



M24LR64-R

64 Kbit EEPROM with password protection & dual interface: 400kHz I²C serial bus & ISO 18000-3 mode 1 RF protocol at 13.56MHz

Preliminary data

Features

I²C interface

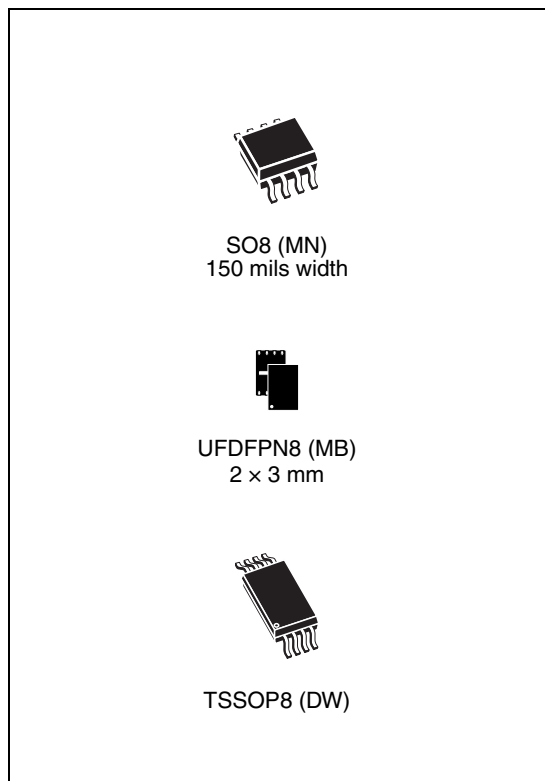
- Two-wire I²C serial interface supports 400 kHz protocol
- Single supply voltage:
 - 1.8 V to 5.5 V
- Byte and Page Write (up to 4 bytes)
- Random and Sequential Read modes
- Self-timed programming cycle
- Automatic address incrementing
- Enhanced ESD/latch-up protection

Contactless interface

- ISO 15693 and ISO 18000-3 mode 1 compliant
- 13.56 MHz \pm 7k Hz carrier frequency
- To tag: 10% or 100% ASK modulation using 1/4 (26 Kbit/s) or 1/256 (1.6 Kbit/s) pulse position coding
- From tag: load modulation using Manchester coding with 423 kHz and 484 kHz subcarriers in low (6.6 Kbit/s) or high (26 Kbit/s) data rate mode. Supports the 53 Kbit/s data rate with Fast commands
- Internal tuning capacitance: 28.5 pF
- 64-bit unique identifier (UID)
- Read Block & Write (32-bit Blocks)

Memory

- 64 Kbit EEPROM organized into:
 - 8192 bytes in I²C mode
 - 2048 blocks of 32 bits in RF mode
- Write time
 - I²C: 5 ms (Max.)
 - RF: 5.75 ms including the internal Verify time



- More than 1 Million erase/write cycles in I²C mode
- More than 100 000 erase/write cycles in RF mode
- Multiple password protection in RF mode
- Single password protection in I²C mode
- More than 40-year data retention
- Package
 - ECOPACK2[®] (RoHS compliant and Halogen-free)

Contents

| | | |
|----------|---|-----------|
| 1 | Description | 12 |
| 2 | Signal description | 14 |
| 2.1 | Serial Clock (SCL) | 14 |
| 2.2 | Serial Data (SDA) | 14 |
| 2.3 | Chip Enable (E0, E1) | 14 |
| 2.4 | Antenna coil (AC0, AC1) | 14 |
| 2.5 | V _{SS} ground | 14 |
| 2.6 | Supply voltage (V _{CC}) | 15 |
| 2.6.1 | Operating supply voltage V _{CC} | 15 |
| 2.6.2 | Power-up conditions | 15 |
| 2.6.3 | Device reset | 15 |
| 2.6.4 | Power-down conditions | 15 |
| 3 | User memory organization | 17 |
| 4 | System memory area | 22 |
| 4.1 | M24LR64-R RF block security | 22 |
| 4.2 | Example of the M24LR64-R security protection | 24 |
| 4.3 | I2C_Write_Lock bit area | 25 |
| 4.4 | System parameters | 25 |
| 4.5 | M24LR64-R I ² C password security | 26 |
| 4.5.1 | I ² C Present Password command description | 26 |
| 4.5.2 | I ² C Write Password command description | 27 |
| 5 | I²C device operation | 29 |
| 5.1 | Start condition | 29 |
| 5.2 | Stop condition | 29 |
| 5.3 | Acknowledge bit (ACK) | 29 |
| 5.4 | Data Input | 29 |
| 5.5 | Memory addressing | 30 |
| 5.6 | Write operations | 31 |
| 5.7 | Byte Write | 31 |

