfer Mode must be deactivated before issuing a write password command, otherwise command is NotACK (after address LSB), and programming is not done and device goes in standby mode.

Figure 30. I²C Write Password Sequence



7 RF operations

Contactless exchanges are performed in RF mode as specified by ISO/IEC 15693 or NFC Forum Type 5. The ST25DVxxx communicates via the 13.56 MHz carrier electromagnetic wave on which incoming data are demodulated from the received signal amplitude modulation (ASK: amplitude shift keying). The received ASK wave is 10% or 100% modulated with a data rate of 1.6 Kbit/s using the 1/256 pulse coding mode or a data rate of 26 Kbit/s using the 1/4 pulse coding mode.

Outgoing data are generated by the ST25DVxxx load variation using Manchester coding with one or two subcarrier frequencies at 423 kHz and 484 kHz. Data are transferred from the ST25DVxxx at 6.6 Kbit/s in low data rate mode and 26 Kbit/s in high data rate mode. The ST25DVxxx supports the 53 Kbit/s in high data rate mode in one subcarrier frequency at 423 kHz.

The ST25DVxxx follows ISO/IEC 15693 or NFC Forum Type 5 recommendation for radio-frequency power and signal interface and for anticollision and transmission protocol.

7.1 RF communication

7.1.1 Access to a ISO/IEC 15693 device

The dialog between the “RF reader” and the ST25DVxxx takes place as follows:

These operations use the RF power transfer and communication signal interface described below (see Power transfer, Frequency and Operating field). This technique is called RTF (Reader talk first).

- activation of the ST25DVxxx by the RF operating field of the reader,
- transmission of a command by the reader (ST25DVxxx detects carrier amplitude modulation)
- transmission of a response by the ST25DVxxx(ST25DVxxx modulates is load clocked at subcarrier rate)

Operating field

The ST25DVxxx operates continuously between the minimum and maximum values of the electromagnetic field H defined in *Table 211: RF characteristics*. The Reader has to generate a field within these limits.

Power transfer

Power is transferred to the ST25DVxxx by radio frequency at 13.56 MHz via coupling antennas in the ST25DVxxx and the Reader. The RF operating field of the reader is transformed on the ST25DVxxx antenna to an AC voltage which is rectified, filtered and internally regulated. During communications, the amplitude modulation (ASK) on this received signal is demodulated by the ASK demodulator

Frequency

The ISO 15693 standard defines the carrier frequency (f_C) of the operating field as 13.56 MHz ±7 kHz.

7.2 RF communication and energy harvesting

As the current consumption can affect the AC signal delivered by the antenna, RF communications with ST25DVxxx are not guaranteed during voltage delivery on the energy harvesting analog output V_EH.

RF communication can disturb and possibly stop Energy Harvesting mode.

7.3 Fast Transfer Mode mailbox access in RF

Thanks to dedicated commands, the RF interface has the possibility to check Mailbox availability, and the capability to access it directly to put or get a message from it (see *Section 5.1: Fast transfer mode (FTM)* for specific features).

7.4 RF protocol description

7.4.1 Protocol description

The transmission protocol (or simply “the protocol”) defines the mechanism used to exchange instructions and data between the VCD (Vicinity Coupling Device) and the ST25DVxxx in both directions. It is based on the concept of “VCD talks first”.

This means that a ST25DVxxx does not start transmitting unless it has received and properly decoded an instruction sent by the VCD. The protocol is based on an exchange of:

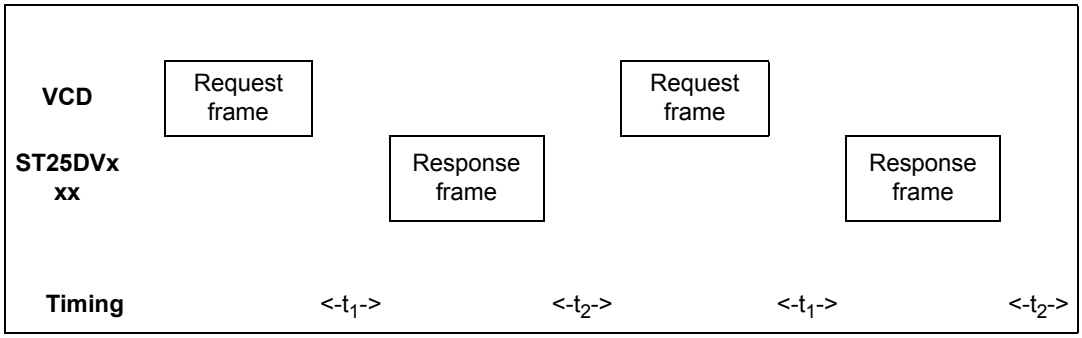
- a request from the VCD to the ST25DVxxx,
- a response from the ST25DVxxx to the VCD.

Each request and each response are contained in a frame. The frame are delimited by a Start of Frame (SOF) and End of Frame (EOF).

The protocol is bit-oriented. The number of bits transmitted in a frame is a multiple of eight (8), that is an integer number of bytes.

A single-byte field is transmitted least significant bit (LSBit) first. A multiple-byte field is transmitted least significant byte (LSByte) first and each byte is transmitted least significant bit (LSBit) first.

Figure 31. ST25DVxxx protocol timing



7.4.2 ST25DVxxx states referring to RF protocol

The ST25DVxxx can be in one of four states:

- Power-off
- Ready
- Quiet
- Selected

Transitions between these states are specified in *Figure 32: ST25DVxxx state transition diagram* and *Table 56: ST25DVxxx response depending on Request_flags*.

Power-off state

The ST25DVxxx is in the Power-off state when it does not receive enough energy from the VCD.

Ready state

The ST25DVxxx is in the Ready state when it receives enough energy from the VCD. When in the Ready state, the ST25DVxxx answers any request where the Select_flag is not set.

Quiet state

When in the Quiet state, the ST25DVxxx answers any request with the Address_flag set, except for Inventory requests.

Selected state

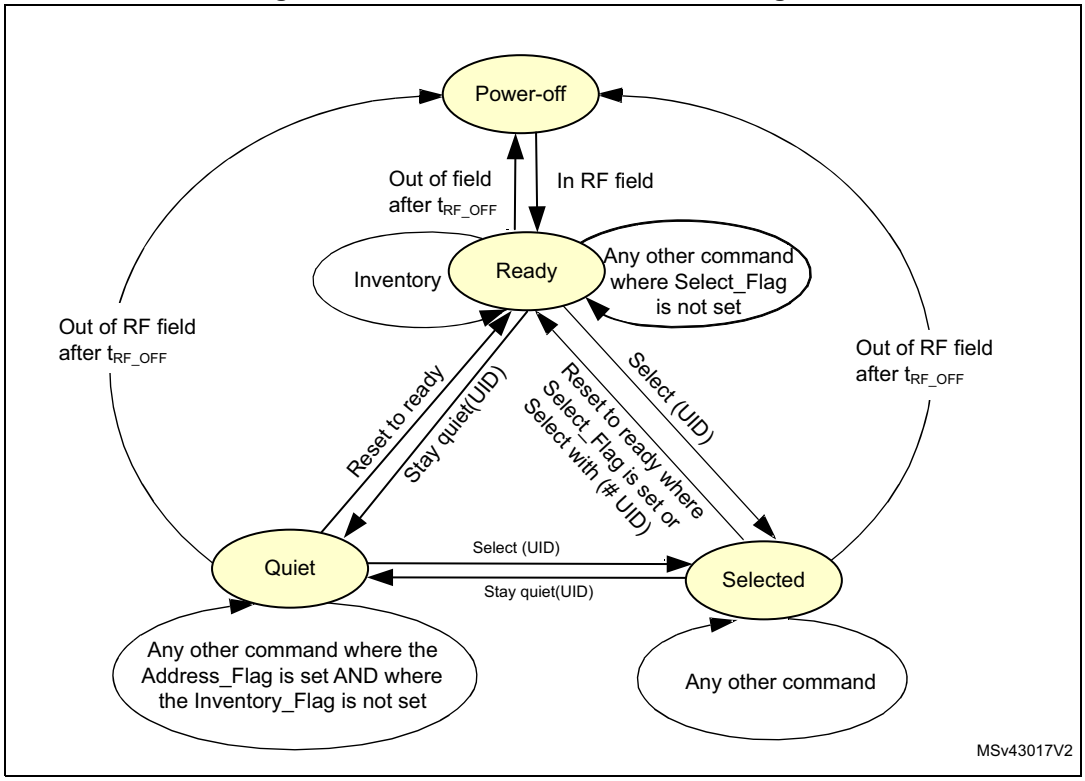
In the Selected state, the ST25DVxxx answers any request in all modes (see *Section 7.4.3: Modes*):

- Request in Select mode with the Select_flag set
- Request in Addressed mode if the UID matches
- Request in Non-Addressed mode as it is the mode for general requests

Table 56. ST25DVxxx response depending on Request_flags

Flags	Address_flag		Select_flag	
	1 Addressed	0 Non addressed	1 Selected	0 Non selected
ST25DVxxx in Ready or Selected state (Devices in Quiet state do not answer)	-	X	-	X
ST25DVxxx in Selected state	-	X	X	-
ST25DVxxx in Ready, Quiet or Selected state (the device which matches the UID)	X	-	-	X
Error (03h) or no response (command dependent)	X	-	X	-

Figure 32. ST25DVxxx state transition diagram



1. The ST25DVxxx returns to the Power Off state if the tag is out of the RF field for at least t_{RF_OFF}.

The intention of the state transition method is that only one ST25DVxxx should be in the Selected state at a time.

When the Select_flag is set to 1, the request shall NOT contain a unique ID.

When the address_flag is set to 0, the request shall NOT contain a unique ID.

7.4.3 Modes

The term “mode” refers to the mechanism used in a request to specify the set of ST25DVxxx devices that shall execute the request.

Addressed mode

When the Address_flag is set to 1 (Addressed mode), the request contains the Unique ID (UID) of the addressed ST25DVxxx.

Any ST25DVxxx that receives a request with the Address_flag set to 1 compares the received Unique ID to its own. If it matches, then the ST25DVxxx executes the request (if possible) and returns a response to the VCD as specified in the command description.

If the UID does not match, then it remains silent.

Non-addressed mode (general request)

When the Address_flag is cleared to 0 (Non-Addressed mode), the request does not contain a Unique ID.

Select mode

When the Select_flag is set to 1 (Select mode), the request does not contain a unique ID. The ST25DVxxx in the Selected state that receives a request with the Select_flag set to 1 executes it and returns a response to the VCD as specified in the command description.

Only the ST25DVxxx in the Selected state answers a request where the Select_flag is set to 1.

The system design ensures that only one ST25DVxxx can be in the Select state at a time.

7.4.4 Request format

The request consists of:

- an SOF,
- flags,
- a command code,
- parameters and data,
- a CRC,
- an EOF.

Table 57. General request format

SOF	Request_flags	Command code	Parameters	Data	2 byte CRC	EOF

7.4.5 Request flags

In a request, the “flags” field specifies the actions to be performed by the ST25DVxxx and whether corresponding fields are present or not.

The flags field consists of eight bits. Bit 3 (Inventory_flag) of the request flag defines the contents of the four MSBs (bits 5 to 8). When bit 3 is reset (0), bits 5 to 8 define the

ST25DVxxx selection criteria. When bit 3 is set (1), bits 5 to 8 define the ST25DVxxx Inventory parameters.

Table 58. Definition of request flags 1 to 4

Bit No	Flag	Level	Description
Bit 1	Subcarrier_flag(1)	0	A single subcarrier frequency is used by the ST25DVxxx
		1	Two subcarriers are used by the ST25DVxxx
Bit 2	Data_rate_flag(2)	0	Low data rate is used
		1	High data rate is used
Bit 3	Inventory_flag	0	The meaning of flags 5 to 8 is described in *Table 59: Request flags 5 to 8 when inventory_flag, Bit 3 = 0*
		1	The meaning of flags 5 to 8 is described in *Table 60: Request flags 5 to 8 when inventory_flag, Bit 3 = 1*
Bit 4	Protocol_extension_flag	0	No Protocol format extension
		1	Protocol format extension. Reserved for future use.

1. Subcarrier_flag refers to the ST25DVxxx-to-VCD communication.
2. Data_rate_flag refers to the ST25DVxxx-to-VCD communication.

Table 59. Request flags 5 to 8 when inventory_flag, Bit 3 = 0

Bit nb	Flag	Level	Description
Bit 5	Select flag(1)	0	The request is executed by any ST25DVxxx according to the setting of Address_flag
		1	The request is executed only by the ST25DVxxx in Selected state
Bit 6	Address flag	0	The request is not addressed. UID field is not present. The request is executed by all ST25DVxxxs.
		1	The request is addressed. UID field is present. The request is executed only by the ST25DVxxx whose UID matches the UID specified in the request.
Bit 7	Option flag	0	Option not activated.
		1	Option activated.
Bit 8	RFU	0	-

1. If the Select_flag is set to 1, the Address_flag is set to 0 and the UID field is not present in the request.

Table 60. Request flags 5 to 8 when inventory_flag, Bit 3 = 1

Bit nb	Flag	Level	Description
Bit 5	AFI flag	0	AFI field is not present
		1	AFI field is present
Bit 6	Nb_slots flag	0	16 slots
		1	1 slot
Bit 7	Option flag	0	-
Bit 8	RFU	0	-

7.4.6 Response format

The response consists of:

- an SOF,
- flags,
- parameters and data,
- a CRC,
- an EOF.

Table 61. General response format

SOF	Response_flags	Parameters	Data	2 byte CRC	EOF

7.4.7 Response flags

In a response, the flags indicate how actions have been performed by the ST25DVxxx and whether corresponding fields are present or not. The response flags consist of eight bits.

Table 62. Definitions of response flags 1 to 8

Bit Nb	Flag	Level	Description
Bit 1	Error_flag	0	No error
		1	Error detected. Error code is in the “Error” field.
Bit 2	RFU	0	-
Bit 3	RFU	0	-
Bit 4	Extension flag	0	No extension
Bit 5	RFU	0	-
Bit 6	RFU	0	-
Bit 7	RFU	0	-
Bit 8	RFU	0	-

7.4.8 Response and error code

If the Error_flag is set by the ST25DVxxx in the response, the Error code field is present and provides information about the error that occurred.

Error codes not specified in *Table 63: Response error code definition* are reserved for future use.

Table 63. Response error code definition

Error code	Meaning
01h	Command is not supported.
02h	Command is not recognized (format error).
03h	The option is not supported.
0Fh	Error with no information given.
10h	The specified block is not available.
11h	The specified block is already locked and thus cannot be locked again.
12h	The specified block is locked and its contents cannot be changed.
13h	The specified block was not successfully programmed.
14h	The specified block was not successfully locked.
15h	The specified block is protected in read.

7.5 Timing definition

t_1: ST25DVxxx response delay

Upon detection of the rising edge of the EOF received from the VCD, the ST25DVxxx waits for a t_{1nom} time before transmitting its response to a VCD request or switching to the next slot during an inventory process. Values of t_1 are given in *Table 64: Timing values*.

t_2: VCD new request delay

t_2 is the time after which the VCD may send an EOF to switch to the next slot when one or more ST25DVxxx responses have been received during an Inventory command. It starts from the reception of the EOF from the ST25DVxxxs.

The EOF sent by the VCD may be either 10% or 100% modulated regardless of the modulation index used for transmitting the VCD request to the ST25DVxxx.

t_2 is also the time after which the VCD may send a new request to the ST25DVxxx, as described in *Figure 31: ST25DVxxx protocol timing*.

Values of t_2 are given in *Table 64: Timing values*.

t_3: VCD new request delay when no response is received from the ST25DVxxx

t_3 is the time after which the VCD may send an EOF to switch to the next slot when no ST25DVxxx response has been received.

The EOF sent by the VCD may be either 10% or 100% modulated regardless of the modulation index used for transmitting the VCD request to the ST25DVxxx.

From the time the VCD has generated the rising edge of an EOF:

- If this EOF is 100% modulated, the VCD waits for a time at least equal to t_{3min} for 100% modulation before sending a new EOF.
- If this EOF is 10% modulated, the VCD waits for a time at least equal to t_{3min} for 10% modulation before sending a new EOF.

Table 64. Timing values(1)

	Minimum (min) values		Nominal (nom) values	Maximum (max) values
	100% modulation	10% modulation		
t_1	4320 / f_c = 318.6 µs		4352 / f_c = 320.9 µs	4384 / f_c = 323.3 µs(2)
t_2	4192 / f_c = 309.2 µs		No t_{nom}	No t_{max}
t_3	t_{1max}(3)(3) + t_{SOF}(4)	t_{1max}(3) + t_{NRT}(5) + t_{2min}	No t_{nom}	No t_{max}

1. The tolerance of specific timings is ± 32/f_C.
2. VCD request will not be interpreted during the first milliseconds following the RF field rising.
3. t_{1max} does not apply for write-alike requests. Timing conditions for write-alike requests are defined in the command description.
4. t_{SOF} is the time taken by the ST25DVxxx to transmit an SOF to the VCD. t_{SOF} depends on the current data rate: High data rate or Low data rate.
5. t_{NRT} is the nominal response time of the ST25DVxxx. t_{NRT} depends on V_{ICC} to ST25DVxxx data rate and subcarrier modulation mode.

7.6 RF Commands

7.6.1 RF command code list

The ST25DVxxx supports the following legacy and extended RF command set:

- **Inventory**, used to perform the anticollision sequence.
- **Stay Quiet**, used to put the ST25DVxxx in quiet mode, where it does not respond to any inventory command.
- **Select**, used to select the ST25DVxxx. After this command, the ST25DVxxx processes all Read/Write commands with Select_flag set.
- **Reset To Ready**, used to put the ST25DVxxx in the ready state.
- **Read Single Block** and **Extended Read Single Block**, used to output the 32 bit of the selected block and its locking status.
- **Write Single Block** and **Extended Write Single Block**, used to write and verify the new content for an update of a 32 bit block, provided that it is not in a locked memory area.
- **Read Multiple Blocks** and **Extended Read Multiple Block**, used to read the selected blocks in an unique area, and send back their value.
- **Write Multiple Blocks** and **Extended Write Multiple Block**, used to write and verify the new content for an update of up to 4 blocks located in the same memory area, which was not previously locked for writing.
- **Write AFI**, used to write the 8-bit value in the AFI register.
- **Lock AFI**, used to lock the AFI register.
- **Write DSFID**, used to write the 8-bit value in the DSFID register.
- **Lock DSFID**, used to lock the DSFID register.
- **Get System information** and **Extended Get System Information**, used to provide the system information value.
- **Get System information**, used to provide the standard system information values.
- **Extended Get System Information**, used to provide the extended system information values.
- **Write Password**, used to update the 64 bit of the selected areas or configuration password, but only after presenting the current one.
- **Lock Block** and **Extended Lock block**, used to write the CC file blocks security status bits (Protect the CC File content against writing).
- **Present Password**, enables the user to present a password to open a security session.
- **Fast Read Single Block** and **Fast Extended Read Single Block,** used to output the 32 bits of the selected block and its locking status at doubled data rate.
- **Fast Read Multiple Blocks** and **Fast Extended Read Multiple Blocks**, used to read the selected blocks in a single area and send back their value at doubled data rate.
- **Read Message**, used to output up to 256 byte of the Mailbox.
- **Read Message Length**, used to output the Mailbox message length.
- **Fast Read Message**, used to output up to 256 byte of the mailbox, at double data rate.
- **Write Message**, used to write up to 256 byte in the Mailbox.
- **Fast Read Message Length, used to ouput the mailbox length, at double data rate.**

- **Fast Write Message, used to write up to 256 bytes in the mailbox, with answer at double data rate.**
- **Read Configuration**, used to read static configuration registers.
- **Write Configuration**, used to write static configuration registers.
- **Read Dynamic Configuration,** used to read dynamic register.
- **Write Dynamic Configuration,** used to write dynamic register.
- **Fast Read Dynamic Configuration**, used to read dynamic register, at double data rate.
- **Fast Write Dynamic Configuration**, used to write dynamic register, with answer at double data rate.
- **Manage GPO,** used to drive GPO output value when corresponding GPO mode is enabled.

7.6.2 Command codes list

The ST25DVxxx supports the commands described in this section. Their codes are given in *Table 65*.

Table 65. Command codes

Command code standard	Function	Command code custom	Function
01h	*Inventory*	A0h	*Read Configuration*
02h	*Stay Quiet*	A1h	*Write Configuration*
20h	*Read Single Block*	A9h	*Manage GPO*
21h	*Write Single Block*	AAh	*Write Message*
22h	*Lock block*	ABh	*Read Message Length*
23h	*Read Multiple Blocks*	ACh	*Read Message*
24h	*Write Multiple Blocks*	ADh	*Read Dynamic Configuration*
25h	*Select*	AEh	*Write Dynamic Configuration*
26h	*Reset to Ready*	B1h	*Write Password*
27h	*Write AFI*	B3h	*Present Password*
28h	*Lock AFI*	C0h	*Fast Read Single Block*
29h	*Write DSFID*	C3h	*Fast Read Multiple Blocks*
30h	*Extended Read Single Block*	C4h	*Fast Extended Read Single Block*
31h	*Extended Write Single Block*	C5h	*Fast Extended Read Multiple Block*
32h	*Extended Lock block*	CAh	*Fast Write Message*
33h	*Extended Read Multiple Blocks*	CBh	*Fast Read Message Length*
34h	*Extended Write Multiple Blocks*	CCh	*Fast Read Message*
2Ah	*Lock DSFID*	CDh	*Fast Read Dynamic Configuration*
2Bh	*Get System Info*	CEh	*Fast Write Dynamic Configuration*
2Ch	*Get Multiple Block Security Status*		
3Bh	*Extended Get System Info*		
3Ch	*Extended Get Multiple Block Security Status*		

7.6.3 General Command Rules

In case of a valid command, the following paragraphs will describe the expected behavior for each command.

But in case of an invalid command, in a general manner, the ST25DVxxx will behave as follows:

1. if flag usage is incorrect, the error code 03h will be issued only if the right UID is used in the command, otherwise no response will be issued.
2. error 02h will be issued if the custom command is used with the manufacturer code different from the ST one

Another case is if I^2C is busy. In this case, any RF command (except Inventory, Select, Stay quiet and Reset to ready) will get 0Fh error code as response only:

a) if select flag and address flags are not set at the same time (except if ST25DVxxx is in quiet state)

b) if select flag is set and ST25DVxxx is in selected state.

For all other commands, if I^2C is busy, no response will be issued by ST25DVxxx.

7.6.4 Inventory

Upon receiving the Inventory request, the ST25DVxxx runs the anticollision sequence. The Inventory_flag is set to 1. The meaning of flags 5 to 8 is shown in *Table 60: Request flags 5 to 8 when inventory_flag, Bit 3 = 1*.

The request contains:

- the flags
- the Inventory command code (001)
- the AFI if the AFI flag is set
- the mask length
- the mask value if mask length is different from 0
- the CRC

The ST25DVxxx does not generate any answer in case of error.

Table 66. Inventory request format

Request SOF	Request_flags	Inventory	Optional AFI	Mask length	Mask value	CRC16	Request EOF
-	8 bits	01h	8 bits	8 bits	0 - 64 bits	16 bits	-

The response contains:

- the flags
- the Unique ID

Table 67. Inventory response format

Response SOF	Response_flags	DSFID	UID	CRC16	Response EOF
-	8 bits	8 bits	64 bits	16 bits	-

During an Inventory process, if the VCD does not receive an RF ST25DVxxx response, it waits for a time t_3 before sending an EOF to switch to the next slot. t_3 starts from the rising edge of the request EOF sent by the VCD.

- If the VCD sends a 100% modulated EOF, the minimum value of t_3 is:
  $t_3min = 4384/f_C$ (323.3µs) + t_{SOF}
- If the VCD sends a 10% modulated EOF, the minimum value of t_3 is:
  $t_3min = 4384/f_C$ (323.3µs) + t_{NRT} + t_{2min}

where:

- t_{SOF} is the time required by the ST25DVxxx to transmit an SOF to the VCD,
- t_{NRT} is the nominal response time of the ST25DVxxx.

t_{NRT} and t_{SOF} are dependent on the ST25DVxxx-to-VCD data rate and subcarrier modulation mode.

Note: In case of error, no response is sent by ST25DVxxx.

7.6.5 Stay Quiet

On receiving the Stay Quiet command, the ST25DVxxx enters the Quiet state if no error occurs, and does NOT send back a response. There is NO response to the Stay Quiet command even if an error occurs.

When in the Quiet state:

- the ST25DVxxx does not process any request if the Inventory_flag is set,
- the ST25DVxxx processes any Addressed request.

The ST25DVxxx exits the Quiet state when:

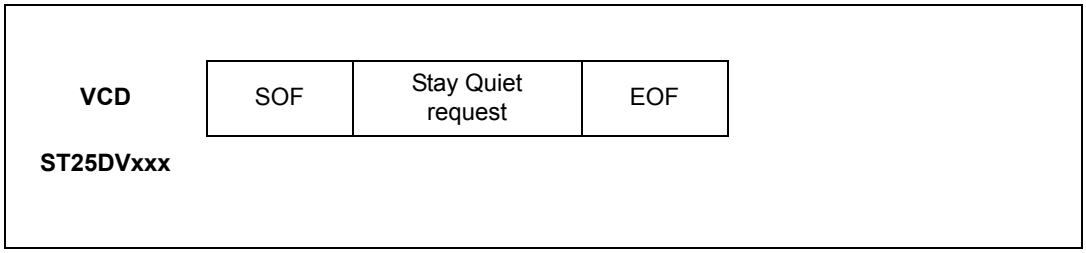
- it is reset (power off),
- receiving a Select request. It then goes to the Selected state,
- receiving a Reset to Ready request. It then goes to the Ready state.

Table 68. Stay Quiet request format

Request SOF	Request flags	Stay Quiet	UID	CRC16	Request EOF
-	8 bits	02h	64 bits	16 bits	-

The Stay Quiet command must always be executed in Addressed mode (Select_flag is reset to 0 and Address_flag is set to 1).

Figure 33. Stay Quiet frame exchange between VCD and ST25DVxxx



7.6.6 Read Single Block

On receiving the Read Single Block command, the ST25DVxxx reads the requested block and sends back its 32-bit value in the response. The Option_flag is supported, when set response include the Block Security Status.

Block number is coded on 1 Byte and only first 256 blocks of ST25DV16K-xx and ST25DV64K-xx can be addressed using this command.

Table 69. Read Single Block request format

Request SOF	Request_flags	Read Single Block	UID(1)	Block number	CRC16	Request EOF
-	8 bits	20h	64 bits	8 bits	16 bits	-

1. Gray color means that the field is optional.

Request parameters:

- Request flags
- UID (optional)
- Block number

Table 70. Read Single Block response format when Error_flag is NOT set

Response SOF	Response_flags	Block security status(1)	Data	CRC16	Response EOF
-	8 bits	8 bits	32 bits	16 bits	-

1. Gray color means that the field is optional.

Response parameters:

- Block security status if Option_flag is set (see *Table 71: Block security status*)
- Four bytes of block data

Table 71. Block security status

b_7	b_6	b_5	b_4	b_3	b_2	b_1	b_0
Reserved for future use. All at 0.							0: Current block not locked 1: Current block locked

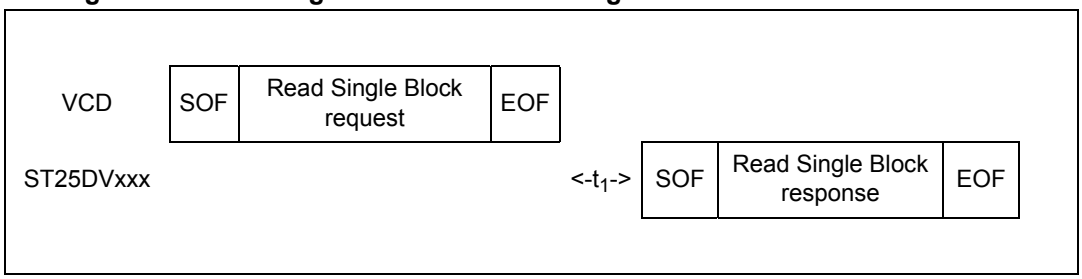
Table 72. Read Single Block response format when Error_flag is set

Response SOF	Response_flags	Error code	CRC16	Response EOF
-	8 bits	8 bits	16 bits	-

Response parameter:

- Error code as Error_flag is set
  - 03h: command option not supported
  - 0Fh: error with no information
  - 10h: the specified block is not available
  - 15h: the specified block is read-protected

Figure 34. Read Single Block frame exchange between VCD and ST25DVxxx



7.6.7 Extended Read Single Block

On receiving the Extended Read Single Block command, the ST25DVxxx reads the requested block and sends back its 32-bit value in the response.

When the Option_flag is set, the response includes the Block Security Status.

Block number is coded on 2 Bytes so all memory blocks of ST25DV16K-xx and ST25DV64K-xx can be addressed using this command.

Table 73. Extended Read Single Block request format

Request SOF	Request_flags	Extended Read Single Block	UID(1)	Block number	CRC16	Request EOF
-	8 bits	30h	64 bits	16 bits	16 bits	-

1. Gray color means that the field is optional.

Request parameters:

- Request flags
- UID (optional)
- Block number

Table 74. Extended Read Single Block response format when Error_flag is NOT set

Response SOF	Response_flags	Block security status(1)	Data	CRC16	Response EOF
-	8 bits	8 bits	32 bits	16 bits	-

1. Gray color means that the field is optional.

Response parameters:

- Block security status if Option_flag is set (see *Table 71: Block security status*)
- Four bytes of block data

Table 75. Block security status

b_7	b_6	b_5	b_4	b_3	b_2	b_1	b_0
Reserved for future use. All at 0.							0: Current block not locked 1: Current block locked

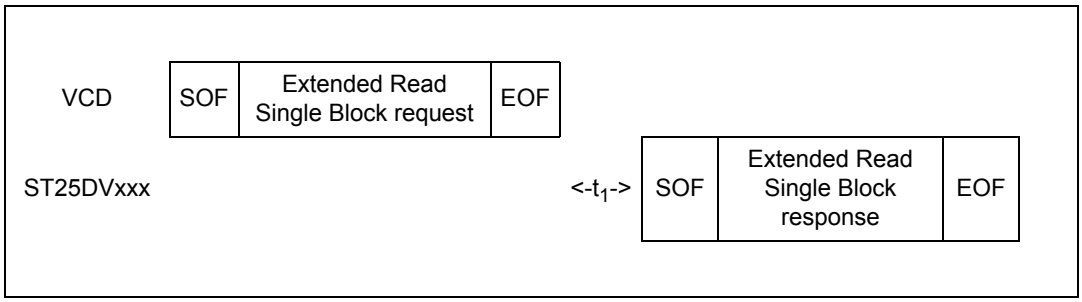
Table 76. Extended Read Single Block response format when Error_flag is set

Response SOF	Response_flags	Error code	CRC16	Response EOF
-	8 bits	8 bits	16 bits	-

Response parameter:

- Error code as Error_flag is set
  - 03h: command option not supported or no response
  - 0Fh: error with no information
  - 10h: the specified block is not available
  - 15h: the specified block is read-protected

Figure 35. Extended Read Single Block frame exchange between VCD and ST25DVxxx



7.6.8 Write Single Block

On receiving the Write Single Block command, the ST25DVxxx writes the data contained in the request to the targeted block and reports whether the write operation was successful in the response. When the Option_flag is set, wait for EOF to respond.

During the RF write cycle W_t, there should be no modulation (neither 100% nor 10%), otherwise the ST25DVxxx may not program correctly the data into the memory. The W_t time is equal to t_{1nom} + N × 302 µs (N is an integer).

Block number is coded on 1 Byte and only first 256 blocks of ST25DV16K-xx and ST25DV64K-xx can be addressed using this command.

Table 77. Write Single Block request format

Request SOF	Request_flags	Write Single Block	UID(1)	Block number	Data	CRC16	Request EOF
-	8 bits	21h	64 bits	8 bits	32 bits	16 bits	-

1. Gray color means that the field is optional.

Request parameters:

- Request flags
- UID (optional)
- Block number
- Data

Table 78. Write Single Block response format when Error_flag is NOT set

Response SOF	Response_flags	CRC16	Response EOF
-	8 bits	16 bits	-

Response parameter:

- No parameter. The response is sent back after the writing cycle.

Table 79. Write Single Block response format when Error_flag is set

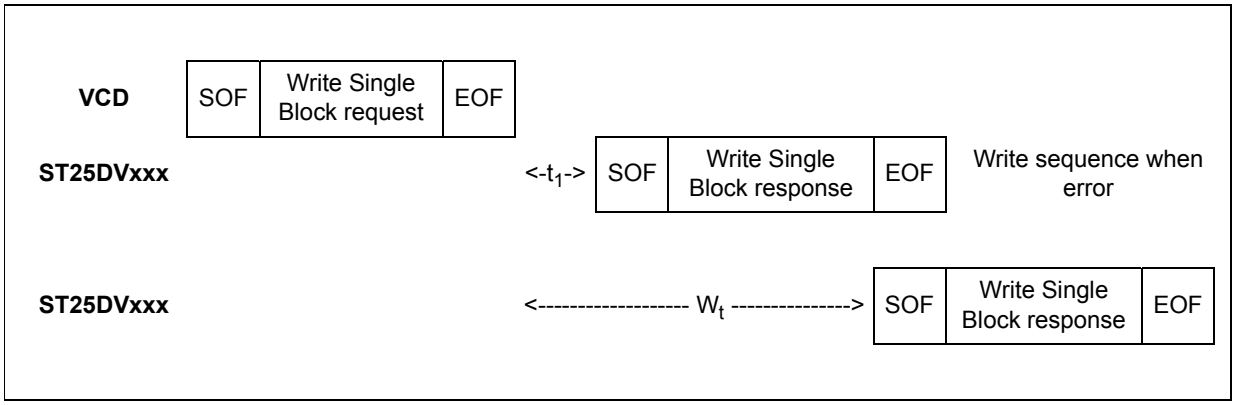
Response SOF	Response_flags	Error code	CRC16	Response EOF
-	8 bits	8 bits	16 bits	-

Response parameter:

- Error code as Error_flag is set(a):
  - 03h: command option not supported
  - 0Fh: error with no information given
  - 10h: the specified block is not available
  - 12h: the specified block is locked or protected and its contents cannot be changed
  - 13h: the specified block was not successfully programmed

a. For more details, see *Figure 6: Memory organization*

Figure 36. Write Single Block frame exchange between VCD and ST25DVxxx



7.6.9 Extended Write Single Block

On receiving the Extended Write Single command, the ST25DVxxx writes the data contained in the request to the targeted block and reports whether the write operation was successful in the response. When the Option_flag is set, wait for EOF to respond.

During the RF write cycle W_t, there should be no modulation (neither 100% nor 10%), otherwise the ST25DVxxx may not program correctly the data into the memory. The W_t time is equal to t_{1nom} + N × 302 µs (N is an integer).

Block number is coded on 1 Byte and only first 256 blocks of ST25DV16K-xx and ST25DV64K-xx can be addressed using this command.

Table 80. Extended Write Single request format

Request SOF	Request_flags	Extended Write Single Block	UID(1)	Block number	Data	CRC16	Request EOF
-	8 bits	31h	64 bits	16 bits	32 bits	16 bits	-

1. Gray color means that the field is optional.

Request parameters:

- Request flags
- UID (optional)
- Block number
- Data

Table 81. Extended Write Single response format when Error_flag is NOT set

Response SOF	Response_flags	CRC16	Response EOF
-	8 bits	16 bits	-

Response parameter:

- No parameter. The response is sent back after the writing cycle.

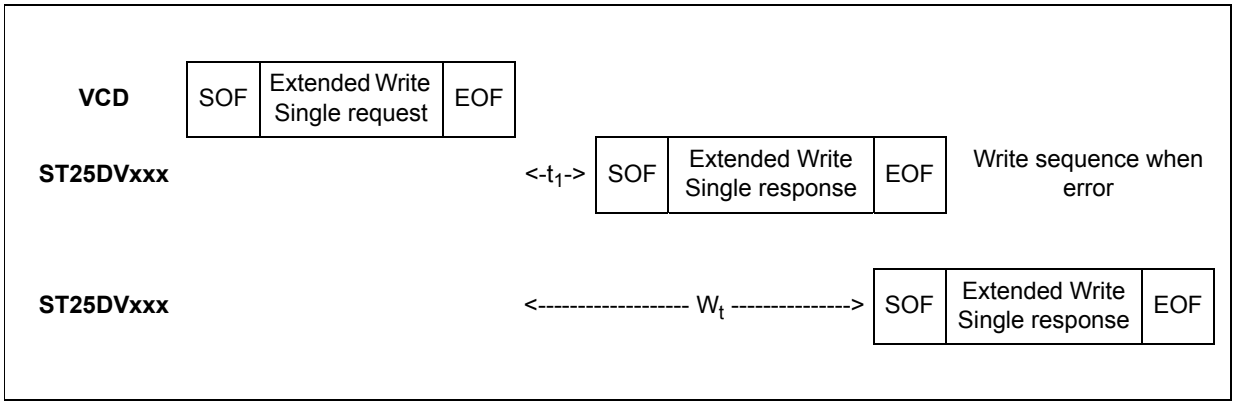
Table 82. Extended Write Single response format when Error_flag is set

Response SOF	Response_flags	Error code	CRC16	Response EOF
-	8 bits	8 bits	16 bits	-

Response parameter:

- Error code as Error_flag is set:
  - 03h: command option not supported
  - 0Fh: error with no information given
  - 10h: the specified block is not available
  - 12h: the specified block is locked and its contents cannot be changed
  - 13h: the specified block was not successfully programmed

Figure 37. Extended Write Single frame exchange between VCD and ST25DVxxx



7.6.10 Lock block

On receiving the Lock block request, the ST25DVxxx locks the single block value permanently and protects its content against new writing.

This command is only applicable for the blocks 0 and 1 which may include a CC file.

For a global protection of a area, update accordingly the RFA_iSS bits in the system area. The Option_flag is supported, when set wait for EOF to respond.

During the RF write cycle W_t, there should be no modulation (neither 100% nor 10%), otherwise the ST25DVxxx may not lock correctly the single block value in memory. The W_t time is equal to t_{1nom} + N × 302 µs (N is an integer).

Table 83. Lock block request format

Request SOF	Request_flags	Lock block	UID(1)	block number	CRC16	Request EOF
-	8 bits	22h	64 bits	8 bits	16 bits	-

1. Gray color means that the field is optional.

Request parameter:

- Request Flags
- UID (optional)
- Block number (only value 00h or 01h) are allowed to protect the CCfile in case of NDEF usage.

Table 84. Lock block response format when Error_flag is NOT set

Response SOF	Response_flags	CRC16	Response EOF
-	8 bits	16 bits	-

Response parameter:

- No parameter

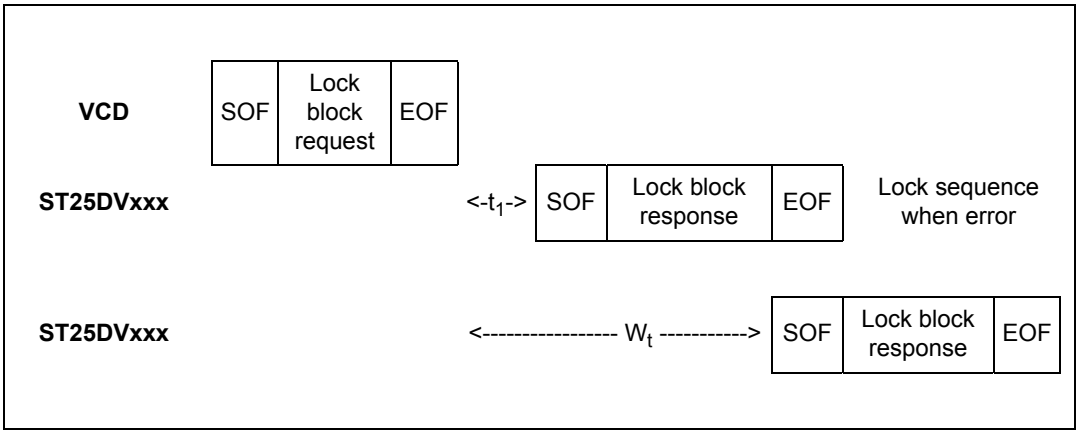
Table 85. Lock single block response format when Error_flag is set

Response SOF	Response_flags	Error code	CRC16	Response EOF
-	8 bits	8 bits	16 bits	-

Response parameter:

- Error code as Error_flag is set
  - 03h: command option not supported
  - 10h: block not available
  - 11h: the specified block is already locked and thus cannot be locked again
  - 14h: the specified block was not successfully locked

Figure 38. Lock single block frame exchange between VCD and ST25DVxxx



7.6.11 Extended Lock block

On receiving the extended Lock block request, the ST25DVxxx locks the single block value permanently and protects its content against new writing.

This command is only applicable for the blocks 0 and 1 which may include a CC file.

For a global protection of a area, update accordingly the AiSS bits in the system area. When the Option_flag is set, wait for EOF to respond.

During the RF write cycle W_t, there should be no modulation (neither 100% nor 10%), otherwise the ST25DVxxx may not lock correctly the single block value in memory. The W_t time is equal to t_{1nom} + 18 × 302 µs.

Table 86. Extended Lock block request format

Request SOF	Request_flags	Extended Lock block	UID(1)	block number	CRC16	Request EOF
-	8 bits	32h	64 bits	16 bits	16 bits	-

1. Gray color means that the field is optional.

Request parameter:

- Request Flags
- UID (optional)
- Block number (only value 00h or 01h) are allowed to protect the CCfile in case of NDEF usage.

Table 87. Extended Lock block response format when Error_flag is NOT set

Response SOF	Response_flags	CRC16	Response EOF
-	8 bits	16 bits	-

Response parameter:

- No parameter

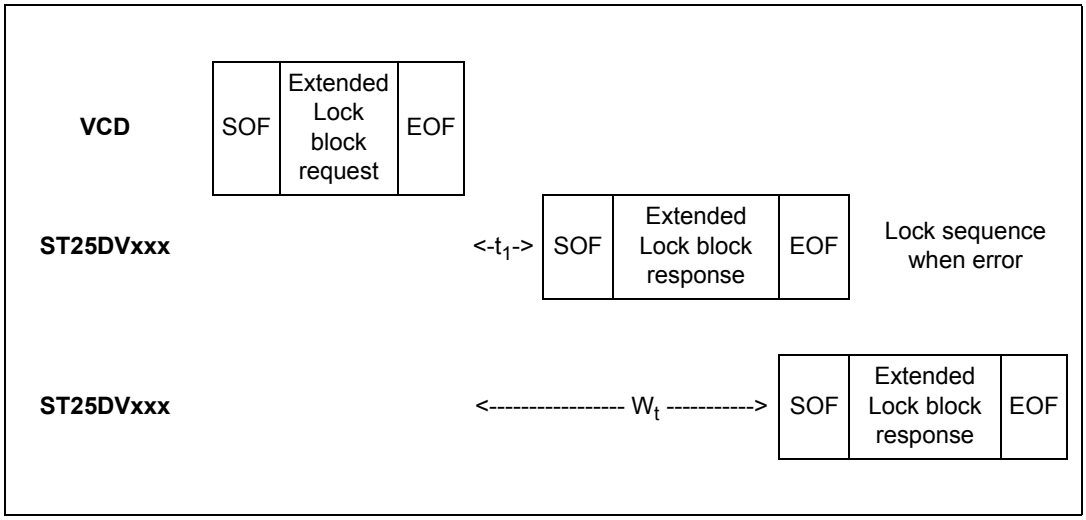
Table 88. Extended Lock block response format when Error_flag is set

Response SOF	Response_flags	Error code	CRC16	Response EOF
-	8 bits	8 bits	16 bits	-

Response parameter:

- Error code as Error_flag is set
  - 03h: command option not supported
  - 10h: block not available
  - 11h: the specified block is already locked and thus cannot be locked again
  - 14h: the specified block was not successfully locked

Figure 39. Extended Lock block frame exchange between VCD and ST25DVxxx



7.6.12 Read Multiple Blocks

When receiving the Read Multiple Block command, the ST25DVxxx reads the selected blocks and sends back their value in multiples of 32 bits in the response. The blocks are numbered from 00h to FFh in the request and the value is minus one (–1) in the field. For example, if the “Number of blocks” field contains the value 06h, seven blocks are read. The maximum number of blocks is fixed at 256 assuming that they are all located in the same area. If the number of blocks overlaps areas, the ST25DVxxx returns an error code. When the Option_flag is set, the response returns the Block Security Status.

Add: Block number is coded on 1 Byte and only first 256 blocks of ST25DV16K-xx and ST25DV64K-xx can be addressed using this command.

Table 89. Read Multiple Block request format

Request SOF	Request_ flags	Read Multiple Block	UID(1)	First block number	Number of blocks	CRC16	Request EOF
-	8 bits	23h	64 bits	8 bits	8 bits	16 bits	-

1. Gray color means that the field is optional.

Request parameters:

- Request flags
- UID (optional)
- First block number
- Number of blocks

Table 90. Read Multiple Block response format when Error_flag is NOT set

Response SOF	Response_ flags	Block security status(1)	Data	CRC16	Response EOF
-	8 bits	8 bits(2)	32 bits(2)	16 bits	-

1. Gray color means that the field is optional.
2. Repeated as needed.

Response parameters:

- Block security status if Option_flag is set (see *Table 91: Block security status*)
- N blocks of data

Table 91. Block security status

b_7	b_6	b_5	b_4	b_3	b_2	b_1	b_0
Reserved for future use. All at 0.							0: Current block not locked 1: Current block locked

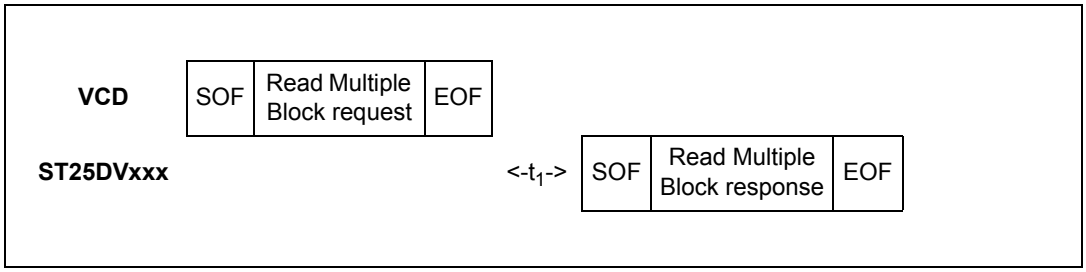
Table 92. Read Multiple Block response format when Error_flag is set

Response SOF	Response_flags	Error code	CRC16	Response EOF
-	8 bits	8 bits	16 bits	-

Response parameter:

- Error code as Error_flag is set:
  - 03h: command option is not supported
  - 0Fh: error with no information given
  - 10h: the specified block is not available
  - 15h: the specified block is read-protected

Figure 40. Read Multiple Block frame exchange between VCD and ST25DVxxx



7.6.13 Extended Read Multiple Blocks

When receiving the Extended Read multiple block command, the ST25DVxxx reads the selected blocks and sends back their value in multiples of 32 bits in the response. The blocks are numbered from 00h to last block of memory in the request and the value is minus one (-1) in the field. For example, if the “Number of blocks” field contains the value 06h, seven blocks are read. The maximum number of blocks is fixed at 2047 assuming that they are all located in the same area. If the number of blocks overlaps areas, the ST25DVxxx returns an error code. When the Option_flag is set, the response returns the Block Security Status.

Block number is coded on 2 Bytes so all memory blocks of ST25DV16K-xx and ST25DV64K-xx can be addressed using this command.

Table 93. Extended Read Multiple Block request format

Request SOF	Request_ flags	Extended Read Multiple Block	UID(1)	First block number	Number of blocks	CRC16	Request EOF
-	8 bits	33h	64 bits	16 bits	16 bits	16 bits	-

1. Gray color means that the field is optional.

Request parameters:

- Request flags
- UID (optional)
- First block number
- Number of blocks

Table 94. Extended Read Multiple Block response format when Error_flag is NOT set

Response SOF	Response_ flags	Block security status(1)	Data	CRC16	Response EOF
-	8 bits	8 bits(2)	32 bits(2)	16 bits	-

1. Gray color means that the field is optional.
2. Repeated as needed.

Response parameters:

- Block security status if Option_flag is set (see *Table 95: Block security status*)
- N blocks of data

Table 95. Block security status

b_7	b_6	b_5	b_4	b_3	b_2	b_1	b_0
Reserved for future use. All at 0							0: Current block not locked 1: Current block locked

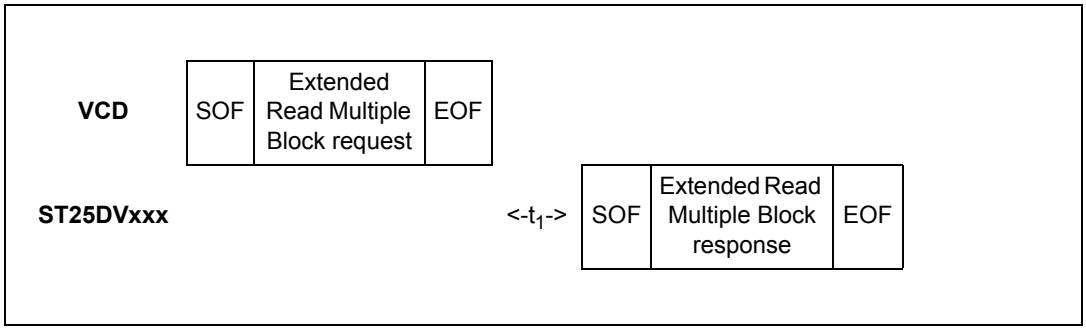
Table 96. Extended Read Multiple Block response format when Error_flag is set

Response SOF	Response_flags	Error code	CRC16	Response EOF
-	8 bits	8 bits	16 bits	-

Response parameter:

- Error code as Error_flag is set:
  - 03h: command option is not supported
  - 0Fh: error with no information given
  - 10h: the specified block is not available
  - 15h: the specified block is read-protected

Figure 41. Extended Read Multiple Block frame exchange between VCD and ST25DVxxx



7.6.14 Write Multiple Blocks

On receiving the Write Multiple Block command, the ST25DVxxx writes the data contained in the request to the requested blocks, and reports whether the write operation were successful in the response. ST25DVxxx supports up to 4 blocks, data field must be coherent with the number of blocks to program.

When the Option_flag is set, wait for EOF to respond. During the RF write cycle Wt, there should be no modulation (neither 100% nor 10%), otherwise the ST25DVxxx may not program correctly the data into the memory. The W_t time is equal to t_{1nom} + m × 302 µs < 20 ms. (m is an integer, it is function of Nb number of blocks to be programmed).

Block number is coded on 1 Byte and only first 256 blocks of ST25DV16K-xx and ST25DV64K-xx can be addressed using this command.

Table 97. Write Multiple Block request format

Request SOF	Request_flags	Write Multiple Block	UID(1)	First Block number	Number of block(2)	Data	CRC16	Request EOF
-	8 bits	24h	64 bits	8 bits	8 bits	Block length(3)	16 bits	-

1. Gray color means that the field is optional.
2. The number of blocks in the request is one less than the number of blocks that the VICC shall write.
3. Repeated as needed

Request parameters:

- Request flags
- UID (optional)
- First Block number
- Number of blocks
- Data

Table 98. Write Multiple Block response format when Error_flag is NOT set

Response SOF	Response_flags	CRC16	Response EOF
-	8 bits	16 bits	-

Response parameter:

- No parameter. The response is sent back after the writing cycle.

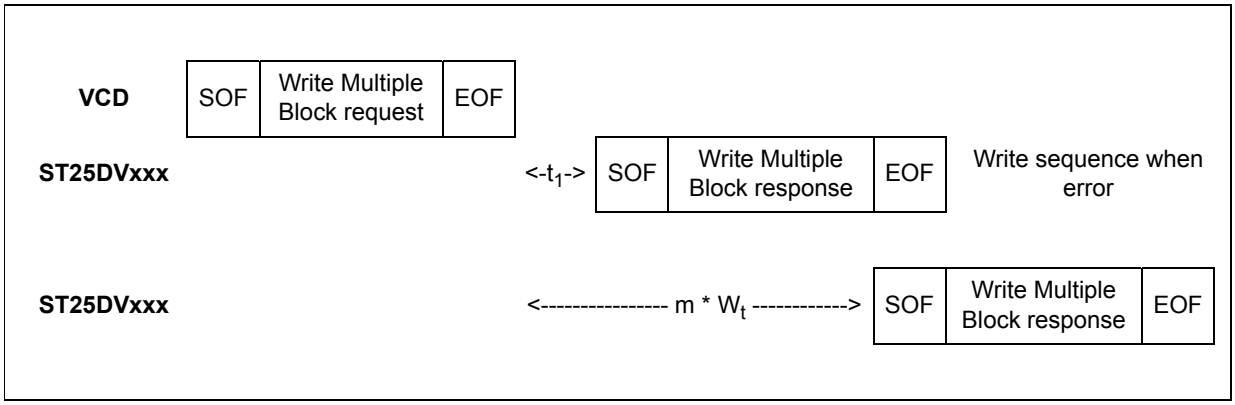
Table 99. Write Multiple Block response format when Error_flag is set

Response SOF	Response_flags	Error code	CRC16	Response EOF
-	8 bits	8 bits	16 bits	-

Response parameter:

- Error code as Error_flag is set:
  - 03h: command option is not supported
  - 0Fh: error with no information given
  - 10h: the specified block is not available
  - 12h: the specified block is locked and its contents cannot be changed
  - 13h: the specified block was not successfully programmed

Figure 42. Write Multiple Block frame exchange between VCD and ST25DVxxx



7.6.15 Extended Write Multiple Blocks

On receiving the Extended Write multiple block command, the ST25DVxxx writes the data contained in the request to the targeted blocks and reports whether the write operation were successful in the response. ST25DVxxx supports up to 4 blocks, data field must be coherent with number of blocks to program.

When the Option_flag is set, wait for EOF to respond. During the RF write cycle Wt, there should be no modulation (neither 100% nor 10%), otherwise the ST25DVxxx may not program correctly the data into the memory. The W_t time is equal to t_{1nom} + m × 302 µs < 20 ms (m is an integer function of Nb number of blocks to be programmed).

Block number is coded on 2 Bytes so all memory blocks of ST25DV16K-xx and ST25DV64K-xx can be addressed using this command.

Table 100. Extended Write Multiple Block request format

Request SOF	Request_flags	Extended Write multiple block	UID(1)	First Block number	Number of block(2)	Data	CRC16	Request EOF
-	8 bits	34h	64 bits	16 bits	16 bits	Block length(3)	16 bits	-

1. Gray color means that the field is optional.
2. The number of blocks in the request is one less than the number of blocks that the VICC shall write.
3. Repeated as needed

Request parameters:

- Request flags
- UID (optional)
- First block number
- Number of block
- Data (from first to last blocks, from LSB bytes to MSB bytes)

Table 101. Extended Write Multiple Block response format when Error_flag is NOT set

Response SOF	Response_flags	CRC16	Response EOF
-	8 bits	16 bits	-

Response parameter:

- No parameter. The response is sent back after the writing cycle.

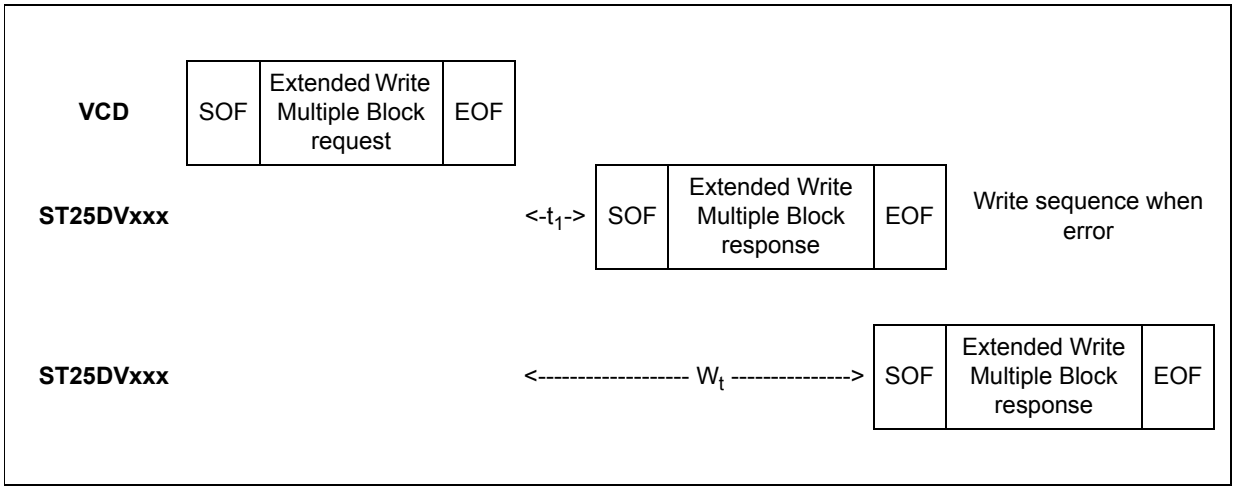
Table 102. Extended Write Multiple Block response format when Error_flag is set

Response SOF	Response_flags	Error code	CRC16	Response EOF
-	8 bits	8 bits	16 bits	-

Response parameter:

- Error code as Error_flag is set:
  - 03h: command option is not supported
  - 0Fh: error with no information given
  - 10h: the specified block is not available
  - 12h: the specified block is locked and its contents cannot be changed
  - 13h: the specified block was not successfully programmed

Figure 43. Extended Write Multiple Block frame exchange between VCD and ST25DVxxx



7.6.16 Select

When receiving the Select command:

- If the UID is equal to its own UID, the ST25DVxxx enters or stays in the Selected state and sends a response.
- If the UID does not match its own UID, the selected ST25DVxxx returns to the Ready state and does not send a response.

The ST25DVxxx answers an error code only if the UID is equal to its own UID. If not, no response is generated. If an error occurs, the ST25DVxxx remains in its current state.

Table 103. Select request format

Request SOF	Request_flags	Select	UID	CRC16	Request EOF
-	8 bits	25h	64 bits	16 bits	-

Request parameter:

- UID

Table 104. Select Block response format when Error_flag is NOT set

Response SOF	Response_flags	CRC16	Response EOF
-	8 bits	16 bits	-

Response parameter:

- No parameter

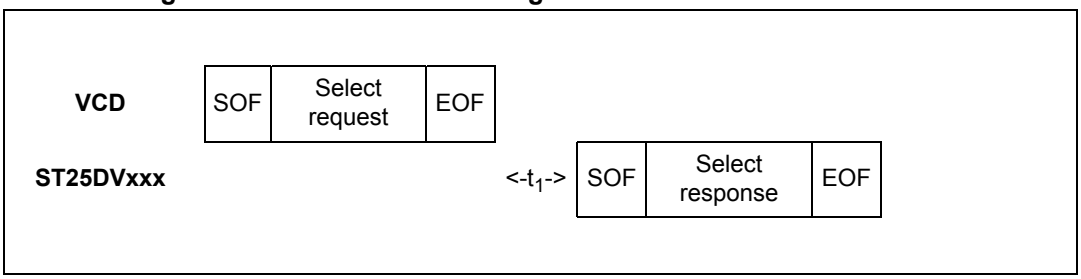
Table 105. Select response format when Error_flag is set

Response SOF	Response_flags	Error code	CRC16	Response EOF
-	8 bits	8 bits	16 bits	-

Response parameter:

- Error code as Error_flag is set:
  - 03h: the option is not supported
  - 0Fh: error with no information given

Figure 44. Select frame exchange between VCD and ST25DVxxx



7.6.17 Reset to Ready

On receiving a Reset to Ready command, the ST25DVxxx returns to the Ready state if no error occurs. In the Addressed mode, the ST25DVxxx answers an error code only if the UID is equal to its own UID. If not, no response is generated.

Table 106. Reset to Ready request format

Request SOF	Request_flags	Reset to Ready	UID(1)	CRC16	Request EOF
-	8 bits	26h	64 bits	16 bits	-

1. Gray color means that the field is optional.

Request parameter:

- UID (optional)

Table 107. Reset to Ready response format when Error_flag is NOT set

Response SOF	Response_flags	CRC16	Response EOF
-	8 bits	16 bits	-

Response parameter:

- No parameter

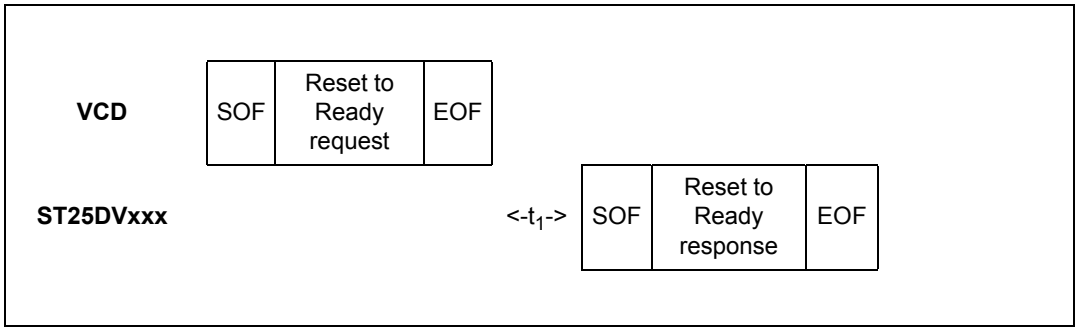
Table 108. Reset to ready response format when Error_flag is set

Response SOF	Response_flags	Error code	CRC16	Response EOF
-	8 bits	8 bits	16 bits	-

Response parameter:

- Error code as Error_flag is set:
  - 03h: the option is not supported
  - 0Fh: error with no information given

Figure 45. Reset to Ready frame exchange between VCD and ST25DVxxx



7.6.18 Write AFI

On receiving the Write AFI request, the ST25DVxxx programs the 8-bit AFI value to its memory. When the Option_flag is set, wait for EOF to respond.

During the RF write cycle W_t, there should be no modulation (neither 100% nor 10%), otherwise the ST25DVxxx may not write correctly the AFI value into the memory. The W_t time is equal to t_{1nom} + N × 302 µs (N is an integer).

Table 109. Write AFI request format

Request SOF	Request_flags	Write AFI	UID(1)	AFI	CRC16	Request EOF
-	8 bits	27h	64 bits	8 bits	16 bits	-

1. Gray color means that the field is optional.

Request parameter:

- Request flags
- UID (optional)
- AFI

Table 110. Write AFI response format when Error_flag is NOT set

Response SOF	Response_flags	CRC16	Response EOF
-	8 bits	16 bits	-

Response parameter:

- No parameter

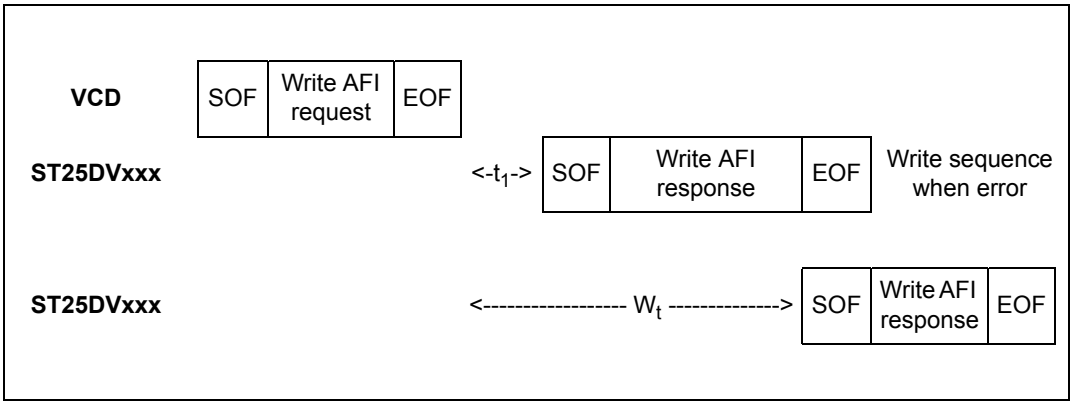
Table 111. Write AFI response format when Error_flag is set

Response SOF	Response_ flags	Error code	CRC16	Response EOF
-	8 bits	8 bits	16 bits	-

Response parameter:

- Error code as Error_flag is set
  - 03h: command option is not supported
  - 0Fh: error with no information given
  - 12h: the specified block is locked and its contents cannot be changed
  - 13h: the specified block was not successfully programmed

Figure 46. Write AFI frame exchange between VCD and ST25DVxxx



7.6.19 Lock AFI

On receiving the Lock AFI request, the ST25DVxxx locks the AFI value permanently. When the Option_flag is set, wait for EOF to respond.

During the RF write cycle W_t, there should be no modulation (neither 100% nor 10%), otherwise the ST25DVxxx may not lock correctly the AFI value in memory. The W_t time is equal to t_{1nom} + 18 × 302 µs.

Table 112. Lock AFI request format

Request SOF	Request_flags	Lock AFI	UID(1)	CRC16	Request EOF
-	8 bits	28h	64 bits	16 bits	-

1. Gray color means that the field is optional.

Request parameter:

- Request Flags
- UID (optional)

Table 113. Lock AFI response format when Error_flag is NOT set

Response SOF	Response_flags	CRC16	Response EOF
-	8 bits	16 bits	-

Response parameter:

- No parameter

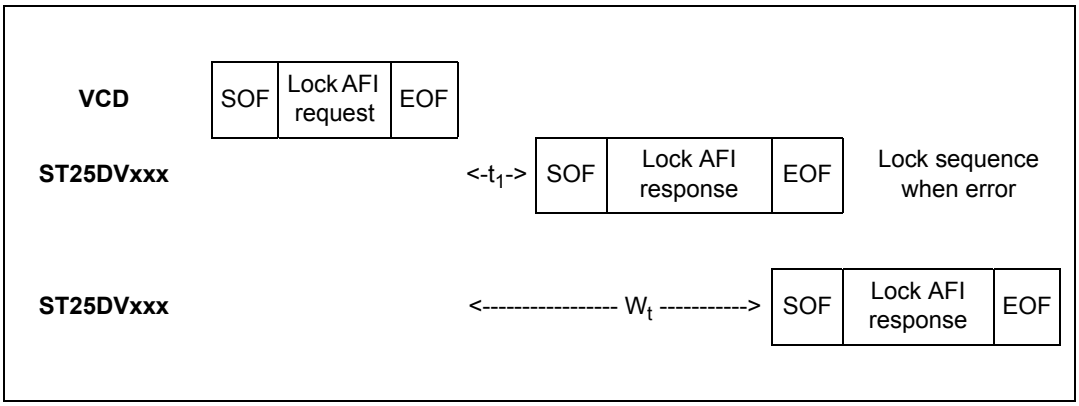
Table 114. Lock AFI response format when Error_flag is set

Response SOF	Response_flags	Error code	CRC16	Response EOF
-	8 bits	8 bits	16 bits	-

Response parameter:

- Error code as Error_flag is set
  - 03h: command option is not supported
  - 0Fh: error with no information given
  - 11h: the specified block is already locked and thus cannot be locked again
  - 14h: the specified block was not successfully locked

Figure 47. Lock AFI frame exchange between VCD and ST25DVxxx



7.6.20 Write DSFID

On receiving the Write DSFID request, the ST25DVxxx programs the 8-bit DSFID value to its memory. When the Option_flag is set, wait for EOF to respond.

During the RF write cycle W_t, there should be no modulation (neither 100% nor 10%), otherwise the ST25DVxxx may not write correctly the DSFID value in memory. The W_t time is equal to t_{1nom} + N × 302 µs (N is an integer).

Table 115. Write DSFID request format

Request SOF	Request_flags	Write DSFID	UID(1)	DSFID	CRC16	Request EOF
-	8 bits	29h	64 bits	8 bits	16 bits	-

1. Gray color means that the field is optional.

Request parameter:

- Request flags
- UID (optional)
- DSFID

Table 116. Write DSFID response format when Error_flag is NOT set

Response SOF	Response_flags	CRC16	Response EOF
-	8 bits	16 bits	-

Response parameter:

- No parameter

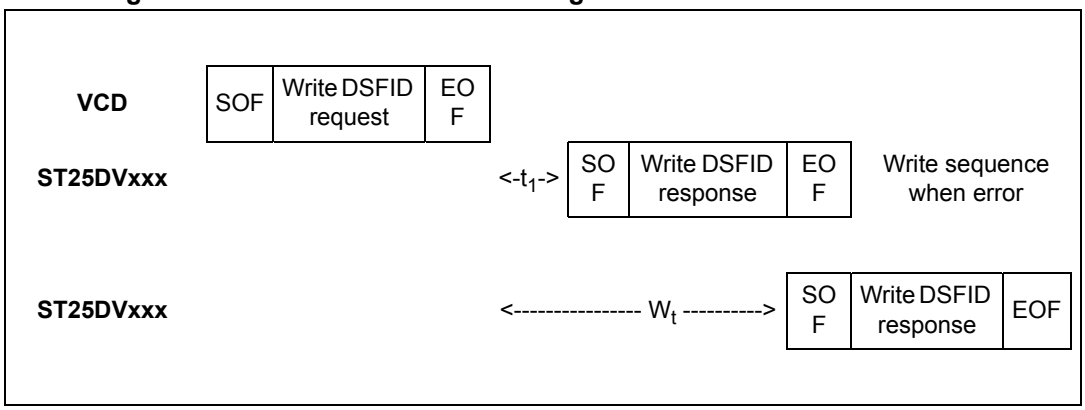
Table 117. Write DSFID response format when Error_flag is set

Response SOF	Response_flags	Error code	CRC16	Response EOF
-	8 bits	8 bits	16 bits	-

Response parameter:

- Error code as Error_flag is set
  - 03h: command option is not supported
  - 0Fh: error with no information given
  - 12h: the specified block is locked and its contents cannot be changed
  - 13h: the specified block was not successfully programmed

Figure 48. Write DSFID frame exchange between VCD and ST25DVxxx



7.6.21 Lock DSFID

On receiving the Lock DSFID request, the ST25DVxxx locks the DSFID value permanently. When the Option_flag is set, wait for EOF to respond.

During the RF write cycle W_t, there should be no modulation (neither 100% nor 10%), otherwise the ST25DVxxx may not lock correctly the DSFID value in memory. The W_t time is equal to t_{1nom} + N × 302 µs (N is an integer).

Table 118. Lock DSFID request format

Request SOF	Request_flags	Lock DSFID	UID(1)	CRC16	Request EOF
-	8 bits	2Ah	64 bits	16 bits	-

1. Gray color means that the field is optional.

Request parameter:

- Request flags
- UID (optional)

Table 119. Lock DSFID response format when Error_flag is NOT set

Response SOF	Response_flags	CRC16	Response EOF
-	8 bits	16 bits	-

Response parameter:

- No parameter.

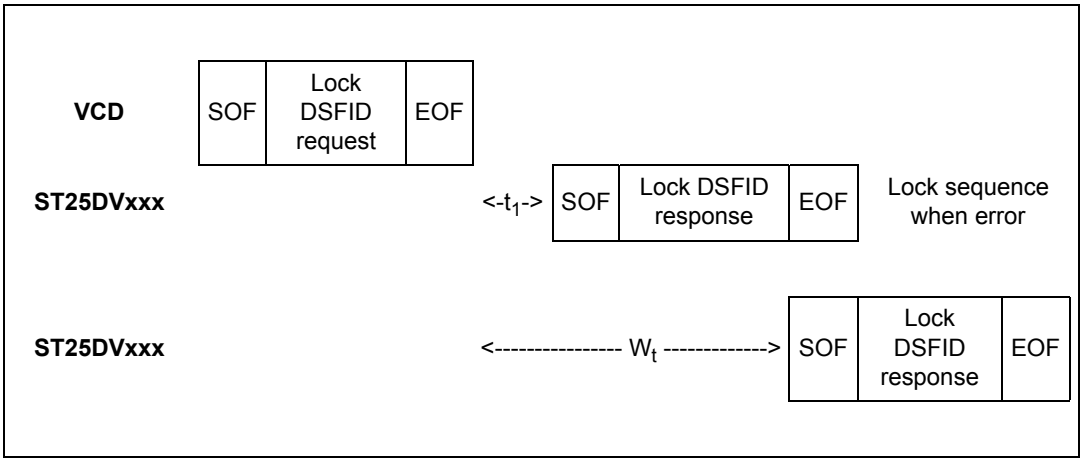
Table 120. Lock DSFID response format when Error_flag is set

Response SOF	Response_flags	Error code	CRC16	Response EOF
-	8 bits	8 bits	16 bits	-

Response parameter:

- Error code as Error_flag is set:
  - 03h: command option is not supported
  - 0Fh: error with no information given
  - 11h: the specified block is already locked and thus cannot be locked again
  - 14h: the specified block was not successfully locked

Figure 49. Lock DSFID frame exchange between VCD and ST25DVxxx



7.6.22 Get System Info

When receiving the Get System Info command, the ST25DVxxx sends back its information data in the response. The Option_flag is not supported. The Get System Info can be issued in both Addressed and Non Addressed modes.

Table 121. Get System Info request format

Request SOF	Request_flags	Get System Info	UID(1)	CRC16	Request EOF
-	8 bits	2Bh	64 bits	16 bits	-

1. Gray color means that the field is optional.

Request parameter:

- Request flags
- UID (optional)

Table 122. Get System Info response format Error_flag is NOT set

Device	Response SOF	Response flags	Information flags	UID	DSFID	AFI	Mem. Size	IC ref.	CRC16	Response EOF
ST25DV64K-xx ST25DV16K-xx	-	00h	0Bh	64 bits	8 bits	8 bits	NA(1)	26h	16 bits	-
ST25DV04K-xx			0Fh				7Fh	24h		

1. Field not present in this configuration

Response parameters:

- Information flags set to 0Bh/0Fh. DSFID, AFI and IC reference fields are present.
- UID code on 64 bits
- DSFID value
- AFI value
- MemSize: memory size in the number of blocks (only present for ST25DV04K-xx configurations)
- ST25DVxxx IC reference: the 8 bits are significant.

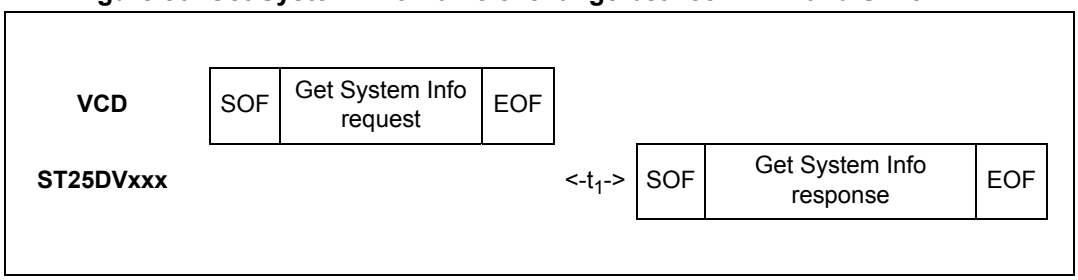
Table 123. Get System Info response format when Error_flag is set

Response SOF	Response_flags	Error code	CRC16	Response EOF
-	01h	8 bits	16 bits	-

Response parameter:

- Error code as Error_flag is set:
  - 03h: Option not supported
  - 0Fh: error with no information given

Figure 50. Get System Info frame exchange between VCD and ST25DVxxx



7.6.23 Extended Get System Info

When receiving the Extended Get System Info command, the ST25DVxxx sends back its information data in the response. The Option_flag is not supported. The Extended Get System Info can be issued in both Addressed and Non Addressed modes.

Table 124. Extended Get System Info request format

Request SOF	Request_flags	Extended Get System Info	Parameter request field	UID(1)	CRC16	Request EOF
-	8 bits	3Bh	8 bits	64 bits	16 bits	-

1. Gray color means that the field is optional.

- Request flags
- Request parameters
- UID (optional)

Table 125. Parameter request list

Bit	Flag name	Value	Description
b1	DSFID	0	No request of DSFID
		1	Request of DSFID
b2	AFI	0	No request of AFI
		1	Request of AFI
b3	VICC memory size	0	No request of data field on VICC memory size
		1	Request of data field on VICC memory size
b4	IC reference	0	No request of Information on IC reference
		1	Request of Information on IC reference
b5	MOI	1	Information on MOI always returned in response flag
b6	VICC Command list	0	No request of Data field of all supported commands
		1	Request of Data field of all supported commands
b7	CSI Information	0	No request of CSI list
		1	Request of CSI list
b8	Extended Get System Info parameter Field	0	One byte length of Extended Get System Info parameter field

Table 126. Extended Get System Info response format when Error_flag is NOT set

Response SOF	Response_flags	Information flags	UID	DSFID(1)(2)	AFI(1)(2)	Other Field(1)(2)	CRC16	Response EOF
-	00h	8 bits(2)	64 bits	8 bits	8 bits	up to 64 bits(3)	16 bits	-

1. Gray color means that the field is optional.
2. See *Table 127: Response Information Flag*.

3. Number of bytes is function of parameter list selected.

Response parameters:

- Information flag defining which fields are present
- UID code on 64 bits
- DSFID value (if requested in Parameters request field)
- AFI value (if requested in Parameters request field)
- Other fields:
  - VICC Memory size (if requested in Parameters request field)
  - ICRef (if requested in Parameters request field)
  - VICC Command list (if requested in Parameters request field)

Table 127. Response Information Flag

Bit	Flag name	Value	Description
b1	DSFID	0	DSFID field is not present
		1	DSFID field is present
b2	AFI	0	AFI field is not present
		1	AFI field is present
b3	VICC memory size	0	Data field on VICC memory size is not present.
		1	Data field on VICC memory size is present.
b4	IC reference	0	Information on IC reference field is not present.
		1	Information on IC reference field is present.
b5	MOI	0	1 byte addressing
		1	2 byte addressing
b6	VICC Command list	0	Data field of all supported commands is not present
		1	Data field of all supported commands is present
b7	CSI Information	0	CSI list is not present
b8	Info flag Field	0	One byte length of Info flag field

Table 128. Response other field: ST25DVxxx VICC memory size

MSB				**LSB**
24	**22**	**21**	**17**	**16** **01**
RFU		Block size in byte		Number of blocks
0h		03h		07FFh (ST25DV64K-xx) 01FFh (ST25DV16K-xx) 007Fh (ST25DV04K-xx)

Table 129. Response other field: ST25DVxxx IC Ref

1 byte
ICRef
24h (ST25DV04K-XX) or 26h (ST25DV16K-xx and ST25DV64K-xx)

Table 130. Response other field: ST25DVxxx VICC command list

MSB			LSB
32 25	24 17	16 09	08 01
Byte 4	Byte3	Byte 2	Byte 1
00h	3Fh	3Fh	FFh

Table 131. Response other field: ST25DVxxx VICC command list Byte 1

Bit	Meaning if bit is set	Comment
b1	Read single block is supported	-
b2	Write single block is supported	-
b3	Lock single block is supported	-
b4	Read multiple block is supported	-
b5	Write multiple block is supported	-
b6	Select is supported	including Select state
b7	Reset to Ready is supported	-
b8	Get multiple block security status is supported	-

Table 132. Response other field: ST25DVxxx VICC command list Byte 2

Bit	Meaning if bit is set	Comment
b1	Write AFI is supported	-
b2	Lock AFI is supported	-
b3	Write DSFID is supported	-
b4	Lock DSFID is supported	-
b5	Get System Information is supported	-
b6	Custom commands are supported	-
b7	RFU	0 shall be returned
b8	RFU	0 shall be returned

Table 133. Response other field: ST25DVxxx VICC command list Byte 3

Bit	Meaning if bit is set	Comment
b1	Extended read single block is supported	-
b2	Extended write single block is supported	-
b3	Extended lock single block is supported	-
b4	Extended read multiple block is supported	-
b5	Extended write multiple block is supported	-
b6	Extended Get Multiple Security Status is supported	-
b7	RFU	0 shall be returned
b8	RFU	0 shall be returned

Table 134. Response other field: ST25DVxxx VICC command list Byte 4

Bit	Meaning if bit is set	Comment
b1	Read Buffer is supported	Means Response Buffer is supported
b2	Select Secure State is supported	Means VCD or Mutual authentication are supported
b3	Final Response always includes crypto result	Means that flag b3 will be set in the Final response
b4	AuthComm crypto format is supported	-
b5	SecureComm crypto format is supported	-
b6	KeyUpdate is supported	-
b7	Challenge is supported	-
b8	If set to 1 a further Byte is transmitted	0 shall be returned

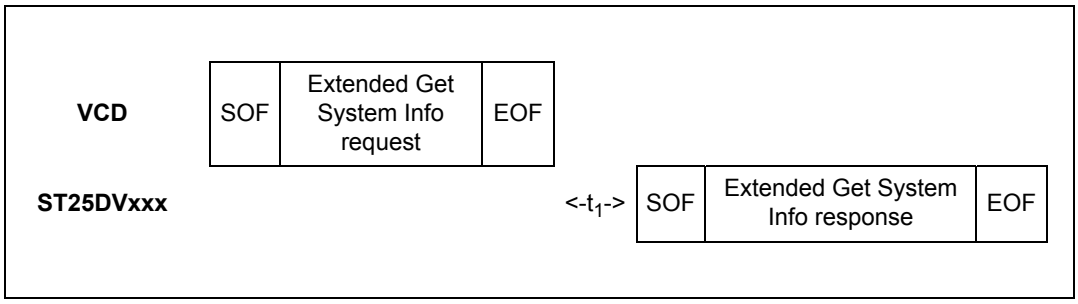
Table 135. Extended Get System Info response format when Error_flag is set

Response SOF	Response_flags	Error code	CRC16	Response EOF
-	01h	8 bits	16 bits	-

Response parameter:

- Error code as Error_flag is set:
  - 03h: Option not supported
  - 0Fh: error with no information given

Figure 51. Extended Get System Info frame exchange between VCD and ST25DVxxx



7.6.24 Get Multiple Block Security Status

When receiving the Get Multiple Block Security Status command, the ST25DVxxx sends back its security status for each address block: 0 when block is writable else 1 when block is locked for writing. The blocks security status are defined by the area security status (and by LCK_CCFILE register for blocks 0 and 1). The blocks are numbered from 00h up to the maximum memory block number in the request, and the value is minus one (–1) in the field. For example, a value of “06” in the “Number of blocks” field requests will return the security status of seven blocks. This command does not respond an error if number of blocks overlap areas.

The number of blocks is coded on 1 Byte and only first 256 blocks of ST25DV16K-xx and ST25DV64K-xx can be addressed using this command.

Table 136. Get Multiple Block Security Status request format

Request SOF	Request _flags	Get Multiple Block Security Status	UID(1)	First block number	Number of blocks	CRC16	Request EOF
-	8 bits	2Ch	64 bits	8 bits	8 bits	16 bits	-

1. Gray color means that the field is optional.

Request parameter:

- Request flags
- UID (optional)
- First block number
- Number of blocks

Table 137. Get Multiple Block Security Status response format when Error_flag is NOT set

Response SOF	Response_flags	Block security status	CRC16	Response EOF
-	8 bits	8 bits(1)	16 bits	-

1. Repeated as needed.

Response parameters:

- Block security status

Table 138. Block security status

b_7	b_6	b_5	b_4	b_3	b_2	b_1	b_0
Reserved for future use All at 0							0: Current block not locked 1: Current block locked

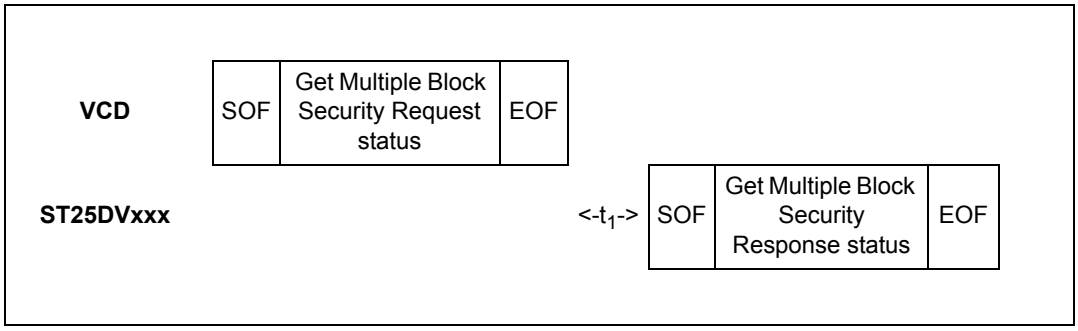
Table 139. Get Multiple Block Security Status response format when Error_flag is set

Response SOF	Response_flags	Error code	CRC16	Response EOF
-	8 bits	8 bits	16 bits	-

Response parameter:

- Error code as Error_flag is set:
  - 03h: the option is not supported
  - 0Fh: error with no information given
  - 10h: the specified block is not available

Figure 52. Get Multiple Block Security Status frame exchange between VCD and ST25DVxxx



7.6.25 Extended Get Multiple Block Security Status

When receiving the Extended Get Multiple Block Security Status command, the ST25DVxxx sends back the security status for each address block: 0 when the block is writable else 1 when block is locked for writing. The block security statuses are defined by the area security status. The blocks are numbered from 00h up to the maximum memory block number in the request, and the value is minus one (–1) in the field. For example, a value of '06' in the “Number of blocks” field requests to return the security status of seven blocks.

This command does not respond an error if number of blocks overlap areas.

The number of blocks is coded on 2 Bytes so all memory blocks of ST25DV16K-xx and ST25DV64K-xx can be addressed using this command.

Table 140. Extended Get Multiple Block Security Status request format

Request SOF	Request _flags	Extended Get Multiple Block Security Status	UID(1)	First block number	Number of blocks	CRC16	Request EOF
-	8 bits	3Ch	64 bits	16 bits	16 bits	16 bits	-

1. Gray color means that the field is optional.

Request parameter:

- Request flags
- UID (optional)
- First block number
- Number of blocks

Table 141. Extended Get Multiple Block Security Status response format when Error_flags NOT set

Response SOF	Response_flags	Block security status	CRC16	Response EOF
-	8 bits	8 bits(1)	16 bits	-

1. Repeated as needed.

Response parameters:

- Block security status

Table 142. Block security status

b_7	b_6	b_5	b_4	b_3	b_2	b_1	b_0
Reserved for future use All at 0							0: Current block not locked 1: Current block locked

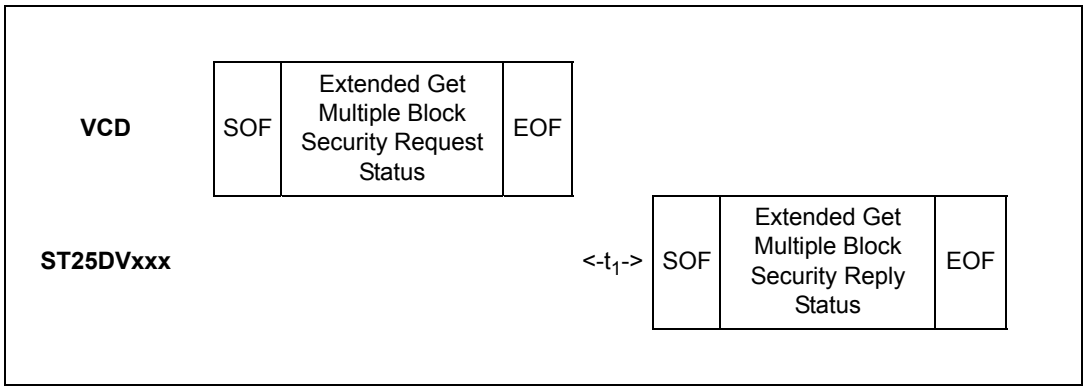
Table 143. Extended Get Multiple Block Security Status response format when Error_flag is set

Response SOF	Response_flags	Error code	CRC16	Response EOF
-	8 bits	8 bits	16 bits	-

Response parameter:

- Error code as Error_flag is set:
  - 03h: the option is not supported
  - 0Fh: error with no information given
  - 10h: the specified block is not available

Figure 53. Extended Get Multiple Block Security Status frame exchange between VCD and ST25DVxxx



7.6.26 Read Configuration

On receiving the Read Configuration command, the ST25DVxxx reads the static system configuration register at the Pointer address and sends back its 8-bit value in the response.

The Option_flag is not supported. The Inventory_flag must be set to 0.

Table 144. Read Configuration request format

Request SOF	Request_flags	Read Configuration	IC Mfg code	UID(1)	Pointer	CRC16	Request EOF
-	8 bits	A0h	02h	64 bits	8 bits	16 bits	-

1. Gray color means that the field is optional.

Note: Please refer to Table 9: System configuration memory map for details on register addresses.

Request parameters:

- System configuration register pointer
- UID (optional)

Table 145. Read Configuration response format when Error_flag is NOT set

Response SOF	Response_flags	Register value	CRC16	Response EOF
-	8 bits	8 bits	16 bits	-

Response parameters:

- One byte of data: system configuration register

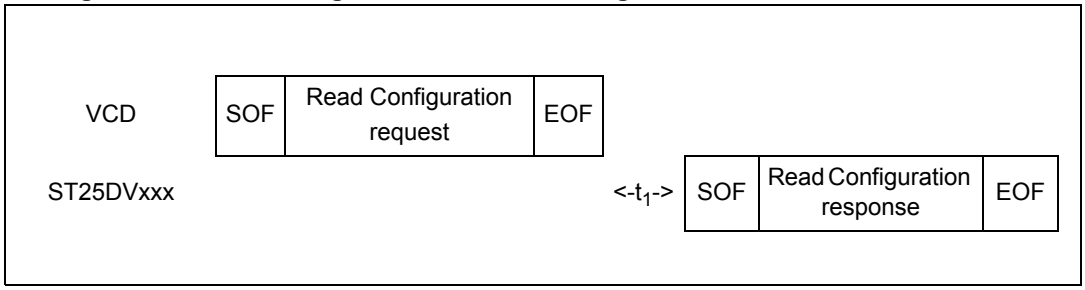
Table 146. Read Configuration response format when Error_flag is set

Response SOF	Response_flags	Error code	CRC16	Response EOF
-	8 bits	8 bits	16 bits	-

Response parameter:

- Error code as Error_flag is set
  - 02h: command not recognized
  - 03h: the option is not supported
  - 10h: block not available
  - 0Fh: error with no information given

Figure 54. Read Configuration frame exchange between VCD and ST25DVxxx



7.6.27 Write Configuration

The Write Configuration command is used to write static system configuration register. The Write Configuration must be preceded by a valid presentation of the RF configuration password (00) to open the RF configuration security session.

On receiving the Write Configuration command, the ST25DVxxx writes the data contained in the request to the system configuration register at the Pointer address and reports whether the write operation was successful in the response or not.

When the Option_flag is set, wait for EOF to respond. The Inventory_flag is not supported.

During the RF write cycle W_t, there should be no modulation (neither 100% nor 10%), otherwise the ST25DVxxx may not program correctly the data into the Configuration byte. The W_t time is equal to t_{1nom} + N × 302 µs (N is an integer).

Table 147. Write Configuration request format

Request SOF	Request_ flags	Write Configuration	IC Mfg code	UID(1)	Pointer	Register Value(2)	CRC16	Request EOF
-	8 bits	A1h	02h	64 bits	8 bits	8 bits	16 bits	-

1. Gray color means that the field is optional.
2. Before updating the register value, check the meaning of each bit in previous sections.

Request parameters:

- Request flags
- Register pointer
- Register value
- UID (optional)

Table 148. Write Configuration response format when Error_flag is NOT set

Response SOF	Response_flags	CRC16	Response EOF
-	8 bits	16 bits	-

Note: Please refer to Table 9: System configuration memory map for details on register addresses.

Response parameter:

- No parameter. The response is sent back after the writing cycle.

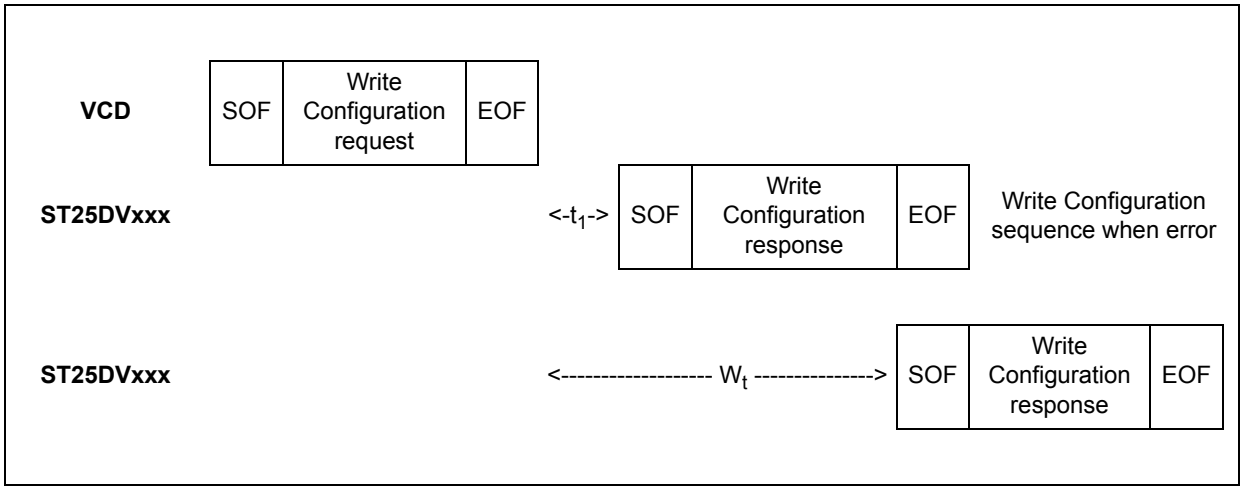
Table 149. Write Configuration response format when Error_flag is set

Response SOF	Response_flags	Error code	CRC16	Response EOF
-	8 bits	8 bits	16 bits	-

Response parameter:

- Error code as Error_flag is set:
  - 02h: command not recognized
  - 03h: command option is not supported
  - 0Fh: error with no information given
  - 10h: block not available
  - 11h: block is already locked, can't lock again
  - 12h: block already locked, content can't change
  - 13h: the specified block was not successfully programmed
  - -14h: block was not successfully locked

Figure 55. Write Configuration frame exchange between VCD and ST25DVxxx



7.6.28 Read Dynamic Configuration

On receiving the Read Dynamic Configuration command, the ST25DVxxx reads the Dynamic register address indicated by the pointer and sends back its 8-bit value in the response.

The Option_flag is not supported. The Inventory_flag must be set to 0.

Table 150. Read Dynamic Configuration request format

Request SOF	Request_flags	Read Dynamic Configuration	IC Mfg code	UID(1)	Pointer address	CRC16	Request EOF
-	8 bits	ADh	02h	64 bits	8 bits	16 bits	-

1. Gray color means that the field is optional.

Request parameters:

- UID (optional)

Table 151. Read Dynamic Configuration response format when Error_flag is NOT set

Response SOF	Response_flags	Data	CRC16	Response EOF
-	8 bits	8 bits	16 bits	-

Response parameters:

- One byte of data

Note: Please refer to Table 9: System configuration memory map for details on register addresses.

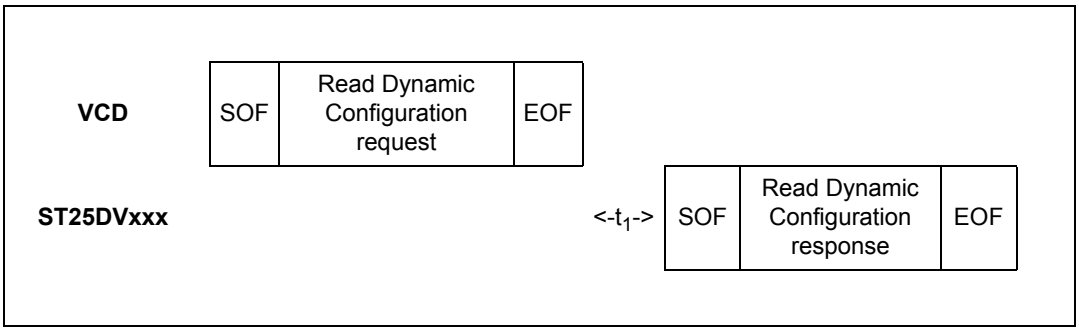
Table 152. Read Dynamic Configuration response format when Error_flag is set

Response SOF	Response_flags	Error code	CRC16	Response EOF
-	8 bits	8 bits	16 bits	-

Response parameter:

- Error code as Error_flag is set
  - 02h: command not recognized
  - 03h: the option is not supported
  - 0Fh: error with no information given
  - 10h: block not available

Figure 56. Read Dynamic Configuration frame exchange between VCD and ST25DVxxx



7.6.29 Write Dynamic Configuration

On receiving the Write Dynamic Configuration command, the ST25DVxxx updates the Dynamic register addressed by the pointer.

The Option_flag is not supported. The Inventory_flag must be set to 0.

Table 153. Write Dynamic Configuration request format

Request SOF	Request_flags	Write Dynamic Configuration	IC Mfg code	UID(1)	Pointer address	Register Value	CRC16	Request EOF
-	8 bits	AEh	02h	64 bits	8 bits	8 bits	16 bits	-

1. Gray color means that the field is optional.

Request parameters:

- Request flags
- UID (optional)
- Pointer address
- Register value

Table 154. Write Dynamic Configuration response format when Error_flag is NOT set

Response SOF	Response_flags	CRC16	Response EOF
-	8 bits	16 bits	-

Response parameter:

- No parameter. The response is sent back after t_1.

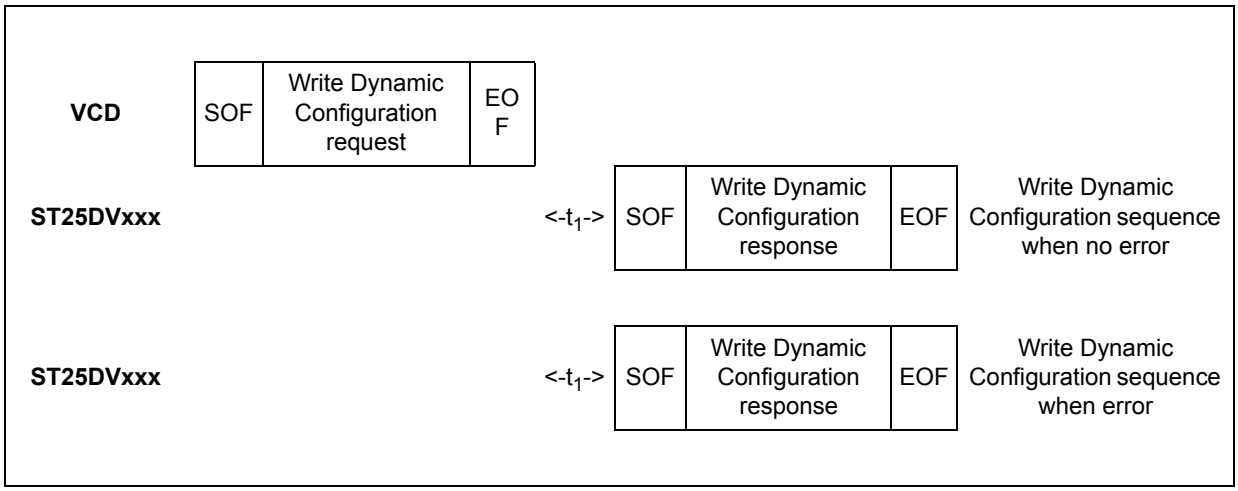
Table 155. Write Dynamic Configuration response format when Error_flag is set

Response SOF	Response_flags	Error code	CRC16	Response EOF
-	8 bits	8 bits	16 bits	-

Response parameter:

- Error code as Error_flag is set:
  - 02h: command not recognized
  - 03h: the option is not supported
  - 0Fh: error with no information given
  - 10h: block not available

Figure 57. Write Dynamic Configuration frame exchange between VCD and ST25DVxxx



7.6.30 Manage GPO

On receiving the Manage GPO command. Depending on the command argument, the ST25DV force the GPO output level if RF_USER interrupt is enabled, or send a pulse on GPO output if RF_INTERRUPT is enabled. If neither RF_USER nor RF_INTERRUPT was enabled, the command is not executed and ST25DVxxx responds an Error code “0F”.

The IT duration is defined by IT_TIME register and occurs just after the command response.

For the ST25DVxx-JF (CMOS output), a set means that the GPO pin is driven to a High level (V_{DCG}) and a Reset pulls the GPO pin to a low level (V_{SS}).

The IT corresponds to a transmission of a positive pulse on the GPO pin.

For the ST25DVxx-IE (Open Drain output), a Set means that the GPO pin is driven to a low level (V_{SS}) and a Reset releases the GPO (High impedance).

IT corresponds to the GPO pin driven to ground during the IT duration, then pin is released. Thanks to an external pull up, the high level will be recovered.

Option_flag is not supported. The Inventory_flag must be set to 0.

Table 156. ManageGPO request format

Request SOF	Request_ flags	ManageGPO	IC Mfg code	UID(1)	GPO VAL(2)	CRC16	Request EOF
-	8 bits	A9h	02h	64 bits	8 bits	16 bits	-

1. Gray color means that the field is optional.
2. See *Table 157: GPOVAL*

Table 157. GPOVAL

GPOVAL	IT	ST25DVxx-IE (OD)	ST25DVxx-JF (CMOS)
0xxxxxx0b	RF_USER enabled	Pin pull to 0	GPO Pin set to logic One (V_{DCG})
0xxxxxx1b	RF_USER enabled	Pin released (HZ)	GPO Pin reset to logic zero
1xxxxxxxb	RF_INTERRUPT enabled	GPO pin pulled to 0 during IT Time then released (HZ)	GPO Pin drives a positive pulse
Any other conditions		GPO realeased (Hz)	GPO pin reset to logic zero

Request parameters:

- Request flag
- UID (optional)
- Data: Define static or dynamic Interrupt

Table 158. ManageGPO response format when Error_flag is NOT set

Response SOF	Response_flags	CRC16	Response EOF
-	8 bits	16 bits	-

Response parameter:

- No parameter. The response is sent back after the write cycle.

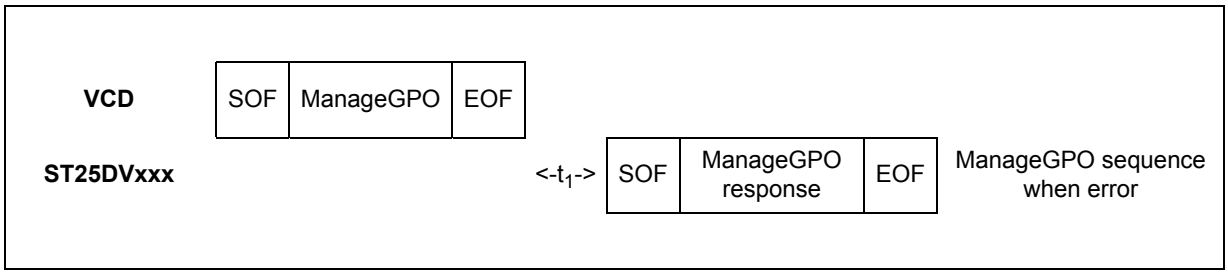
Table 159. ManageGPO response format when Error_flag is set

Response SOF	Response_flags	Error code	CRC16	Response EOF
-	8 bits	8 bits	16 bits	-

Response parameter:

- Error code as Error_flag is set:
  - 02h: command not recognized
  - 13h: the specified block was not successfully programmed

Figure 58. ManageGPO frame exchange between VCD and ST25DVxxx



7.6.31 Write Message

On receiving the Write Message command, the ST25DVxxx puts the data contained in the request into the Mailbox buffer, update the MB_LEN_Dyn register, and set bit RF_PUT_MSG in MB_CTRL_Dyn register. It then reports if the write operation was successful in the response. The ST25DVxxx Mailbox contains up to 256 data bytes which are filled from the first location '00'. MSGlength parameter of the command is the number of Data bytes minus - 1 (00 for 1 byte of data, FFh for 256 bytes of data). Write Message could be executed only when Mailbox is accessible by RF (Fast Transfer Mode is enabled, previous RF message was read or time-out occurs, no I^2C message to be read). User can check it by reading b1 of MB_CTRL_Dyn “HOST_PUT_MSG” which must be reset to “0”. (refer to *Section 5.1: Fast transfer mode (FTM)*)

Table 160. Write Message request format

Request SOF	Request_ flags	Write Message	IC Mfg code	UID(1)	MSGLength	Message Data	CRC16	Request EOF
-	8 bits	AAh	02h	64 bits	1 byte	(MSGLength + 1) bytes	16 bits	-

1. Gray color means that the field is optional.

Request parameters:

- Request flags
- UID (optional)
- Message Length
- Message Data

Table 161. Write Message response format when Error_flag is NOT set

Response SOF	Response_flags	CRC16	Response EOF
-	8 bits	16 bits	-

Response parameter:

- No parameter. The response is sent back after the write cycle.

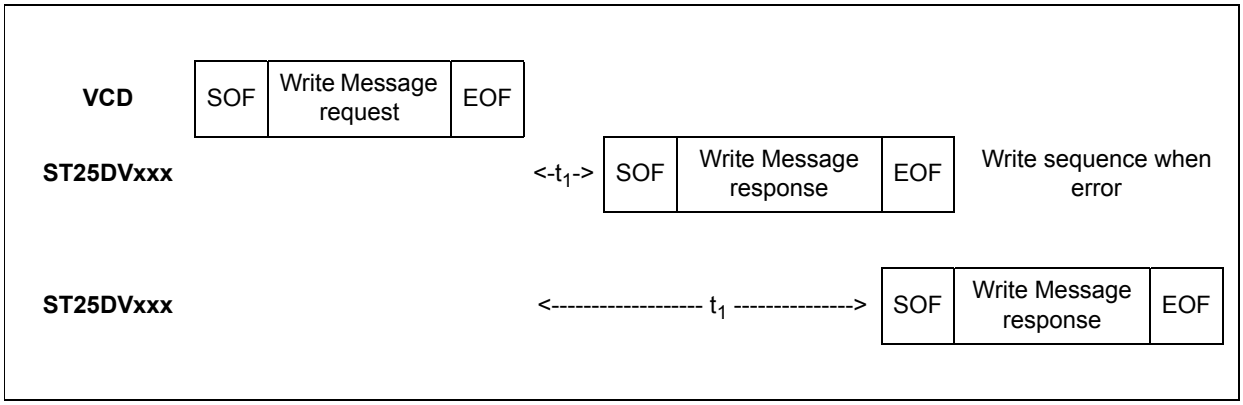
Table 162. Write Message response format when Error_flag is set

Response SOF	Response_flags	Error code	CRC16	Response EOF
-	8 bits	8 bits	16 bits	-

Response parameter:

- Error code as Error_flag is set:
  - 02h: command not recognized
  - 03h: command option not supported
  - 01h: command not supported
  - 0Fh: error with no information given

Figure 59. Write Message frame exchange between VCD and ST25DVxxx



7.6.32 Read Message Length

On receiving the Read Message Length command, the ST25DVxxx reads the MB_LEN_Dyn register which contains the Mailbox message length and sends back its 8-bit value in the response.

The Option_flag is not supported. The Inventory_flag must be set to 0.

Table 163. Read Message Length request format

Request SOF	Request_flags	Read Message Length	IC Mfg code	UID(1)	CRC16	Request EOF
-	8 bits	ABh	02h	64 bits	16 bits	-

1. Gray color means that the field is optional.

Request parameters:

- UID (optional)

Table 164. Read Message Length response format when Error_flag is NOT set

Response SOF	Response_flags	Data	CRC16	Response EOF
-	8 bits	8 bits	16 bits	-

Response parameters:

- One byte of data: MB_LEN_Dyn register value

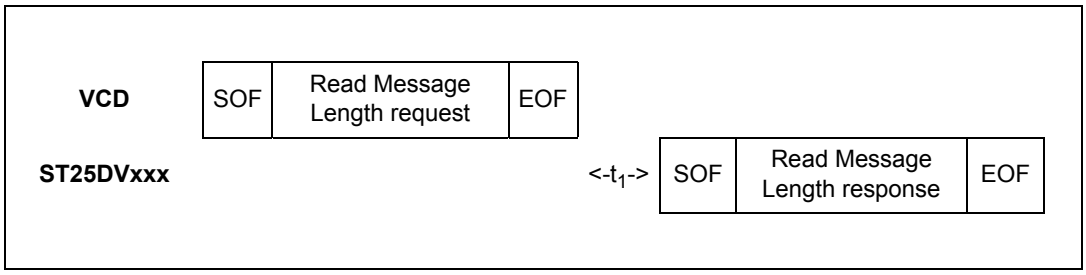
Table 165. Read Message Length response format when Error_flag is set

Response SOF	Response_flags	Error code	CRC16	Response EOF
-	8 bits	8 bits	16 bits	-

Response parameter:

- Error code as Error_flag is set
  - 01h: command not supported (option no mailbox)
  - 02h: command not recognized
  - 03h: the option is not supported
  - 0Fh: error given with no information

Figure 60. Read Message Length frame exchange between VCD and ST25DVxxx



7.6.33 Read Message

On receiving the Read Message command, the ST25DVxxx reads up to 256 byte in the Mailbox from the location specified by MBpointer and sends back their value in the response. First MailBox location is '00'. When Number of bytes is set to 00h and MBPointer is equals to 00h, the MB_LEN bytes of the full message are returned. Otherwise, Read Message command returns (Number of Bytes + 1) bytes (i.e. 01h returns 2 bytes, FFh returns 256 bytes).

An error is reported if (Pointer + Nb of bytes + 1) is greater than the message length. RF Reading of the last byte of the mailbox message automatically clears b1 of MB_CTRL_Dyn "HOST_PUT_MSG", and allows RF to put a new message.

Table 166. Read Message request format

Request SOF	Request_ flags	Read Message	IC Mfg code	UID(1)	MBpointer	Number of Bytes	CRC16	Request EOF
-	8 bits	ACh	02h	64 bits	8 bits	8 bits	16 bits	-

1. Gray color means that the field is optional.

Request parameters:

- Request flag
- UID (Optional)
- Pointer (start at 00h)
- Number of bytes is one less then the requested data

Table 167. Read Message response format when Error_flag is NOT set

Response SOF	Response_flags	Mailbox content	CRC16	Response EOF
-	8 bits	(Number of bytes + 1) bytes(1)	16 bits	-

1. Number of message Bytes when Number of Bytes is set to 00h.

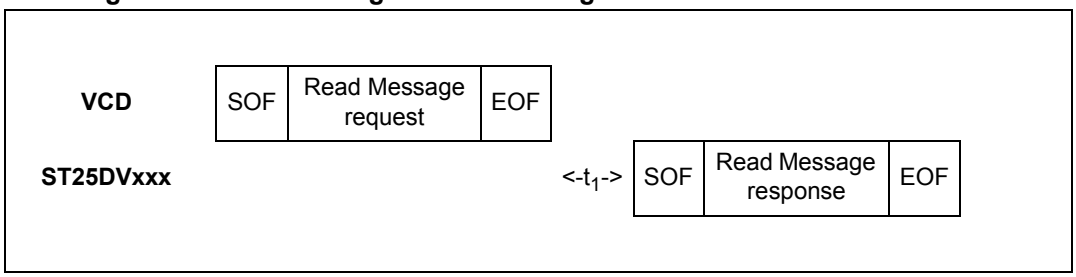
Response parameters:

- (number of data + 1) data bytes

Response parameter:

- Error code as Error_flag is set:
  - 01h: command not supported (option without mailbox)
  - 02h: command not recognized
  - 03h: command option not supported
  - 0Fh: error with no information given or No Host message available

Figure 61. Read Message frame exchange between VCD and ST25DVxxx



7.6.34 Fast Read Message

On receiving the Fast Read Message command, the ST25DVxxx reads up to 256 byte in the Mailbox from the location specified by MBpointer and sends back their value in the response. First MailBox location is '00'. When Number of bytes is set to 00h and MBPointer is equals to 00h, the MB_LEN bytes of the full message are returned. Otherwise, Fast Read Message command returns (Number of Bytes + 1) bytes (i.e. 01h returns 2 bytes, FFh returns 256 bytes).

An error is reported if (Pointer + Nb of bytes + 1) is greater than the message length..

RF Reading of the last byte of mailbox message automatically clears b1 of MB_CTRL_Dyn “HOST_PUT_MSG” and allows RF to put a new message.

The data rate of the response is multiplied by 2 compated to Read Message.

The subcarrier_flag should be set to 0, otherwise the ST25DVxxx answers with an error code.

Request parameters:

- Request flag
- UID (Optional)
- Pointer (start at 00h)
- Number of bytes is one less than the requested data

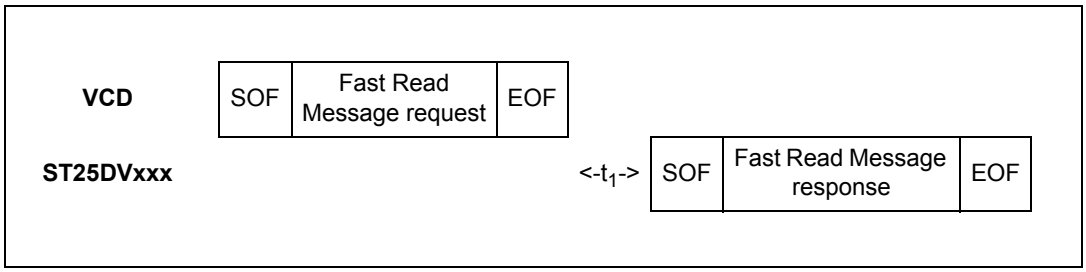
Response parameters:

- (number of bytes + 1) data bytes

Response parameter:

- Error code as Error_flag is set:
  - 01h: command not supported (option without mailbox)
  - 02h: command not recognized
  - 03h: command option not supported
  - 0Fh: error with no information given or No Host message available.

Figure 62. Fast Read Message frame exchange between VCD and ST25DVxxx



7.6.35 Write Password

On receiving the Write Password command, the ST25DVxxx uses the data contained in the request to write the password and reports whether the operation was successful in the response. It is possible to modify a Password value only after issuing a valid Present password command (of the same password number). When the Option_flag is set, wait for EOF to respond. Refer to *Section 5.6: Data Protection* for details on password Management.

During the RF write cycle time, W_t, there must be no modulation at all (neither 100% nor 10%), otherwise the ST25DVxxx may not correctly program the data into the memory.

The W_t time is equal to t_{1nom} + N × 302 µs (N is an integer). After a successful write, the new value of the selected password is automatically activated. It is not required to present the new password value until the ST25DVxxx power-down.

Caution: If ST25DVxxx is powered through V_{CC}, removing V_{CC} or setting LPD high during Write Password command can abort the command. As a consequence, before writing a new password, RF user should check if V_{CC} is ON, by reading EH_CTRL_Dyn register bit 3 (VCC_ON), and eventually ask host to maintain or to shut down V_{CC}, and not to change voltage applied on LPD while issuing the Write Password command in order to avoid password corruption.

Table 168. Write Password request format

Request SOF	Request _flags	Write password	IC Mfg code	UID(1)	Password number	Data	CRC16	Request EOF
-	8 bits	B1h	02h	64 bits	8 bits	64 bits	16 bits	-

1. Gray color means that the field is optional.

Request parameter:

- Request flags
- UID (optional)
- Password number:
  - 00h = RF configuration password RF_PWD_0,
  - 01h = RF_PWD_1,
  - 02h = RF_PWD_2,
  - 03h = RF_PWD_3,
  - other = Error)
- Data

Table 169. Write Password response format when Error_flag is NOT set

Response SOF	Response_flags	CRC16	Response EOF
-	8 bits	16 bits	-

Response parameter:

- no parameter.

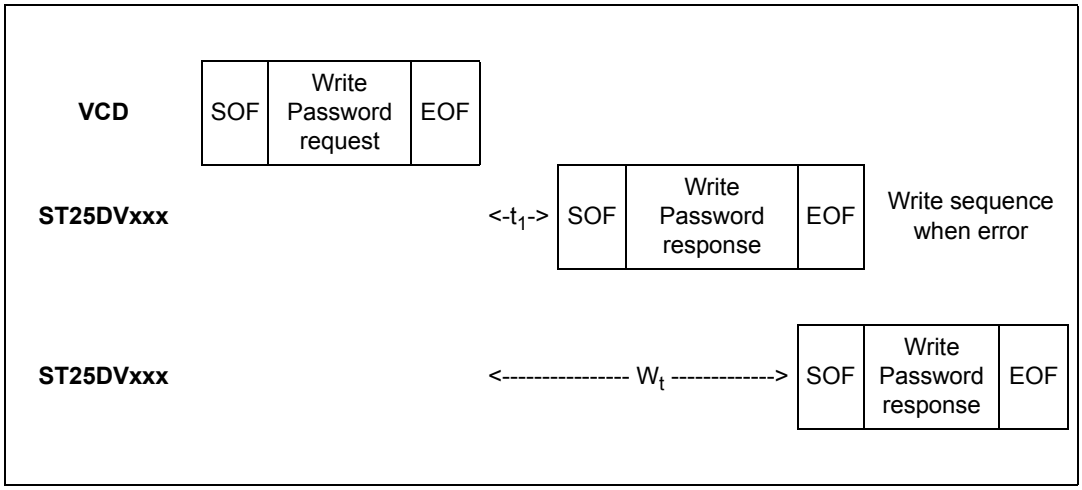
Table 170. Write Password response format when Error_flag is set

Response SOF	Response_flags	Error code	CRC16	Response EOF
-	8 bits	8 bits	16 bits	-

Response parameter:

- Error code as Error_flag is set:
  - 02h: command not recognized
  - 03h: command option not supported
  - 0Fh: the presented password is incorrect
  - 10h: the password number is incorrect
  - 12h: update right not granted, Present Password command not previously executed successfully
  - 13h: the specified block was not successfully programmed

Figure 63. Write Password frame exchange between VCD and ST25DVxxx



7.6.36 Present Password

On receiving the Present Password command, the ST25DVxxx compares the requested password with the data contained in the request and reports if the operation has been successful in the response. Refer to *Section 5.6: Data Protection* for details on password Management. After a successful command, the security session associate to the password is open as described in *Section 5.6: Data Protection*.

Table 171. Present Password request format

Request SOF	Request _flags	Present Password	IC Mfg code	UID(1)	Password number	Password	CRC16	Request EOF
-	8 bits	B3h	02h	64 bits	8 bits	64 bits	16 bits	-

1. Gray color means that the field is optional.

Request parameter:

- Request flags
- UID (optional)
- Password Number (00h = Password configuration, 0x01 = Pswd1, 0x02 = Pswd2, 0x03 = Pswd3, other = Error)
- Password

Table 172. Present Password response format when Error_flag is NOT set

Response SOF	Response_flags	CRC16	Response EOF
-	8 bits	16 bits	-

Response parameter:

- No parameter. The response is sent back after the write cycle.

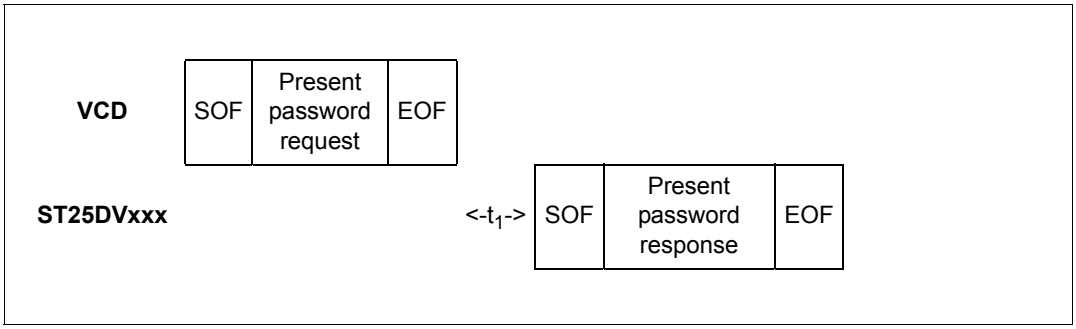
Table 173. Present Password response format when Error_flag is set

Response SOF	Response_flags	Error code	CRC16	Response EOF
-	8 bits	8 bits	16 bits	-

Response parameter:

- Error code as Error_flag is set:
  - 02h: command not recognized
  - 03h: command option not supported
  - 0Fh: the present password is incorrect
  - 10h: the password number is incorrect

Figure 64. Present Password frame exchange between VCD and ST25DVxxx



7.6.37 Fast Read Single Block

On receiving the Fast Read Single Block command, the ST25DVxxx reads the requested block and sends back its 32-bit value in the response. When the Option_flag is set, the response includes the Block Security Status. The data rate of the response is multiplied by 2.

The subcarrier_flag should be set to 0, otherwise the ST25DVxxx answers with an error code.

Block number is coded on 1 Byte and only first 256 blocks of ST25DV16K-xx and ST25DV64K-xx can be addressed using this command.

Table 174. Fast Read Single Block request format

Request SOF	Request_flags	Fast Read Single Block	IC Mfg code	UID(1)	Block number	CRC16	Request EOF
-	8 bits	C0h	02h	64 bits	8 bits	16 bits	-

1. Gray color means that the field is optional.

Request parameters:

- Request flags
- UID (optional)
- Block number

Table 175. Fast Read Single Block response format when Error_flag is NOT set

Response SOF	Response_flags	Block security status(1)	Data	CRC16	Response EOF
-	8 bits	8 bits	32 bits	16 bits	-

1. Gray color means that the field is optional.

Response parameters:

- Block security status if Option_flag is set (see *Table 176: Block security status*)
- Four bytes of block data

Table 176. Block security status

b_7	b_6	b_5	b_4	b_3	b_2	b_1	b_0
Reserved for future use All at 0							0: Current Block not locked 1: Current Block locked

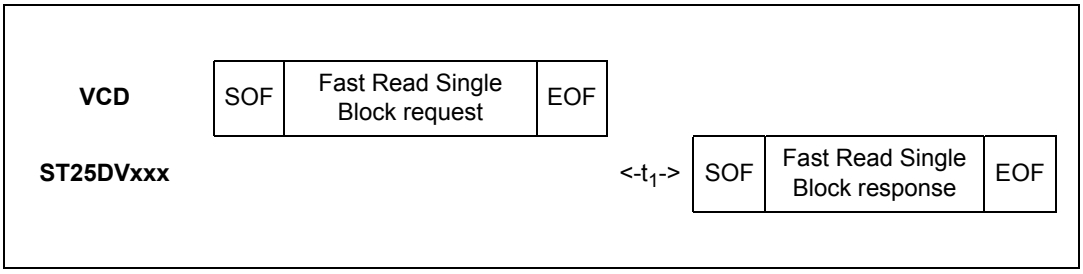
Table 177. Fast Read Single Block response format when Error_flag is set

Response SOF	Response_flags	Error code	CRC16	Response EOF
-	8 bits	8 bits	16 bits	-

Response parameter:

- Error code as Error_flag is set:
  - 02h: command not recognized
  - 03h: command option not supported
  - 0Fh: error with no information given
  - 10h: the specified block is not available
  - 15h: the specified block is read-protected

Figure 65. Fast Read Single Block frame exchange between VCD and ST25DVxxx



7.6.38 Fast Extended Read Single Block

On receiving the Fast Extended Read Single Block command, the ST25DVxxx reads the requested block and sends back its 32-bit value in the response. When the Option_flag is set, the response includes the Block Security Status. The data rate of the response is multiplied by 2.

The subcarrier_flag should be set to 0, otherwise the ST25DVxxx answers with an error code.

Block number is coded on 2 Bytes so all memory blocks of ST25DV16K-xx and ST25DV64K-xx can be addressed using this command

Table 178. Fast Extended Read Single Block request format

Request SOF	Request_flags	Fast Extended Read Single Block	IC Mfg code	UID(1)	Block number	CRC16	Request EOF
-	8 bits	C4h	02h	64 bits	16 bits	16 bits	-

1. Gray color means that the field is optional.

Request parameters:

- Request flags
- UID (optional)
- Block number

Table 179. Fast Extended Read Single Block response format when Error_flag is NOT set

Response SOF	Response_flags	Block security status(1)	Data	CRC16	Response EOF
-	8 bits	8 bits	32 bits	16 bits	-

1. Gray color means that the field is optional.

Response parameters:

- Block security status if Option_flag is set (see *Table 176: Block security status*)
- Four bytes of block data

Table 180. Block security status

b_7	b_6	b_5	b_4	b_3	b_2	b_1	b_0
Reserved for future use All at 0							0: Current Block not locked 1: Current Block locked

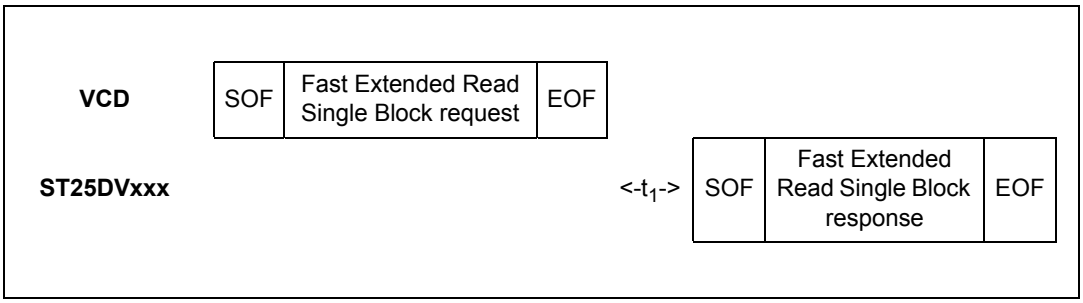
Table 181. Fast Extended Read Single Block response format when Error_flag is set

Response SOF	Response_flags	Error code	CRC16	Response EOF
-	8 bits	8 bits	16 bits	-

Response parameter:

- Error code as Error_flag is set:
  - 02h: command not recognized
  - 03h: command option not supported
  - 0Fh: error with no information given
  - 10h: the specified block is not available
  - 15h: the specified block is read-protected

Figure 66. Fast Extended Read Single Block frame exchange between VCD and ST25DVxxx



7.6.39 Fast Read Multiple Blocks

On receiving the Fast Read Multiple Blocks command, the ST25DVxxx reads the selected blocks and sends back their value in multiples of 32 bits in the response. The blocks are numbered from 00h up to the last block of user memory in the request, and the value is minus one (–1) in the field. For example, if the “Number of blocks” field contains the value 06h, seven blocks are read. The maximum number of blocks is fixed to 256 assuming that they are all located in the same area. If the number of blocks overlaps area, the ST25DVxxx returns an error code.

When the Option_flag is set, the response includes the Block Security Status. The data rate of the response is multiplied by 2.

The subcarrier_flag should be set to 0, otherwise the ST25DVxxx answers with an error code.

Block number is coded on 1 Byte and only first 256 blocks of ST25DV16K-xx and ST25DV64K-xx can be addressed using this command.

Table 182. Fast Read Multiple Block request format

Request SOF	Request_ flags	Fast Read Multiple Block	IC Mfg code	UID(1)	First block number	Number of blocks	CRC16	Request EOF
-	8 bits	C3h	02h	64 bits	8 bits	8 bits	16 bits	-

1. Gray color means that the field is optional.

Request parameters:

- Request flag
- UID (Optional)
- First block number
- Number of blocks

Table 183. Fast Read Multiple Block response format when Error_flag is NOT set

Response SOF	Response_flags	Block security status(1)	Data	CRC16	Response EOF
-	8 bits	8 bits(2)	32 bits(2)	16 bits	-

1. Gray color means that the field is optional.
2. Repeated as needed.

Response parameters:

- Block security status if Option_flag is set (see *Table 184: Block security status if Option_flag is set*)
- N block of data

Table 184. Block security status if Option_flag is set

b_7	b_6	b_5	b_4	b_3	b_2	b_1	b_0
Reserved for future use All at 0							0: Current not locked 1: Current locked

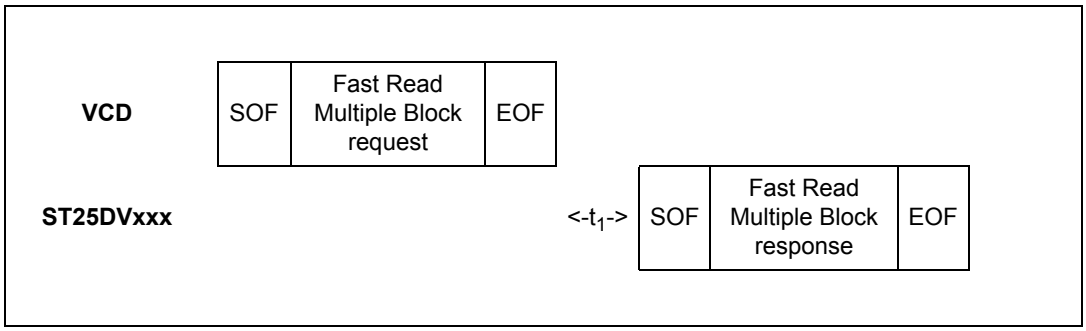
Table 185. Fast Read Multiple Block response format when Error_flag is set

Response SOF	Response_flags	Error code	CRC16	Response EOF
-	8 bits	8 bits	16 bits	-

Response parameter:

- Error code as Error_flag is set:
  - 02h: command not recognized
  - 0Fh: error with no information given
  - 03h: the option is not supported
  - 10h: block address not available
  - 15h: block read-protected

Figure 67. Fast Read Multiple Block frame exchange between VCD and ST25DVxxx



7.6.40 Fast Extended Read Multiple Block

On receiving the Fast Extended Read Multiple Block command, the ST25DVxxx reads the selected blocks and sends back their value in multiples of 32 bits in the response. The blocks are numbered from 00h to up to the last block of memory in the request and the value is minus one (–1) in the field. For example, if the “Number of blocks” field contains the value 06h, seven blocks are read. The maximum number of blocks is fixed to 2047 assuming that they are all located in the same area. If the number of blocks overlaps several areas, the ST25DVxxx returns an error code.

When the Option_flag is set, the response includes the Block Security Status. The data rate of the response is multiplied by 2.

The subcarrier_flag should be set to 0, otherwise the ST25DVxxx answers with an error code.

Block number is coded on 2 Bytes so all memory blocks of ST25DV16K-xx and ST25DV64K-xx can be addressed using this command.

Table 186. Fast Extended Read Multiple Block request format

Request SOF	Request_ flags	Fast Extended Read Multiple Block	IC Mfg code	UID(1)	First block number	Block Number	CRC16	Request EOF
-	8 bits	C5h	02h	64 bits	16 bits	16 bits	16 bits	-

1. Gray color means that the field is optional.

Request parameters:

- Request flag
- UID (Optional)
- First block number
- Number of blocks

Table 187. Fast Extended Read Multiple Block response format when Error_flag is NOT set

Response SOF	Response_flags	Block security status(1)	Data	CRC16	Response EOF
-	8 bits	8 bits(2)	32 bits(2)	16 bits	-

1. Gray color means that the field is optional.
2. Repeated as needed.

Response parameters:

- Block security status if Option_flag is set (see *Table 184: Block security status if Option_flag is set*)
- N block of data

Table 188. Block security status if Option_flag is set

b_7	b_6	b_5	b_4	b_3	b_2	b_1	b_0
Reserved for future use All at 0							0: Current not locked 1: Current locked

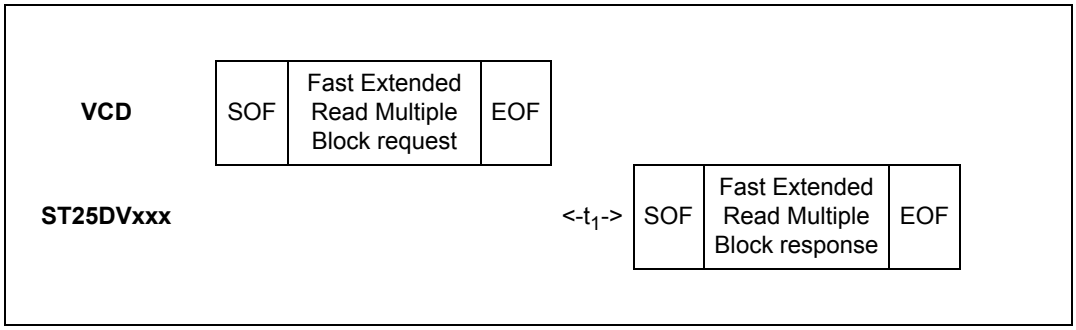
Table 189. Fast Read Multiple Block response format when Error_flag is set

Response SOF	Response_flags	Error code	CRC16	Response EOF
-	8 bits	8 bits	16 bits	-

Response parameter:

- Error code as Error_flag is set:
  - 02h: command not recognized
  - 03h: the option is not supported
  - 0Fh: error with no information given
  - 10h: block address not available
  - 15h: block read-protected

Figure 68. Fast Extended Read Multiple Block frame exchange between VCD and ST25DVxxx



7.6.41 Fast Write Message

On receiving the Fast Write Message command, the ST25DVxxx puts the data contained in the request into the mailbox buffer, updates the Message Length register MB_LEN_Dyn, and set Mailbox loaded bit RF_PUT_MSG. It then reports if the write operation was successful in the response. The ST25DVxxx mailbox contains up to 256 data bytes which are filled from the first location '00'. MSGlength parameter of the command is the number of Data bytes minus - 1 (00 for 1 byte of data, FFh for 256 bytes of data). Fast Write Message can be executed only when Mailbox is accessible by RF (previous RF message was read or time-out occurs, no I^2C message to be read). User can check it by reading b1 of MB_CTRL_Dyn “HOST_PUT_MSG”, which must be reset to “0”. (refer to *Section 5.1: Fast transfer mode (FTM)*).

The data rate of the response is multiplied by 2 compared to Write Message command.

Table 190. Fast Write Message request format

Request SOF	Request _flags	Fast Write Message	IC Mfg code	UID(1)	MSGLength	Message Data	CRC16	Request EOF
-	8 bits	CAh	02h	64 bits	1 byte	(MsgLenght + 1) bytes	16 bits	-

1. Gray color means that the field is optional.

Request parameters:

- Request flag
- UID (optional)
- Message Lenght
- Message Data

Table 191. Fast Write Message response format when Error_flag is NOT set

Response SOF	Response_flags	CRC16	Response EOF
-	8 bits	16 bits	-

Response parameters:

- No parameter. The response is sent back after the write cycle.

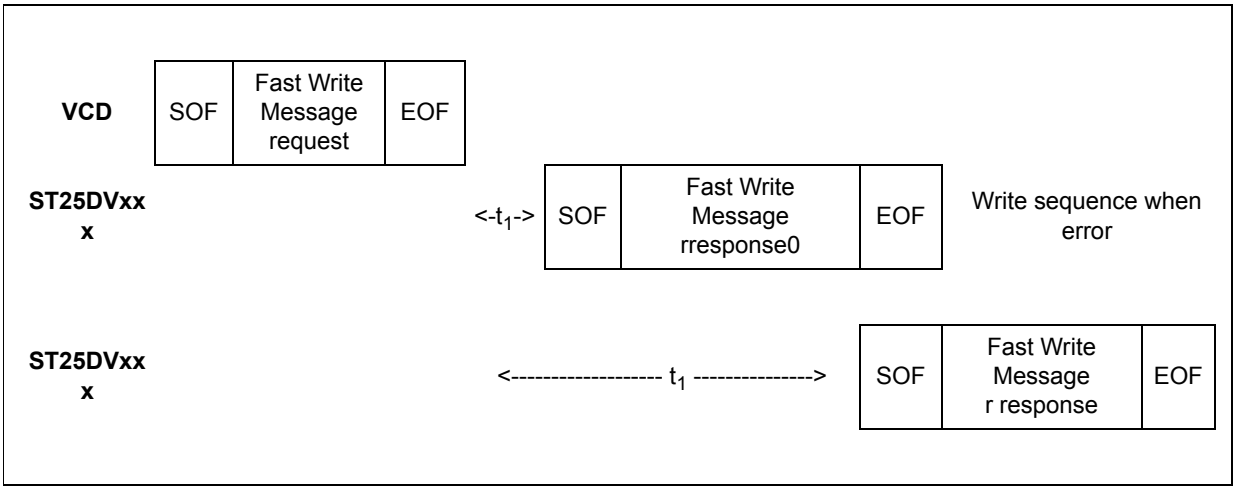
Table 192. Fast Write Message response format when Error_flag is set

Response SOF	Response_flags	CRC16	Response EOF
-	8 bits	8 bits	16 bits

Response parameter:

- Error code as Error_flag is set:
  - 02h: command not recognized
  - 03h: command option not supported
  - 01h: command not supported
  - 0Fh: error with no information given

Figure 69. Fast Write Message frame exchange between VCD and ST25DVxxx



7.6.42 Fast Read Message Length

On receiving the Fast Read Message Length command, the ST25DV reads the MB_LEN_dyn register which contains the mailbox message length and sends back its 8-bit value in the response. The Option_flag is not supported. The Inventory_flag must be set to 0.

The data rate of the response is multiplied by 2 compared to Read Message Length command.

Table 193. Fast Read Message Length request format

Request SOF	Request_flags	Fast Read Message Length	IC Mfg code	UID(1)	CRC16	Request EOF
-	8 bits	CBh	02h	64 bits	16 bits	-

1. Gray color means that the field is optional.

Request parameters:

- Request flag
- UID (optional)

Table 194. Fast Read Message Length response format when Error_flag is NOT set

Response SOF	Response_flags	Data	CRC16	Response EOF
-	8 bits	8 bits	16 bits	-

Response parameters:

- One byte of data: volatile Control register.

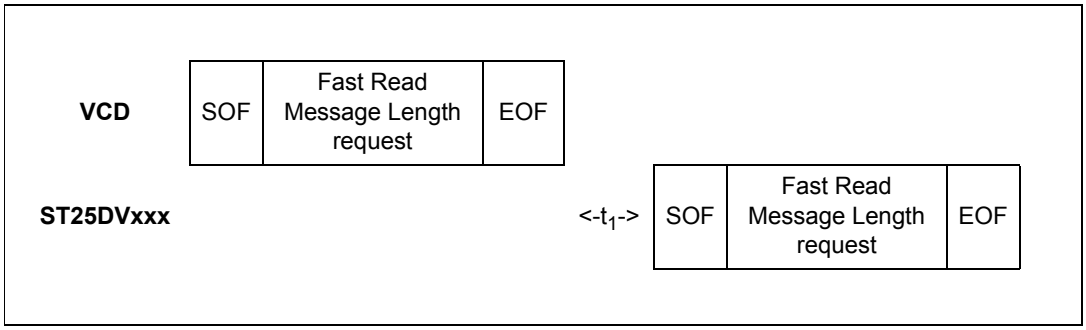
Table 195. Fast Read Message Length response format when Error_flag is set

Response SOF	Response_flags	Error code	CRC16	Response EOF
-	8 bits	8 bits	16 bits	-

Response parameter:

- Error code as Error_flag is set:
  - 01h: command not supported
  - 02h: command option not recognized
  - 03h: command not supported
  - 0Fh: error with no information given

Figure 70. Fast Read Message Length frame exchange between VCD and ST25DVxxx



7.6.43 Fast Read Dynamic Configuration

On receiving the Fast Read Dynamic Configuration command, the ST25DVxxx reads the Dynamic register address by the pointer and sends back its 8-bit value in the response.

The Option_flag is not supported. The Inventory_flag must be set to 0.

The data rate of the response is multiplied by 2 compared to Read Dynamic Configuration command.

Table 196. Fast Read Dynamic Configuration request format

Request SOF	Request_flags	Fast Read Dynamic Configuration	IC Mfg code	UID(1)	Pointer address	CRC16	Request EOF
-	8 bits	CDh	02h	64 bits	8 bits	16 bits	-

1. Gray color means that the field is optional.

Request parameters:

- Request flag
- UID (optional)

Table 197. Fast Read Dynamic Configuration response format when Error_flag is NOT set

Response SOF	Response_flags	Data	CRC16	Response EOF
-	8 bits	8 bits	16 bits	-

Response parameters:

- One byte of data

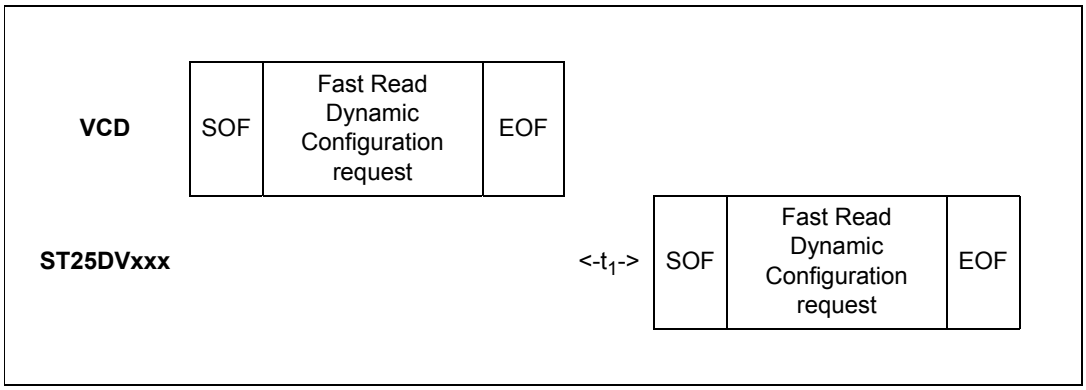
Table 198. Fast Read Dynamic Configuration response format when Error_flag is set

Response SOF	Response_flags	Error code	CRC16	Response EOF
-	8 bits	8 bits	16 bits	-

Response parameter:

- Error code as Error_flag is set:
  - 02h: command not recognized
  - 03h: command option not supported
  - 0Fh: error with no information given
  - 10h: block not available

Figure 71. Fast Read Dynamic Configuration frame exchange between VCD and ST25DVxxx



7.6.44 Fast Write Dynamic Configuration

On receiving the Fast Write Dynamic Configuration command, the ST25DV updates the Dynamic register addressed by the pointer.

The Option_flag is not supported. The Inventory_flag must be set to 0.

The data rate of the response is multiplied by 2 compared to Write Dynamic Configuration command.

Table 199. Fast Write Dynamic Configuration request format

Request SOF	Request_flags	Fast Write Dynamic Configuration	IC Mfg code	UID(1)	Pointer address	Register Value	CRC16	Request EOF
-	8 bits	CEh	02h	64 bits	8 bits	8 bits	16 bits	-

1. Gray color means that the field is optional.

Request parameters:

- Request flag
- UID (optional)
- Pointer address
- Register value

Table 200. Fast Write Dynamic Configuration response format when Error_flag is NOT set

Response SOF	Response_flags	CRC16	Response EOF
-	8 bits	16 bits	-

Response parameters:

- No parameter. The response is sent back after t_1.

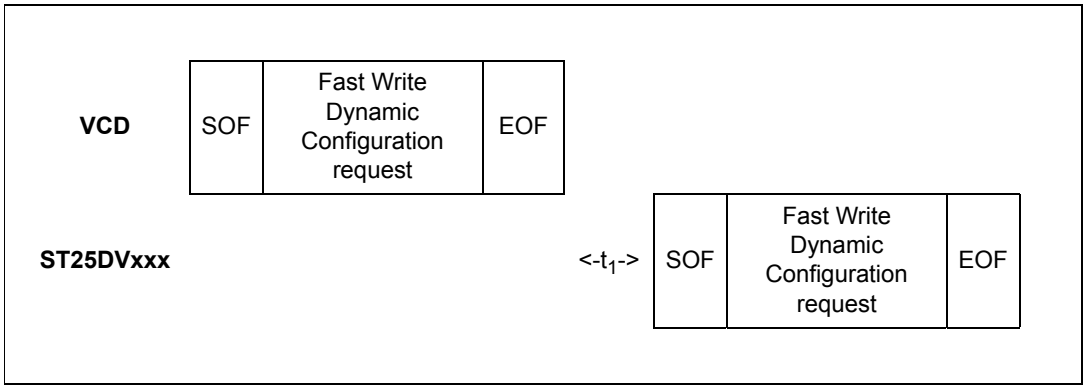
Table 201. Fast Write Dynamic Configuration response format when Error_flag is set

Response SOF	Response_flags	Error code	CRC16	Response EOF
-	8 bits	8 bits	16 bits	-

Response parameter:

- Error code as Error_flag is set:
  - 02h: command not recognized
  - 03h: command option not supported
  - 0Fh: error with no information given
  - 10h: block not available

Figure 72. Fast Write Dynamic Configuration frame exchange between VCD and ST25DVxxx



8 Unique identifier (UID)

The ST25DVxxx is uniquely identified by a 64-bit unique identifier (UID). This UID complies with ISO/IEC 15963 and ISO/IEC 7816-6. The UID is a read-only code and comprises:

- eight MSBs with a value of E0h,
- the IC manufacturer code “ST 02h” on 8 bits (ISO/IEC 7816-6/AM1),
- a unique serial number on 48 bits.

Table 202. UID format

MSB		LSB	
63 56	**55 48**	**47 40**	**40 0**
0xE0	0x02	ST product code(1)	Unique serial number

1. See *Table 50: UID* for ST product code value definition.

With the UID, each ST25DVxxx can be addressed uniquely and individually during the anticollision loop and for one-to-one exchanges between a VCD and an ST25DVxxx.

9 Device parameters

9.1 Maximum rating

Stressing the device above the rating listed in *Table 203: Absolute maximum ratings* may cause permanent damage to the device. These are stress ratings only and operation of the device, at these or any other conditions above those indicated in the operating sections of this specification, is not implied. Exposure to absolute maximum rating conditions for extended periods may affect the device reliability. Refer also to the STMicroelectronics SURE Program and other relevant quality documents.

Table 203. Absolute maximum ratings

Symbol	Parameter		Min.	Max.	Unit
T_A	Ambient operating temperature		- 40	85	°C
T_{STG}	Storage Temperature	Sawn wafer on UV tape kept in its original packing form	15	25	°C
t_{STG}	Retain		-	9(1)	months
T_{STG}	Storage temperature	UFDFPN8 (MLP8), SO8, TSSOP8, UFDFPN12	- 65	150	°C
T_{LEAD}	Lead temperature during soldering		see note (2)		°C
V_{IO}	I^2C input or output range		- 0.50	6.5	V
V_{DCG}	Supply GPO CMOS driver		- 0.50	6.5	V
V_{CC}	I^2C supply voltage		- 0.50	6.5	V
$I_{OL_MAX_SDA}$	DC output current on pin SDA (when equal to 0)		-	5	mA
$I_{OL_MAX_GPO_OD}$	DC output current on pin GPO Open Drain (when equal to 0)		-	1.5	mA
V_{MAX_1}(3)	RF input voltage amplitude peak to peak between AC0 and AC1, V_{SS} pin left floating	V_{AC0} - V_{AC1}	-	11	V
V_{MAX_2}(3)	AC voltage between AC0 and V_{SS}, or AC1 and V_{SS}	V_{AC0} - V_{SS}, or V_{AC1} - V_{SS}	- 0.50	5.5	V
V_{ESD}	Electrostatic discharge voltage (human body model)(4)	All pins	2000	-	V

1. Counted from ST production date.
2. Compliant with JEDEC Std J-STD-020C (for small body, Sn-Pb or Pb assembly), the ST ECOPACK® 7191395 specification, and the European directive on Restrictions on Hazardous Substances (RoHS) 2002/95/EU.
3. Based on characterization, not tested in production.
4. AEC-Q100-002 (compliant with JEDEC Std JESD22-A114, C1 = 100 pF, R1 = 1500 Ω, R2 = 500 Ω)

9.2 I^2C DC and AC parameters

This section summarizes the operating and measurement conditions, and the DC and AC characteristics of the device in I^2C mode. The parameters in the DC and AC characteristic tables that follow are derived from tests performed under the measurement conditions summarized in the relevant tables. Designers should check that the operating conditions in their circuit match the measurement conditions when relying on the quoted parameters.

Table 204. I^2C operating conditions

Symbol	Parameter	Min.	Max.	Unit
V_{CC}	Supply voltage	1.8	5.5	V
T_A	Ambient operating temperature	- 40	85	°C

Table 205. AC test measurement conditions

Symbol	Parameter	Min.	Max.	Unit
C_L	Load capacitance	100		pF
t_r, t_f	Input rise and fall times	-	50	ns
V_{hi-lo}	Input levels	$0.2V_{CC}$ to $0.8V_{CC}$		V
$V_{ref(t)}$	Input and output timing reference levels	$0.3V_{CC}$ to $0.7V_{CC}$		V

Figure 73. AC test measurement I/O waveform

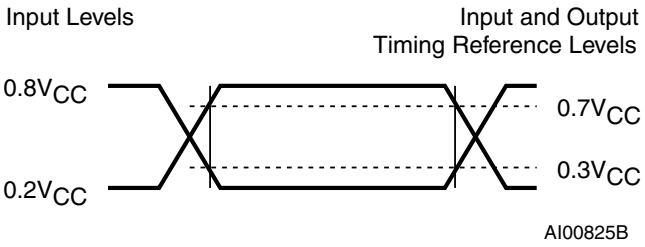


Table 206. Input parameters

Symbol	Parameter	Min.	Max.	Unit
C_{IN}	Input capacitance (SDA)	-	8	pF
C_{IN}	Input capacitance (other pins)	-	6	pF
t_{NS}(1)	Pulse width ignored (Input filter on SCL and SDA)	-	80	ns

1. Characterized only.

Table 207. I^2C DC characteristics

Symbol	Parameter	Test condition	Min.	Typ.	Max.	Unit
I_{LI}	Input leakage current (SCL, SDA)	$V_{IN} = V_{SS}$ or V_{CC} device in Standby mode	-	0.03	± 0.1	µA
I_{LI}	Input leakage current (LPD)	$V_{IN} = V_{SS}$ device in Standby mode	-	0.1	± 0.5	µA
I_{LO}	Output leakage current (SDA)	SDA in Hi-Z, external voltage applied on SDA: V_{SS} or V_{CC}	-	0.03	± 0.1	µA
$I_{CC}_E^2$	Operating Supply current (Device select E^2 Address) Read(1)	= 1.8 V, f_C = 1MHz (rise/fall time < 50 ns)	-	116	160	µA
		V_{CC} = 3.3 V, f_C = 1MHz (rise/fall time < 50 ns)	-	220	240	
		V_{CC} = 5.5 V, f_C = 1MHz (rise/fall time < 50 ns)	-	510	550	
I_{CC}_MB	Operating Supply current (Device select MB Address) Read(1)	V_{CC} = 1.8 V, f_C = 1MHz (rise/fall time < 50 ns)	-	116	160	µA
		V_{CC} = 3.3 V, f_C = 1MHz (rise/fall time < 50 ns)	-	220	240	
		V_{CC} = 5.5 V, f_C = 1MHz (rise/fall time < 50 ns)	-	510	550	
I_{CC0}	Operating Supply current (Device select E^2 Address) Write(1)	= 1.8 V, f_C = 1MHz (rise/fall time < 50 ns)	-	110	300	µA
		V_{CC} = 3.3 V, f_C = 1MHz (rise/fall time < 50 ns)	-	110	330	
		V_{CC} = 5.5 V, f_C = 1MHz (rise/fall time < 50 ns)	-	130	430	
I_{CC0}_MB	Operating Supply current (Device select MB Address) Write(1)	V_{CC} = 1.8 V, f_C = 1MHz (rise/fall time < 50 ns)	-	170	200	µA
		V_{CC} = 3.3 V, f_C = 1MHz (rise/fall time < 50 ns)	-	280	300	
		V_{CC} = 5.5 V, f_C = 1MHz (rise/fall time < 50 ns)	-	520	600	
I_{CC1} (LPD = 1)	Low Power Down supply current	V_{CC} = 1.8 V	-	0.84	1.5	µA
		V_{CC} = 3.3 V	-	1.3	2	
		V_{CC} = 5.5 V	-	1.7	3	
I_{CC1}_PON (LPD = 0)	Static Standby supply current after power ON or device select stop or time out	V_{CC} = 1.8 V	-	72	100	µA
		V_{CC} = 3.3 V	-	76	100	
		V_{CC} = 5.5 V	-	87	120	

Table 207. I^2C DC characteristics (continued)

Symbol	Parameter	Test condition	Min.	Typ.	Max.	Unit
V_{IL}	Input low voltage (SDA, SCL)	V_{CC} = 1.8 V	- 0.45	-	0.25 V_{CC}	V
		V_{CC} = 3.3 V	- 0.45	-	0.3 V_{CC}	
		V_{CC} = 5.5 V	- 0.45	-	0.3 V_{CC}	
V_{IL_LPD}	Input low voltage (LPD)	V_{CC} = 3.3 V	- 0.45	-	0.2 V_{CC}	V
V_{IH}	Input high voltage (SDA, SCL)	V_{CC} = 1.8 V	0.75 V_{CC}	-	V_{CC} + 1	V
		V_{CC} = 3.3 V	0.75 V_{CC}	-	V_{CC} + 1	
		V_{CC} = 5.5 V	0.75 V_{CC}	-	V_{CC} + 1	
V_{IH}_LPD	Input high voltage (LPD)	V_{CC} = 1.8 V	0.85 V_{CC}	-	V_{CC} + 1	V
		V_{CC} = 3.3 V	0.85 V_{CC}	-	V_{CC} + 1	
		V_{CC} = 5.5 V	0.85 V_{CC}	-	V_{CC} + 1	
V_{OL}_SDA	Output low voltage SDA (1 MHz)	I_{OL} = 1 mA, V_{CC} = 1.8 V	-	0.05	0.4	V
		I_{OL} = 2.1 mA, V_{CC} = 3.3 V	-	0.075	0.4	
		I_{OL} = 3 mA, V_{CC} = 5.5 V	-	0.09	0.4	
V_{CC}_Power_up	Device Select Acknowledge	f_C = 100 KHz	-	1.48	1.7	V

1. SCL, SDA connected to Ground or V_{CC}. SDA connected to V_{CC} through a pull-up resistor.

Table 208. I²C AC characteristics

Test conditions specified in *Table 204*					
Symbol	**Alt.**	**Parameter**	**Min.**	**Max.**	**Unit**
f_C	f_{SCL}	Clock frequency	0.05	1000	kHz
t_{CHCL}	t_{HIGH}	Clock pulse width high	0.26	25000(1)	µs
t_{CLCH}	t_{LOW}	Clock pulse width low	0.5	25000(2)	µs
t_{START_OUT}	-	I²C timeout on Start condition	35	-	ms
t_{XH1XH2}	t_R	Input signal rise time	(3)	(3)	ns
t_{XL1XL2}	t_F	Input signal fall time	(3)	(3)	ns
t_{DL1DL2}(4)	t_F	SDA (out) fall time	20	120	ns
t_{DXCX}	$t_{SU:DAT}$	Data in set up time	0	-	ns
t_{CLDX}	$t_{HD:DAT}$	Data in hold time	0	-	ns
t_{CLQX}(5)	t_{DH}	Data out hold time	100	-	ns
t_{CLQV}(6)	t_{AA}	Clock low to next data valid (access time)	-	450	ns
t_{CHDX}(7)	$t_{SU:STA}$	Start condition set up time	250	-	ns
t_{DLCL}	$t_{HD:STA}$	Start condition hold time	0.25	35000(8)	µs
t_{CHDH}	$t_{SU:STO}$	Stop condition set up time	250	-	ns
t_{DHDL}	t_{BUF}	Time between Stop condition and next Start condition	500	-	ns
t_W	-	I²C write time	-	5	ms
t_{bootDC}	-	RF OFF and LPD = 0	-	0.6	ms
$t_{bootLPD}$	-	RF OFF	-	0.6	ms

1. t_{CHCL} timeout.
2. t_{CLCH} timeout.
3. There is no min. or max. values for the input signal rise and fall times. It is however recommended by the I²C specification that the input signal rise and fall times be less than 120 ns when f_C < 1 MHz.
4. Characterized on bench.
5. To avoid spurious Start and Stop conditions, a minimum delay is placed between SCL=1 and the falling or rising edge of SDA.
6. t_{CLQV} is the time (from the falling edge of SCL) required by the SDA bus line to reach $0.8V_{CC}$ in a compatible way with the I²C specification (which specifies $t_{SU:DAT}$ (min) = 100 ns), assuming that the $R_{bus} \times C_{bus}$ time constant is less than 150 ns (as specified in the *Figure 75: I²C Fast mode (f_C = 1 MHz): maximum R_{bus} value versus bus parasitic capacitance (C_{bus})*).
7. For a reStart condition, or following a write cycle.
8. t_{DLCL} timeout.

Figure 74. I²C AC waveforms

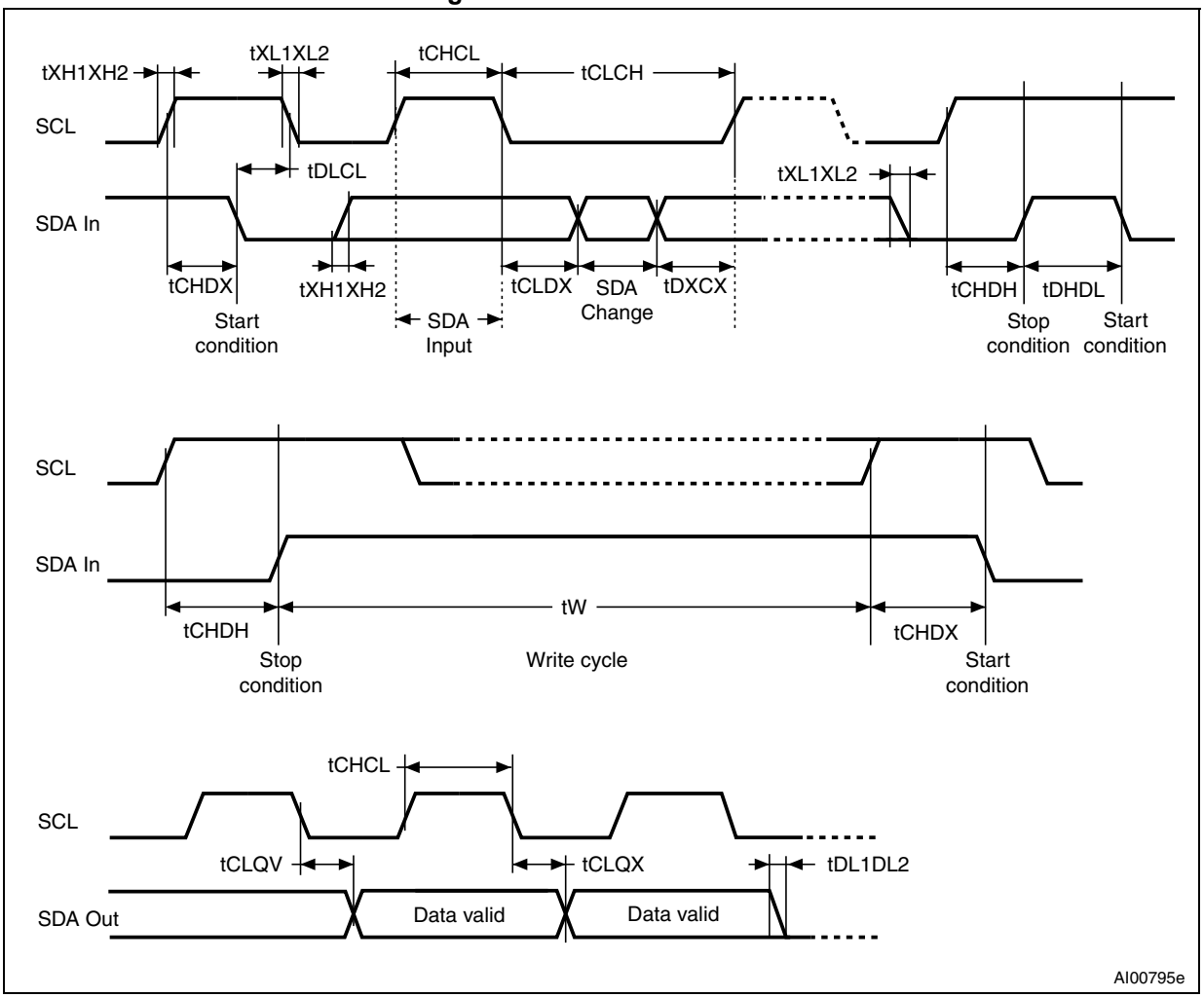
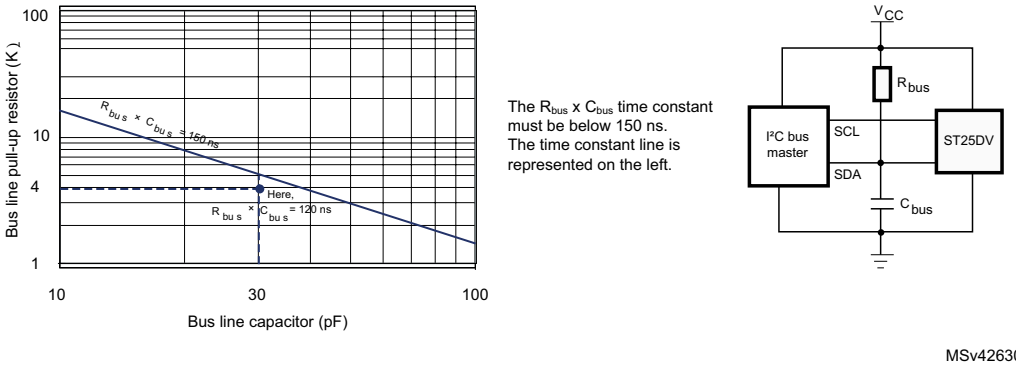


Figure 75 indicates how the value of the pull-up resistor can be calculated. In most applications, though, this method of synchronization is not employed, and so the pull-up resistor is not necessary, provided that the bus master has a push-pull (rather than open drain) output.

Figure 75. I²C Fast mode (f_C = 1 MHz): maximum R_{bus} value versus bus parasitic capacitance (C_{bus})



9.3 GPO Characteristics

This section summarizes the operating and measurement conditions of the GPO feature. The parameters in the DC and AC characteristic tables that follow are derived from tests performed under the measurement conditions summarized in the relevant tables.

Table 209. GPO DC characteristics

Symbol	Parameter	Condition	Min	Typ	Max	Unit
V_{OL}_GPO_CMOS	Output low voltage (GPO CMOS)	V_{DCG} = 1.8 V, I_{OL} = 0.5 mA	-	-	0.4	V
		V_{DCG} = 3.3 V, I_{OL} = 0.5 mA	-	-	0.4	
		V_{DCG} = 5.5 V, I_{OL} = 0.5 mA	-	-	0.4	
V_{OH}_GPO_CMOS	Output high voltage (GPO CMOS)	V_{DCG} = 1.8 V, I_{OH} = - 0.5 mA	V_{DCG} - 0.4	-	-	V
		V_{DCG} = 3.3 V, I_{OH} = - 0.5 mA	V_{DCG} - 0.4	-	-	
		V_{DCG} = 5.5 V, I_{OH} = - 0.5 mA	V_{DCG} - 0.4	-	-	
V_{OL}_GPO_OD	Output low voltage (GPO open drain)	IOL = 1 mA, VCC = 1.8 V	-	0.28	0.4	V
		IOL = 1 mA, VCC = 3.3 V	-	0.20	0.4	
		IOL = 1 mA, VCC = 5.5 V	-	0.20	0.4	
I_L_GPO_OD	Output leakage (GPO open drain)	GPO in Hi-Z, external voltage applied on: GPO, V_{SS} or V_{CC}	- 0.15	0.06	0.15	µA
I_{LI}_V_{DGC}	Input leakage (V_{DGC})	V_{DGC} = 5.5 V	-	-	0.1	µA

Table 210. GPO AC characteristics

Symbol	Parameter	Condition	Min	Max	Unit
t_r_GPO_CMOS	Output rise time	C_L = 30 pF, V_{DCG} = 1.8 V to 5.5 V	-	50	ns
t_f_GPO_CMOS	Output fall time	C_L = 30 pF, V_{DCG} = 1.8 V to 5.5 V	-	50	

9.4 RF electrical parameters

This section summarizes the operating and measurement conditions, and the DC and AC characteristics of the device in RF mode.

The parameters in the DC and AC characteristics tables that follow are derived from tests performed under the Measurement Conditions summarized in the relevant tables. Designers should check that the operating conditions in their circuit match the measurement conditions when relying on the quoted parameters.

Table 211. RF characteristics(1)(2)

Symbol	Parameter	Condition	Min	Typ	Max	Unit
f_{CC}	External RF signal frequency		13.553	13.56	13.567	MHz
H_ISO	Operating field according to ISO	T_A = -40 °C to 85 °C	150	-	5000	mA/m
$MI_{CARRIER}$	10% carrier modulation index (3) MI=(A-B)/(A+B)	150 mA/m > H_ISO > 1000 mA/m	10	-	30	%
	100% carrier modulation index	MI=(A-B)/(A+B)(4)	95	-	100	
$t_{MIN\ CD}$	Minimum time from carrier generation to first data	From H-field min	-	-	1	ms
f_{SH}	Subcarrier frequency high	F_{CC}/32	-	423.75	-	kHz
f_{SL}	Subcarrier frequency low	F_{CC}/28	-	484.28	-	kHz
t_1	Time for ST25DVxxx response	4352/F_C	318.6	320.9	323.3	µs
t_2	Time between commands	4192/F_C	309	311.5	314	µs
t_3	Time between commands	4384/F_C	323.3	-	-	µs
W_t_Block	RF User memory write time (including internal Verify)(5)	1 Block	-	5.2	-	ms
		4 Blocks	-	19.7	-	ms
W_t_Byte	RF system memory write time including internal Verify)(5)	1 Byte	-	4.9	-	ms
W_t_MB	RF User memory write time (including internal Verify)(5)(6)	256 Byte	-	80.7	-	ms
Read_MB	RF Mailbox write time (from VCD request SOF to ST25DVxxx response EOF) (5)(6)	256 Byte	-	81	-	ms
C_{TUN}	Internal tuning capacitor in SO8(6)	f = 13.56 MHz	26.5	28.5	30.5	pF
V_{BACK}	Backscattered level as defined by ISO test	-	10	-	-	mV

Table 211. RF characteristics(1)(2) (continued)

Symbol	Parameter	Condition	Min	Typ	Max	Unit
V_{MIN_1}(3)	RF input voltage amplitude between AC0 and AC1, V_{SS} pin left floating, VAC0-VAC1 peak to peak(3)	Inventory and Read operations	-	4.8	-	V
		Write operations	-	5.25	-	V
V_{MIN_2}(3)	AC voltage between AC0 and V_{SS} or between AC1 and V_{SS}(3)	Inventory and Read operations	-	2.25	-	V
		Write operations	-	2.7	-	V
t_{BootRF}	Without DC supply (No V_{CC})	Set up time	-	0.6	-	ms
t_{RF_OFF}	RF OFF time	Chip reset	2	-	-	ms

1. T_A = - –40 to 85 °C. Characterized only.
2. All timing characterizations were performed on a reference antenna with the following characteristics:
   ISO antenna class1
   Tuning frequency = 13.7 MHz
3. Characterized on bench.
4. Characterized at room temperature only, on wafer at POR Level.
5. For VCD request coded in 1 out of 4 and ST25DVxxx response in high data rate, single sub carrier.
6. The tuning capacitance value is measured with ST characterization equipment at chip Power On Reset. This value is used as reference for antenna design. Minimum and Maximum values come from correlation with industrial tester limits.

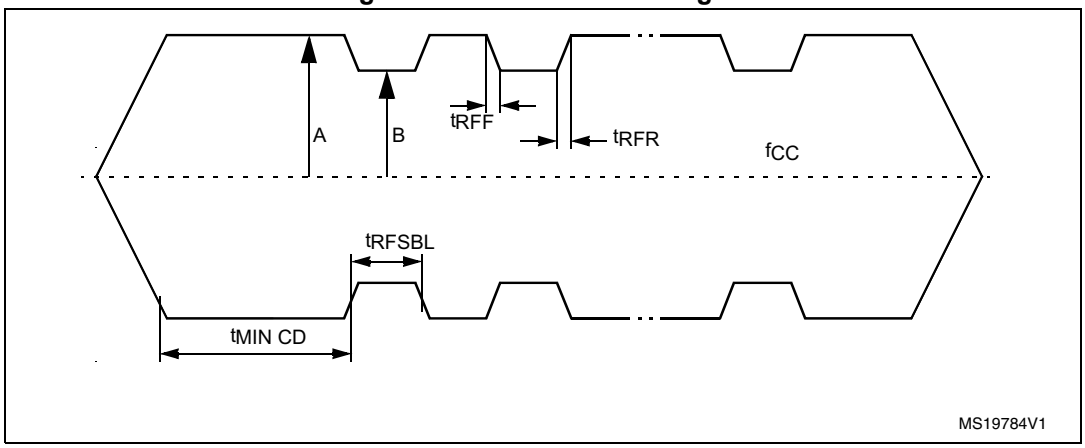
Table 212. Operating conditions

Symbol	Parameter	Min.	Max.	Unit
T_A	Ambient operating temperature	- 40	85	°C

Figure 76: ASK modulated signal shows an ASK modulated signal from the VCD to the ST25DVxxx. The test conditions for the AC/DC parameters are:

- Close coupling condition with tester antenna (1 mm)
- ST25DVxxx performance measured at the tag antenna
- ST25DVxxx synchronous timing, transmit and receive

Figure 76. ASK modulated signal

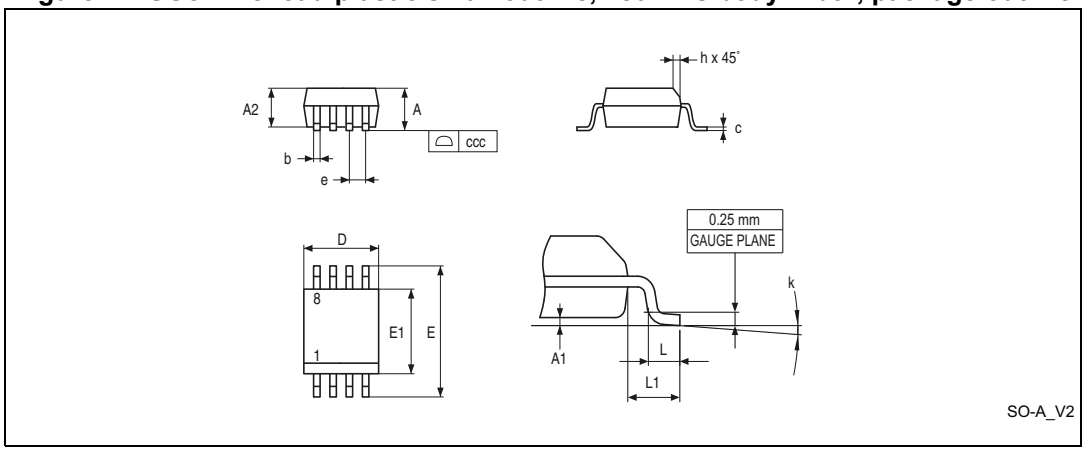


10 Package information

In order to meet environmental requirements, ST offers these devices in different grades of ECOPACK® packages, depending on their level of environmental compliance. ECOPACK® specifications, grade definitions and product status are available at: *www.st.com*. ECOPACK® is an ST trademark.

10.1 SO8N package information

Figure 77. SO8N – 8-lead plastic small outline, 150 mils body width, package outline



1. Drawing is not to scale.

Table 213. SO8N – 8-lead plastic small outline, 150 mils body width, package mechanical data

Symbol	millimeters			inches(1)		
	Min.	Typ.	Max.	Min.	Typ.	Max.
A	-	-	1.750	-	-	0.0689
A1	0.100	-	0.250	0.0039	-	0.0098
A2	1.250	-	-	0.0492	-	-
b	0.280	-	0.480	0.0110	-	0.0189
c	0.170	-	0.230	0.0067	-	0.0091
D	4.800	4.900	5.000	0.1890	0.1929	0.1969
E	5.800	6.000	6.200	0.2283	0.2362	0.2441
E1	3.800	3.900	4.000	0.1496	0.1535	0.1575
e	-	1.270	-	-	0.0500	-

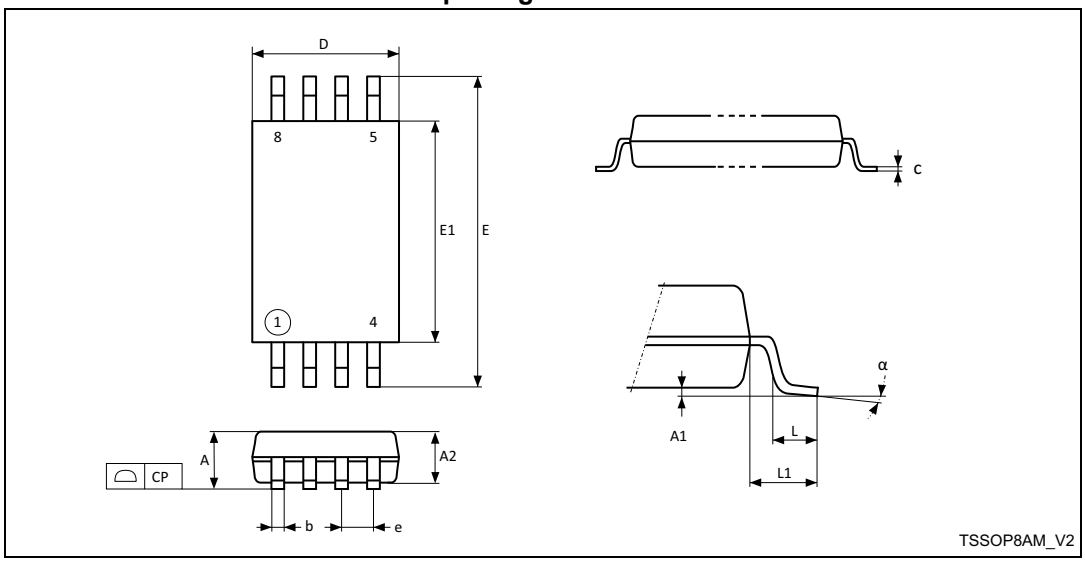
Table 213. SO8N – 8-lead plastic small outline, 150 mils body width, package mechanical data (continued)

Symbol	millimeters			inches[1]		
	Min.	Typ.	Max.	Min.	Typ.	Max.
h	0.250	-	0.500	0.0098	-	0.0197
k	0°	-	8°	0°	-	8°
L	0.400	-	1.270	0.0157	-	0.0500
L1	-	1.040	-	-	0.0409	-
ccc	-	-	0.100	-	-	0.0039

1. Values in inches are converted from mm and rounded to four decimal digits.

10.2 TSSOP8 package information

Figure 78.TSSOP8 – 8-lead thin shrink small outline, 3 x 4.4 mm, 0.5 mm pitch, package outline



1. Drawing is not to scale.

Table 214. TSSOP8 – 8-lead thin shrink small outline, 3 x 4.4 mm, 0.5 mm pitch, package mechanical data

Symbol	millimeters			inches[1]		
	Min.	Typ.	Max.	Min.	Typ.	Max.
A	-	-	1.200	-	-	0.0472
A1	0.050	-	0.150	0.0020	-	0.0059
A2	0.800	1.000	1.050	0.0315	0.0394	0.0413
b	0.190	-	0.300	0.0075	-	0.0118
c	0.090	-	0.200	0.0035	-	0.0079

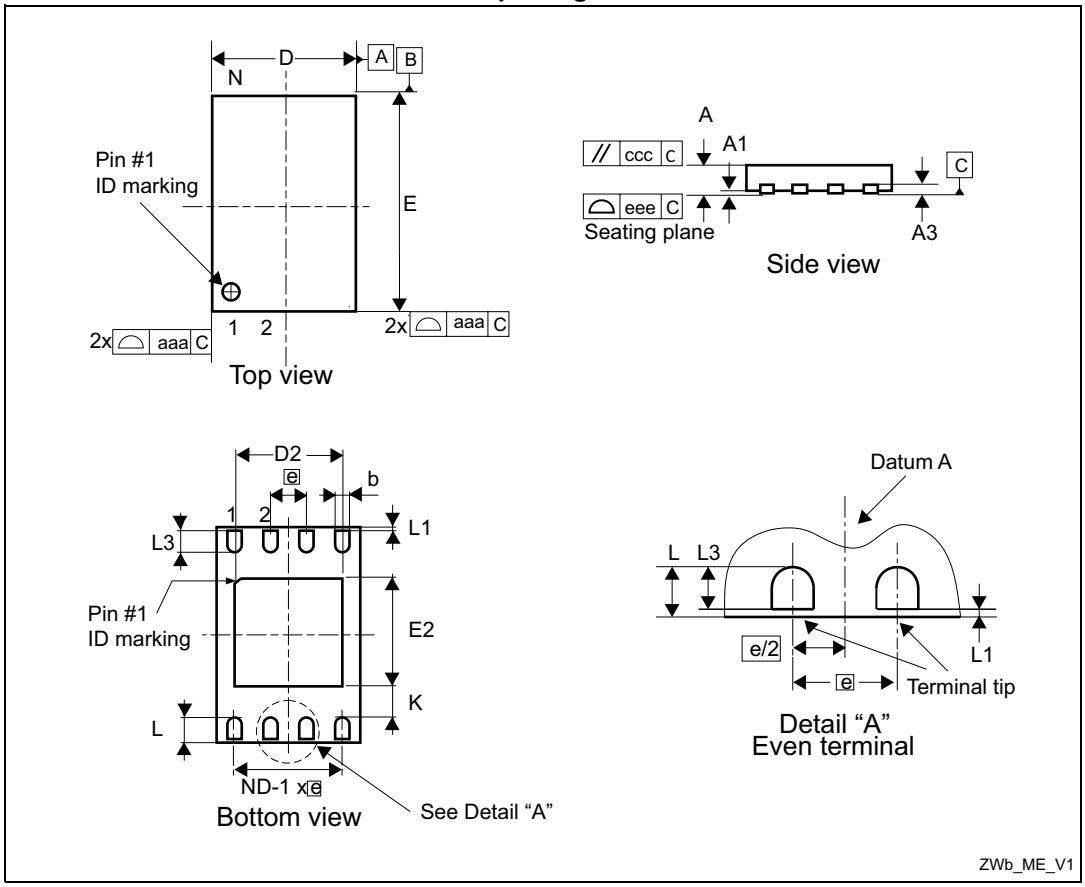
Table 214. TSSOP8 – 8-lead thin shrink small outline, 3 x 4.4 mm, 0.5 mm pitch, package mechanical data (continued)

Symbol	millimeters			inches[1]		
	Min.	Typ.	Max.	Min.	Typ.	Max.
CP	-	-	0.100	-	-	0.0039
D	2.900	3.000	3.100	0.1142	0.1181	0.1220
e	-	0.650	-	-	0.0256	-
E	6.200	6.400	6.600	0.2441	0.2520	0.2598
E1	4.300	4.400	4.500	0.1693	0.1732	0.1772
L	0.450	0.600	0.750	0.0177	0.0236	0.0295
L1	-	1.000	-	-	0.0394	-
α	0°	-	8°	0°	-	8°

1. Values in inches are converted from mm and rounded to four decimal digits.

10.3 UFDFN8 package information

Figure 79. UFDFN8 - 8-lead, 2 × 3 mm, 0.5 mm pitch ultra thin profile fine pitch dual flat package outline



1. Max. package warpage is 0.05 mm.
2. Exposed copper is not systematic and can appear partially or totally according to the cross section.
3. Drawing is not to scale.

Table 215. UFDFN8 - 8-lead, 2 × 3 mm, 0.5 mm pitch ultra thin profile fine pitch dual flat package mechanical data

Symbol	millimeters			inches[1]		
	Min	Typ	Max	Min	Typ	Max
A	0.450	0.550	0.600	0.0177	0.0217	0.0236
A1	0.000	0.020	0.050	0.0000	0.0008	0.0020
b[2]	0.200	0.250	0.300	0.0079	0.0098	0.0118
D	1.900	2.000	2.100	0.0748	0.0787	0.0827
D2	1.200	-	1.600	0.0472	-	0.0630
E	2.900	3.000	3.100	0.1142	0.1181	0.1220
E2	1.200	-	1.600	0.0472	-	0.0630

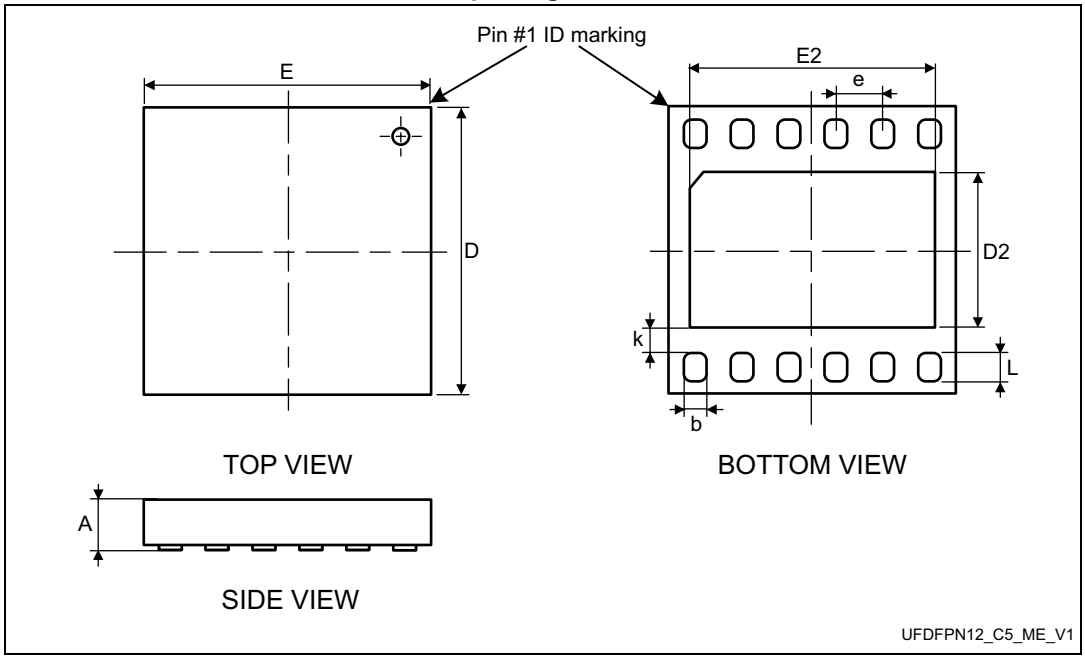
Table 215. UFDFN8 - 8-lead, 2 × 3 mm, 0.5 mm pitch ultra thin profile fine pitch dual flat package mechanical data (continued)

Symbol	millimeters			inches[1]		
	Min	Typ	Max	Min	Typ	Max
e	-	0.500	-		0.0197	
K	0.300	-	-	0.0118	-	-
L	0.300	-	0.500	0.0118	-	0.0197
L1	-	-	0.150	-	-	0.0059
L3	0.300	-	-	0.0118	-	-
aaa	-	-	0.150	-	-	0.0059
bbb	-	-	0.100	-	-	0.0039
ccc	-	-	0.100	-	-	0.0039
ddd	-	-	0.050	-	-	0.0020
eee[3]	-	-	0.080	-	-	0.0031

1. Values in inches are converted from mm and rounded to 4 decimal digits.
2. Dimension b applies to plated terminal and is measured between 0.15 and 0.30 mm from the terminal tip.
3. Applied for exposed die paddle and terminals. Exclude embedding part of exposed die paddle from measuring.

10.4 UFDFPN12 package information

Figure 80. UFDFPN12 - 12-lead, 3x3 mm, 0.5 mm pitch ultra thin profile fine pitch dual flat package outline



1. Drawing is not to scale.
2. Preliminary drawing.

Table 216. UFDFPN12 - 12-lead, 3x3 mm, 0.5 mm pitch ultra thin profile fine pitch dual flat package mechanical data(1)

Symbol	millimeters			inches(2)		
	Min	Typ	Max	Min	Typ	Max
A(3)	0.45	0.55	0.60	0.0177	0.0217	0.0236
b	0.20	0.25	0.30	0.0079	0.0098	0.0118
D	2.95	3.00	3.10	0.1161	0.1181	0.1220
D2	1.35	1.40	1.45	0.0531	0.0551	0.0571
e		0.50			0.0197	
E	2.95	3.00	3.10	0.1161	0.1181	0.1220
E2	2.50	2.55	2.60	0.0984	0.1004	0.1024
L	0.25	0.30	0.35	0.0098	0.0118	0.0138
k		0.40			0.0157	

1. Preliminary data.
2. Values in inches are converted from mm and rounded to 4 decimal digits.
3. Package total thickness.

11 Part numbering

Table 217. Ordering information scheme for packaged devices

Example: ST25DV 64K -JF R 6 D 3

Device type

ST25DV = Dynamic NFC/RFID tag based on ISO 15693 and NFC T5T

Memory size

04K = 4 Kbits

16K = 16 Kbits

64K = 64 Kbits

Device Features

IE = I^2C & GPO Open Drain, Fast Transfer Mode & Energy Harvesting

JF = I^2C & GPO CMOS, Fast Transfer Mode, Energy Harvesting & Low power mode

Operating voltage

R = V_{CC} = 1.8 to 5.5 V

Device grade

6 = industrial: device tested with standard test flow over - 40 to 85 °C

Package

D = UFDFPN12

S = SO8N

T = TSSOP8

C = UFDFPN8

G = 120um +/- 15 um bumped sawn wafer

Capacitance

3 = 28.5 pF

Note: Parts marked as "ES", "E" or accompanied by an Engineering Sample notification letter, are not yet qualified and therefore not yet ready to be used in production and any consequences deriving from such usage will not be at ST charge. In no event, ST will be liable for any customer usage of these engineering samples in production. ST Quality has to be contacted prior to any decision to use these Engineering samples to run qualification activity.

Appendix A Bit representation and coding for fast commands

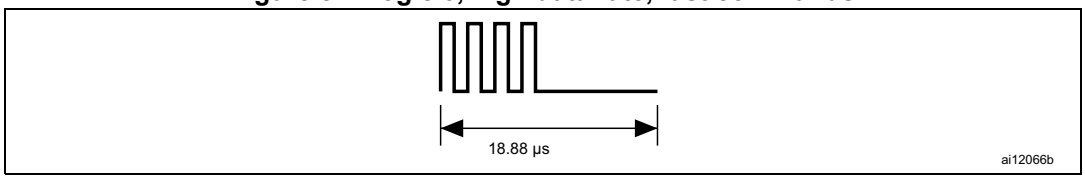
Data bits are encoded using Manchester coding, according to the following schemes. For the low data rate, same subcarrier frequency or frequencies is/are used. In this case, the number of pulses is multiplied by 4 and all times increase by this factor. For the Fast commands using one subcarrier, all pulse numbers and times are divided by 2.

A.1 Bit coding using one subcarrier

A.1.1 High data rate

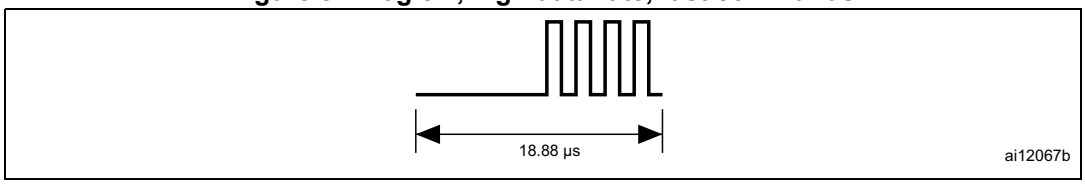
For the fast commands, a logic 0 starts with four pulses at 423.75 kHz ($f_C/32$) followed by an unmodulated time of 9.44 µs, as shown in *Figure 81*.

Figure 81. Logic 0, high data rate, fast commands



For the Fast commands, a logic 1 starts with an unmodulated time of 9.44 µs followed by four pulses of 423.75 kHz ($f_C/32$), as shown in *Figure 82*.

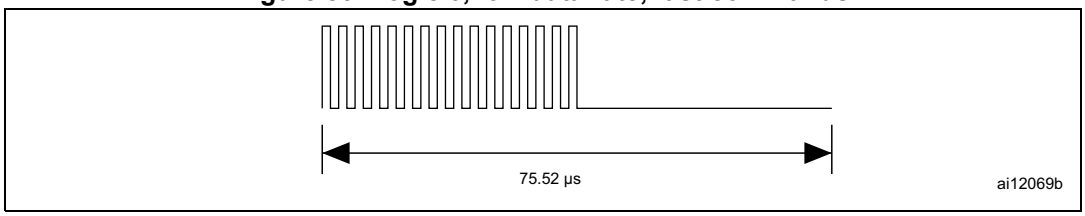
Figure 82. Logic 1, high data rate, fast commands



A.1.2 Low data rate

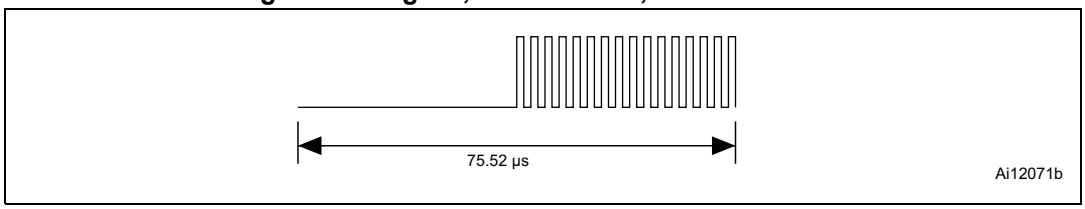
For the Fast commands, a logic 0 starts with 16 pulses at 423.75 kHz ($f_C/32$) followed by an unmodulated time of 37.76 µs, as shown in *Figure 83*.

Figure 83. Logic 0, low data rate, fast commands



For the Fast commands, a logic 1 starts with an unmodulated time of 37.76 µs followed by 16 pulses at 423.75 kHz ($f_C/32$), as shown in *Figure 84*.

Figure 84. Logic 1, low data rate, fast commands



Note: For fast commands, bit coding using two subcarriers is not supported.

A.2 ST25DVxxx to VCD frames

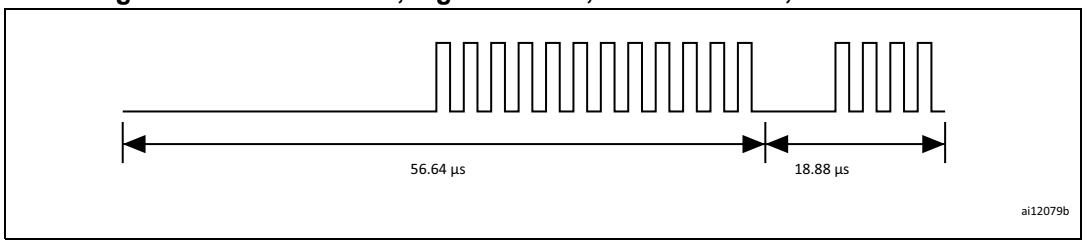
Frames are delimited by an SOF and an EOF. They are implemented using code violation. Unused options are reserved for future use. For the low data rate, the same subcarrier frequency or frequencies is/are used. In this case, the number of pulses is multiplied by 4. For the Fast commands using one subcarrier, all pulse numbers and times are divided by 2.

A.3 SOF when using one subcarrier

A.3.1 High data rate

For the Fast commands, the SOF comprises an unmodulated time of 28.32 µs, followed by 12 pulses at 423.75 kHz (f_C/32), and a logic 1 that consists of an unmodulated time of 9.44 µs followed by four pulses at 423.75 kHz, as shown in *Figure 85*.

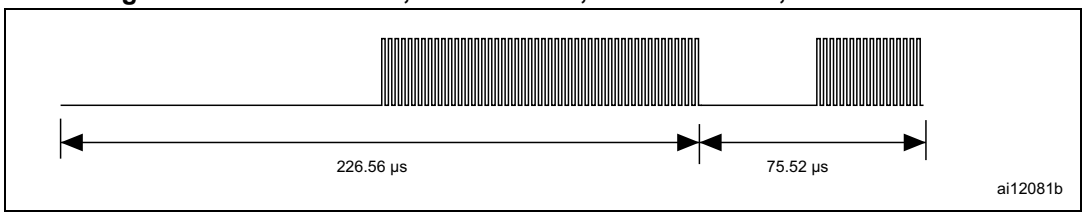
Figure 85. Start of frame, high data rate, one subcarrier, fast commands



A.3.2 Low data rate

For the Fast commands, the SOF comprises an unmodulated time of 113.28 µs, followed by 48 pulses at 423.75 kHz (f_C/32), and a logic 1 that includes an unmodulated time of 37.76 µs followed by 16 pulses at 423.75 kHz, as shown in *Figure 86*.

Figure 86. Start of frame, low data rate, one subcarrier, fast commands

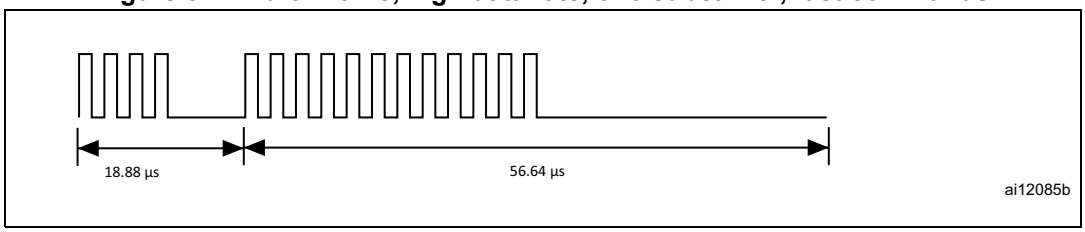


A.4 EOF when using one subcarrier

A.4.1 High data rate

For the Fast commands, the EOF comprises a logic 0 that includes four pulses at 423.75 kHz and an unmodulated time of 9.44 µs, followed by 12 pulses at 423.75 kHz (f_C/32) and an unmodulated time of 37.76 µs, as shown in *Figure 87*.

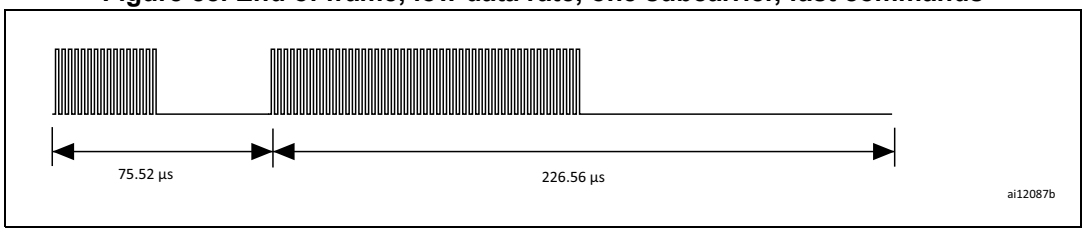
Figure 87. End of frame, high data rate, one subcarrier, fast commands



A.4.2 Low data rate

For the Fast commands, the EOF comprises a logic 0 that includes 16 pulses at 423.75 kHz and an unmodulated time of 37.76 µs, followed by 48 pulses at 423.75 kHz (f_C/32) and an unmodulated time of 113.28 µs, as shown in *Figure 88*.

Figure 88. End of frame, low data rate, one subcarrier, fast commands



Note: For SOF and EOF in fast commands, bit coding using two subcarriers is not supported.

Appendix B I²C sequences

B.1 Device select codes

Table 218. ST25DVxxx Device select usage

millimeters		Comment
Hexadecimal	Binary	
-	1010 E2 11 R/$\overline{W}$	Dev select generic E2 = 0b User memory, Dynamic registers, FTM mailbox E2 = 1b System memory
A6h	1010 0110b	User memory, Dynamic registers, FTM mailbox writing
A7h	1010 0111b	User memory, Dynamic registers, FTM mailbox reading
AEh	1010 0110b	System memory writing
AFh	1010 0111b	System memory reading

B.2 I²C Byte writing and polling

B.2.1 I²C byte write in user memory

Table 219. Byte Write in user memory when write operation allowed

Request/Response Frame		Comment
Master drives SDA	Slave drives SDA	
Start A6h	-	Device select for writing
-	ACK	9th bit
ADDRESS_MSB	-	Send Address MSB (1 Byte)
-	ACK	9th bit
ADDRESS_LSB	-	Send Address LSB (1 Byte)
-	ACK	9th bit
DATA	-	Send Data (1 Byte)
-	NoACK	9th bit
Stop	-	Start of Programming

Table 220. Polling during programming after byte writing in user memory

Request/Response Frame		Comment
Master drives SDA	**Slave drives SDA**	
Start A6h	-	Device select for writing
-	NoACK	9th bit Device Busy
Start A6h	-	Device select for writing
-	NoACK	9th bit Device Busy
...		... Device select for writing
...	...	... 9th bit Device Busy
Start A6h	-	Device select for writing
-	ACK	9th bit Device ready Programing completed
Stop	-	End of Polling

Table 221. Byte Write in user memory when write operation is not allowed

Request/Response Frame		Comment
Master drives SDA	**Slave drives SDA**	
Start A6h	-	Device select for writing
-	ACK	9th bit
ADDRESS_MSB	-	Send Address MSB (1 Byte)
-	ACK	9th bit
ADDRESS_LSB	-	Send Address LSB (1 Byte)
-	ACK	9th bit
DATA	-	Send Data
-	ACK	9th bit: Write access not granted or FTM activated.
Stop	-	No Programming Device return in Standby

B.2.2 I²C byte writing in dynamic registers and polling

Table 222. Byte Write in Dynamic Register (if not Read Only)

Request/Response Frame		Comment
Master drives SDA	**Slave drives SDA**	
Start A6h	-	Device select for writing
-	ACK	9th bit
ADDRESS_MSB	-	Send Address MSB (1 Byte)
-	ACK	9th bit
Dynamic Register ADDRESS_LSB	-	Send Address LSB (1 Byte) Dynamic register are located from address 2000h to 2007h , some are only readable
-	ACK	9th bit
DATA	-	Send Data
-	ACK	9th bit
Stop	-	**Immediate** update of Dynamic register

Table 223. Polling during programming after byte write in Dynamic Register

Request/Response Frame		Comment
Master drives SDA	**Slave drives SDA**	
Start A6h	-	Device select for writing
-	ACK	9th bit Device Busy Dynamic register updates is immediate
Stop	-	End of Polling

Table 224. Byte Write in Dynamic Register if Read Only

Request/Response Frame		Comment
Master drives SDA	**Slave drives SDA**	
Start A6h	-	Device select for writing
-	ACK	9th bit
20h	-	Send Address MSB (1 Byte)
-	NoACK	9th bit

Table 224. Byte Write in Dynamic Register if Read Only (continued)

Request/Response Frame		Comment
Master drives SDA	**Slave drives SDA**	
RO Dynamic Register ADDRESS_LSB	-	Send Address LSB (1 Byte) Addresses 2001h, 2004h, 2005h and 2007h are Read Only registers.
-	ACK	9th bit
DATA	-	Send Data
-	NoACK	9th bit
Stop	-	No Programming Device return in Standby

B.2.3 I²C byte write in mailbox and polling

Table 225. Byte Write in mailbox when mailbox is free from RF message and Fast transfer Mode is activated

Request/Response Frame		Comment
Master drives SDA	**Slave drives SDA**	
Start A6h	-	Device select for writing
-	ACK	9th bit
20h	-	Send mailbox address MSB (1 Byte)
-	ACK	9th bit
08h	-	Send Address LSB (1 Byte) Write must be done at first address of mailbox
-	ACK	9th bit
DATA	-	Send Data
-	ACK	9th bit
Stop	-	**Immediate** update of mailbox

Table 226. Byte Write in mailbox when mailbox is not free from RF message Fast transfer Mode is not activated

Request/Response Frame		Comment
Master drives SDA	**Slave drives SDA**	
Start A6h	-	Device select for writing
-	ACK	9th bit
20h	-	Send mailbox address MSB (1 Byte)
-	ACK	9th bit
08h	-	Send Address LSB (1 Byte) Write must be done at first address of mailbox
-	ACK	9th bit
DATA	-	Send Data
-	NoACK	9th bit Access Mailbox busy or FTM not activated
Stop	-	No Programming Device return in Standby

B.2.4 I²C byte write and polling in system memory

Table 227. Byte Write in System memory if I²C security session is open and register is not RO

Request/Response Frame		Comment
Master drives SDA	**Slave drives SDA**	
Start AEh	-	Device select for writing
-	ACK	9th bit
ADDRESS_MSB	-	Send Address MSB (1 Byte)
-	ACK	9th bit
ADDRESS_LSB	-	Send Address LSB (1 Byte)
-	ACK	9th bit
DATA	-	Send Data
-	ACK	9th bit
Stop	-	Start of Programming

Table 228. Polling during programing after byte write in System memory if I^2C security session is open and register is not RO

Request/Response Frame		Comment
Master drives SDA	**Slave drives SDA**	
Start AEh	-	Device select for writing
-	NoACK	9th bit Device Busy
Start AEh	-	Device select for writing
-	NoACK	9th bit Device Busy
Start AEh	-	Device select for writing
-	...	9th bit
Start AEh	-	Device select for writing
-	ACK	9th bit Device ready Programing completed
Stop	-	end of Polling

Table 229. Byte Write in System memory if I^2C security session is closed or register is RO

Request/Response Frame		Comment
Master drives SDA	**Slave drives SDA**	
Start AEh	-	Device select for writing
-	ACK	9th bit
ADDRESS_MSB	-	Send Address MSB (1 Byte)
-	ACK	9th bit
ADDRESS_LSB	-	Send Address LSB (1 Byte)
-	ACK	9th bit
DATA	-	Send Data
-	NoACK	9th bit
Stop	-	No Programming Device return in Standby

B.3 I²C sequential writing and polling

B.3.1 I²C sequential write in user memory and polling

Table 230. Sequential write User memory when write operation allowed and all bytes belong to same area

Request/Response Frame		Comment
Master drives SDA	**Slave drives SDA**	
Start A6h	-	Device select for writing
-	ACK	9th bit
ADDRESS_MSB	-	Send Address MSB (1 Byte)
-	ACK	9th bit
ADDRESS_LSB	-	Send Address LSB (1 Byte)
-	ACK	9th bit
DATA 0	-	Send Data 0
-	ACK	9th bit
DATA 1	-	Send Data 1
-	ACK	9th bit
...	-	...
-	...	...
DATA n	-	Send Data n n ≤ 256
-	ACK	9th bit
Stop	-	Start of Programming

Table 231. Polling during programing after sequential write in User memory when write operation allowed and all bytes belong to same area.

Request/Response Frame		Comment
Master drives SDA	**Slave drives SDA**	
Start A6h	-	Device select for writing
-	NoACK	9th bit Device Busy
Start A6h	-	Device select for writing
-	NoACK	9th bit Device Busy

Table 231. Polling during programing after sequential write in User memory when write operation allowed and all bytes belong to same area. (continued)

Request/Response Frame		Comment
Master drives SDA	**Slave drives SDA**	
Start A6h	-	Device select for writing
-	...	9th bit Device Busy
Start A6h	-	Device select for writing
-	ACK	9th bit Device ready Programing completed
Stop	-	End of Polling

Table 232. Sequential write in User memory when write operation allowed and crossing over area border

Request/Response Frame		Comment
Master drives SDA	**Slave drives SDA**	
Start A6h	-	Device select for writing
-	ACK	9th bit
ADDRESS_MSB	-	Send Address MSB (1 Byte)
-	ACK	9th bit
ADDRESS_LSB	-	Send Address LSB (1 Byte)
-	ACK	9th bit
DATA 0	-	Send Data 0
-	ACK	9th bit
DATA 1	-	Send Data 1
-	ACK	9th bit
...	-	...
-	...	...
DATA n	-	Send Data n Address is located in next memory area
-	**NoACK**	9th bit
Stop	-	**No programming** Device return in Standby

Table 233. Polling during programing after sequential write in User memory when write operation allowed and crossing over area border

Request/Response Frame		Comment
Master drives SDA	**Slave drives SDA**	
Start A6h	-	Device select for writing
-	ACK	9th bit Device ready No programming
Stop	-	End of Polling

B.3.2 I²C sequential write in mailbox and polling

Table 234. Sequential write in mailbox when mailbox is free from RF message and Fast transfer Mode is activated

Request/Response Frame		Comment
Master drives SDA	**Slave drives SDA**	
Start A6h	-	Device select for writing
-	ACK	9th bit
ADDRESS_MSB	-	Send mailbox Address MSB (1 Byte)
-	ACK	9th bit
ADDRESS_LSB	-	Send mailbox Address LSB (1 Byte)
-	ACK	9th bit
DATA 0	-	Send Data 0
-	ACK	9th bit
DATA 1	-	Send Data 1
-	ACK	9th bit
...	-	...
-	...	...
DATA n	-	Send Data n n ≤ 256
-	ACK	9th bit
Stop	-	**Immediate** mailbox content update

Table 235. Polling during programing after sequential write in mailbox

Request/Response Frame		Comment
Master drives SDA	**Slave drives SDA**	
Start A6h	-	Device select for writing
-	ACK	9th bit Device ready Mailbox is immediately updated
Stop	-	End of Polling

B.4 I²C Read current address

B.4.1 I²C current address read in User memory

Table 236. Current byte Read in User memory if read operation allowed (depending on area protection and RF user security session)

Request/Response Frame		Comment
Master drives SDA	**Slave drives SDA**	
Start A7h	-	Device select for reading
-	ACK	9th bit
	DATA	Receive Data located on last pointed address+1, or at address 0 after power-up, in user memory
NO_ACK	-	9th bit
Stop	-	End of Reading

Table 237. Current Read in User memory if read operation not allowed (depending on area protection and RF user security session)

Request/Response Frame		Comment
Master drives SDA	**Slave drives SDA**	
Start A7h	-	Device select for reading
-	ACK	9th bit
	FFh	Read of data not allowed ST25DV release SDA
NO_ACK		9th bit
Stop	-	End of Reading

B.5 I²C random address read

B.5.1 I²C random address read in system memory

Table 238. Random byte read in User memory if read operation allowed (depending on area protection and RF user security session)

Request/Response Frame		Comment
Master drives SDA	**Slave drives SDA**	
Start A6h	-	Device select for writing
-	ACK	9th bit
ADDRESS_MSB	-	Send Address MSB (1 Byte)
-	ACK	9th bit
ADDRESS_LSB	-	Send Address LSB (1 Byte)
-	ACK	9th bit
Start A7h	-	Device select for reading
-	ACK	9th bit
-	DATA	Receive Data
NO_ACK	-	9th bit
Stop	-	End of Reading

Table 239. Random byte read in User memory if operation not allowed (depending on area protection and RF user security)

Request/Response Frame		Comment
Master drives SDA	**Slave drives SDA**	
Start A6h	-	Device select for writing
-	ACK	9th bit
ADDRESS_MSB	-	Send Address MSB (1 Byte)
-	ACK	9th bit
ADDRESS_LSB	-	Send Address LSB (1 Byte)
-	ACK	9th bit
Start A7h	-	Device select for reading
-	ACK	9th bit
-	**FFh**	Read of data not allowed ST25DVxxx release SDA
NO_ACK	-	9th bit
Stop	-	End of Reading

B.5.2 I²C Random address read in system memory

Table 240. Byte Read System memory
(Static register or I2C Password after a valid Present I2C Password)

Request/Response Frame		Comment
Master drives SDA	**Slave drives SDA**	
Start AEh	-	Device select for writing
-	ACK	9th bit
ADDRESS_MSB	-	Send Address MSB (1 Byte)
-	ACK	9th bit
ADDRESS_LSB	-	Send Address LSB (1 Byte)
-	ACK	9th bit
Start AFh	-	Device select for reading
-	ACK	9th bit
-	DATA	Receive Data
NO_ACK	-	9th bit
Stop	-	End of reading

B.5.3 I²C Random address read in dynamic registers

Table 241. Random byte read in Dynamic registers

Request/Response Frame		Comment
Master drives SDA	**Slave drives SDA**	
Start A6h	-	Device select for writing
-	ACK	9th bit
20h	-	Send Address MSB (1 Byte)
-	ACK	9th bit
ADDRESS_LSB	-	Send Adress LSB (1 Byte)
-	ACK	9th bit
Start A7h	-	Device select for reading
-	ACK	9th bit
-	DATA	Receive Data
NO_ACK	-	9th bit
Stop	-	End of reading

B.6 I²C sequential read

B.6.1 I²C sequential read in user memory

Table 242. Sequential Read User memory if read operation allowed (depending on area protection and RF user security session) and all bytes belong to the same area

Request/Response Frame		Comment
Master drives SDA	**Slave drives SDA**	
Start A6h	-	Device select for writing
-	ACK	9th bit
ADDRESS_MSB	-	Send Address MSB (1 Byte)
-	ACK	9th bit
ADDRESS_LSB	-	Send Address LSB (1 Byte)
-	ACK	9th bit
Start A7h0	-	Device select for reading
-	ACK	9th bit
-	DATA 0	Receive Data 0
ACK	-	9th bit
-	DATA 1	Receive Data 1
ACK	-	9th bit
-	...	...
...	-	...
-	DATA n	Receive Data n
NO_ACK	-	9th bit
Stop	-	End of Reading

Table 243. Sequential Read User memory if read operation allowed (depending on area protection and RF user security session) but crossing area border

Request/Response Frame		Comment
Master drives SDA	**Slave drives SDA**	
Start A6h	-	Device select for writing
-	ACK	9th bit
ADDRESS_MSB	-	Send Address MSB (1 Byte)
-	ACK	9th bit
ADDRESS_LSB	-	Send Address LSB (1 Byte)
-	ACK	9th bit

Table 243. Sequential Read User memory if read operation allowed (depending on area protection and RF user security session) but crossing area border (continued)

Request/Response Frame		Comment
Master drives SDA	**Slave drives SDA**	
Start A7h	-	Device select for reading
-	ACK	9th bit
-	DATA 0	Receive Data 0
ACK	-	9th bit
-	DATA 1	Receive Data 1
ACK	-	9th bit
-	...	...
...	-	...
-	DATA n	Receive Data last Address available
ACK	-	9th bit
-	FFh	Data is located in next memory area ST25DV release SDA
ACK	-	9th bit
-	...	...
...	-	...
-	FFh	Data is located in next memory area ST25DV release SDA
Stop	-	End of reading

Table 244. Sequential Read User memory if read operation allowed (depending on area protection and RF user security session)

Request/Response Frame		Comment
Master drives SDA	**Slave drives SDA**	
Start A6h	-	Device select for writing
-	ACK	9th bit
ADDRESS_MSB	-	Send Address MSB (1 Byte)
-	ACK	9th bit
ADDRESS_LSB	-	Send Address LSB (1 Byte)
-	ACK	9th bit
Start A7h	-	Device select for reading
-	ACK	9th bit
-	FFh	ST25DV release SDA Reading access not granted

Table 244. Sequential Read User memory if read operation allowed (depending on area protection and RF user security session) (continued)

Request/Response Frame		Comment
Master drives SDA	**Slave drives SDA**	
ACK	-	9th bit
-	...	...
...	-	...
-	FFh	ST25DV release SDA Reading access not granted
NO_ACK	-	9th bit
Stop	-	End of reading

B.6.2 I^2C sequential read in system memory

Table 245. Sequential in Read System memory (I^2C security session open if reading I2C_PWD)

Request/Response Frame		Comment
Master drives SDA	**Slave drives SDA**	
Start AEh	-	Device select for writing
-	ACK	9th bit
ADDRESS_MSB	-	Send Address MSB (1 Byte)
-	ACK	9th bit
ADDRESS_LSB	-	Send Address LSB (1 Byte)
-	ACK	9th bit
Start AF7h	-	Device select for reading
-	ACK	9th bit
-	DATA	Receive Data 0
ACK	-	9th bit
-	DATA	Receive Data 1
ACK	-	9th bit
-	...	...
...	-	...
-	DATA	Receive Data n
NO_ACK	-	9th bit
Stop	-	End of Reading

Table 246. Sequential Read system memory when access is not granted (I^2C password I2C_PWD)

Request/Response Frame		Comment
Master drives SDA	**Slave drives SDA**	
Start AEh	-	Device select for writing
-	ACK	9th bit
90h	-	Send Address MSB (1 Byte)
-	ACK	9th bit
ADDRESS_LSB	-	Send Address LSB (1 Byte)
-	ACK	9th bit
Start AFh	-	Device select for reading
-	ACK	9th bit
-	DATA	Receive Data 0
-	FFh	ST25DV release SDA Reading access is not granted
ACK	-	9th bit
-	...	...
...	-	...
-	FFh	ST25DV release SDA Reading access is not granted
NO_ACK	-	9th bit
Stop	-	End of reading

B.6.3 I^2C sequential read in dynamic registers

Table 247. Sequential read in dynamic register

Request/Response Frame		Comment
Master drives SDA	**Slave drives SDA**	
Start A6h	-	Device select for writing
-	ACK	9th bit
20h	-	Send Address MSB (1 Byte)
-	ACK	9th bit
Dynamic register ADDRESS_LSB	-	Send Address LSB (1 Byte) Fynamic register are located form address 2000h to 2007
-	ACK	9th bit
Start A7h	-	Device select for reading
-	ACK	9th bit

Table 247. Sequential read in dynamic register (continued)

Request/Response Frame		Comment
Master drives SDA	**Slave drives SDA**	
-	DATA	Receive Data 0
ACK	-	9th bit
-	DATA	Receive Data 1
ACK	-	9th bit
-	...	...
...	-	...
-	Data	Receive Data n
NO_ACK	-	9th bit
Stop	-	End of reading

Table 248. Sequential read in Dynamic register and mailbox continuously if Fast Transfer Mode is activated

Request/Response Frame		Comment
Master drives SDA	**Slave drives SDA**	
Start A6h	-	Device select for writing
-	ACK	9th bit
20h	-	Send Address MSB (1 Byte)
-	ACK	9th bit
Dynamic Register ADDRESS_LSB	-	Send Address LSB (1 Byte) Dynamic register are located from address 2000h to 2007h
-	ACK	9th bit
Start A7h	-	Device select for reading
-	ACK	9th bit
-	DATA 0	Receive Data 0
ACK	-	9th bit
-	DATA 1	Receive Data 1
ACK	-	9th bit
-	...	...
...	-	...
-	DATA n	Receive Data n (n ≤ 8) Last Dynamic register address 2007h
ACK	-	9th bit
-	DATA n + 1	Mailbox byte 0

Table 248. Sequential read in Dynamic register and mailbox continuously if Fast Transfer Mode is activated (continued)

Request/Response Frame		Comment
Master drives SDA	**Slave drives SDA**	
ACK	-	9th bit
-	DATA n + 2	Mailbox byte 1
ACK	-	9th bit
-	...	...
...	-	...
-	Data n + i	Mailbox byte i (i < 256)
NO_ACK	-	9th bit
Stop	-	End of reading

B.6.4 I²C sequential read in mailbox

Table 249. Sequential in mailbox if Fast Transfer Mode is activated

Request/Response Frame		Comment
Master drives SDA	**Slave drives SDA**	
Start A6h	-	Device select for writing
-	ACK	9th bit
20h or 21h	-	Send Address MSB (1 Byte) 2007h < @ 2108h
-	ACK	9th bit
ADDRESS_LSB	-	Send Address LSB (1 Byte) 2007h < @ 2108h
-	ACK	9th bit
Start A7h	-	Device select for reading
-	ACK	9th bit
-	DATA 0	Receive Data 0
ACK	-	9th bit
-	DATA 1	Receive Data 1
ACK	-	9th bit
-	...	...
...	-	...
-	Data n	Receive Data n
NO_ACK	-	9th bit
Stop	-	End of reading

Table 250. Sequential read in mailbox if Fast Transfer Mode is not activated

Request/Response Frame		Comment
Master drives SDA	**Slave drives SDA**	
Start A6h	-	Device select for writing
-	ACK	9th bit
20h or 21h	-	Send Address MSB (1 Byte) 2007h < @ 2108h
-	ACK	9th bit
ADDRESS_LSB	-	Send Address LSB (1 Byte) 2007h < @ 2108h
-	ACK	9th bit
Start A7h	-	Device select for reading
-	ACK	9th bit
-	FFh	ST25DVxxx release SDA
ACK	-	9th bit
-	FFh	ST25DVxxx release SDA
ACK	-	9th bit
-	...	...
...	-	...
-	FFh	ST25DVxxx release SDA
NO_ACK	-	9th bit
Stop	-	End of reading

B.7 I²C password relative sequences

B.7.1 I²C write password

Table 251. Write Password when I²C security session is already open and Fast Transfer Mode is not activated

Request/Response Frame		Comment
Master drives SDA	**Slave drives SDA**	
Start AEh	-	Device select for writing
-	ACK	9th bit
09h	-	Send I2C_PWD MSB address
-	ACK	9th bit
00h	-	Send I2C_PWD LSB address
-	ACK	9th bit
I2C_PWD_BYTE_7	-	Send I2C_PWD MSB
-	ACK	9th bit
I2C_PWD_BYTE_6	DATA 0	Send Data
-	ACK	9th bit
...	-	...
-	...	...
I2C_PWD_BYTE_0	-	Send I2C_PWD LSB
-	ACK	9th bit
07h	-	Write password command
-	ACK	9th bit
I2C_PWD_BYTE_7	-	Send I2C_PWD MSB
-	ACK	9th bit
I2C_PWD_BYTE_6	DATA 0	Send Data
-	ACK	9th bit
...	-	...
-	...	...
I2C_PWD_BYTE_0	-	Send I2C_PWD LSB
-	ACK	9th bit
Stop	-	Start of I²C password programming

Table 252. Write Password when I²C security session is not open or Fast Transfer Mode activated

Request/Response Frame		Comment
Master drives SDA	**Slave drives SDA**	
Start AEh	-	Device select for writing
-	ACK	9th bit
09h	-	Send I2C_PWD MSB address
-	ACK	9th bit
00h	-	Send I2C_PWD LSB address
-	NoACK	9th bit
Stop	-	No PWD Programming Device return in Standby

B.7.2 I²C present password

Present Password (whatever status of I²C security session or Fast Transfer Mode)

Request/Response Frame		Comment
Master drives SDA	**Slave drives SDA**	
Start AEh	-	Device select for writing
-	ACK	9th bit
09h	-	Send I2C_PWD MSB address
-	ACK	9th bit
00h	-	Send I2C_PWD LSB address
-	ACK	9th bit
I2C_PWD_BYTE_7	-	Send I2C_PWD MSB
-	ACK	9th bit
I2C_PWD_BYTE_6	DATA 0	Send Data
-	ACK	9th bit
...	-	...
-	...	...
I2C_PWD_BYTE_0	-	Send I2C_PWD LSB
-	ACK	9th bit
09h	-	Present password command
-	ACK	9th bit
I2C_PWD_BYTE_7	-	Send I2C_PWD MSB
-	ACK	9th bit

Present Password (whatever status of I^2C security session or Fast Transfer Mode) (continued)

Request/Response Frame		Comment
Master drives SDA	**Slave drives SDA**	
I2C_PWD_BYTE_6	-	Send Data
-	ACK	9th bit
...	-	...
-	...	...
I2C_PWD_BYTE_0	-	Send I2C_PWD LSB
-	ACK	9th bit
Stop	-	ST25DV with active I2C_PWD. Result is immediate.

Revision history

Table 253. Document revision history

Date	Revision	Changes
23-Feb-2017	1	Initial release.

IMPORTANT NOTICE – PLEASE READ CAREFULLY

STMicroelectronics NV and its subsidiaries (“ST”) reserve the right to make changes, corrections, enhancements, modifications, and improvements to ST products and/or to this document at any time without notice. Purchasers should obtain the latest relevant information on ST products before placing orders. ST products are sold pursuant to ST’s terms and conditions of sale in place at the time of order acknowledgement.

Purchasers are solely responsible for the choice, selection, and use of ST products and ST assumes no liability for application assistance or the design of Purchasers’ products.

No license, express or implied, to any intellectual property right is granted by ST herein.

Resale of ST products with provisions different from the information set forth herein shall void any warranty granted by ST for such product.

ST and the ST logo are trademarks of ST. All other product or service names are the property of their respective owners.

Information in this document supersedes and replaces information previously supplied in any prior versions of this document.

© 2017 STMicroelectronics – All rights reserved