| | | |
|-----------|--|-----------|
| 5.8 | Page Write | 31 |
| 5.9 | Minimizing system delays by polling on ACK | 33 |
| 5.10 | Read operations | 35 |
| 5.11 | Random Address Read | 35 |
| 5.12 | Current Address Read | 35 |
| 5.13 | Sequential Read | 35 |
| 5.14 | Acknowledge in Read mode | 35 |
| 6 | Initial delivery state | 36 |
| 7 | RF device operation | 36 |
| 7.1 | Commands | 37 |
| 7.2 | Initial dialog for vicinity cards | 38 |
| 7.2.1 | Power transfer | 38 |
| 7.2.2 | Frequency | 38 |
| 7.2.3 | Operating field | 38 |
| 8 | Communication signal from VCD to M24LR64-R | 39 |
| 9 | Data rate and data coding | 41 |
| 9.1 | Data coding mode: 1 out of 256 | 41 |
| 9.2 | Data coding mode: 1 out of 4 | 42 |
| 9.3 | VCD to M24LR64-R frames | 44 |
| 9.4 | Start of frame (SOF) | 44 |
| 10 | Communications signal from M24LR64-R to VCD | 46 |
| 10.1 | Load modulation | 46 |
| 10.2 | Subcarrier | 46 |
| 10.3 | Data rates | 46 |
| 11 | Bit representation and coding | 47 |
| 11.1 | Bit coding using one subcarrier | 47 |
| 11.1.1 | High data rate | 47 |
| 11.1.2 | Low data rate | 48 |
| 11.2 | Bit coding using two subcarriers | 49 |
| 11.3 | High data rate | 49 |

| | | |
|-----------|---|-----------|
| 11.4 | Low data rate | 49 |
| 12 | M24LR64-R to VCD frames | 50 |
| 12.1 | SOF when using one subcarrier | 50 |
| 12.2 | High data rate | 50 |
| 12.3 | Low data rate | 50 |
| 12.4 | SOF when using two subcarriers | 51 |
| 12.5 | High data rate | 51 |
| 12.6 | Low data rate | 51 |
| 12.7 | EOF when using one subcarrier | 52 |
| 12.8 | High data rate | 52 |
| 12.9 | Low data rate | 52 |
| 12.10 | EOF when using two subcarriers | 53 |
| 12.11 | High data rate | 53 |
| 12.12 | Low data rate | 53 |
| 13 | Unique identifier (UID) | 54 |
| 14 | Application family identifier (AFI) | 55 |
| 15 | Data storage format identifier (DSFID) | 56 |
| 15.1 | CRC | 56 |
| 16 | M24LR64-R protocol description | 57 |
| 17 | M24LR64-R states | 59 |
| 17.1 | Power-off state | 59 |
| 17.2 | Ready state | 59 |
| 17.3 | Quiet state | 59 |
| 17.4 | Selected state | 59 |
| 18 | Modes | 61 |
| 18.1 | Addressed mode | 61 |
| 18.2 | Non-addressed mode (general request) | 61 |
| 18.3 | Select mode | 61 |

| | | |
|-----------|--|-----------|
| 19 | Request format | 62 |
| | 19.1 Request flags | 62 |
| 20 | Response format | 64 |
| | 20.1 Response flags | 64 |
| | 20.2 Response error code | 65 |
| 21 | Anticollision | 66 |
| | 21.1 Request parameters | 66 |
| 22 | Request processing by the M24LR64-R | 68 |
| 23 | Explanation of the possible cases | 69 |
| 24 | Inventory Initiated command | 71 |
| 25 | Timing definition | 72 |
| | 25.1 t ₁ : M24LR64-R response delay | 72 |
| | 25.2 t ₂ : VCD new request delay | 72 |
| | 25.3 t ₃ : VCD new request delay in the absence of a response from the M24LR64-R | 72 |
| 26 | Commands codes | 73 |
| | 26.1 Inventory | 74 |
| | 26.2 Stay Quiet | 75 |
| | 26.3 Read Single Block | 76 |
| | 26.4 Write Single Block | 78 |
| | 26.5 Read Multiple Block | 80 |
| | 26.6 Select | 82 |
| | 26.7 Reset to Ready | 83 |
| | 26.8 Write AFI | 84 |
| | 26.9 Lock AFI | 86 |
| | 26.10 Write DSFID | 88 |
| | 26.11 Lock DSFID | 90 |
| | 26.12 Get System Info | 92 |
| | 26.13 Get Multiple Block Security Status | 94 |

| | | |
|-------------------|--|------------|
| 26.14 | Write-sector Password | 96 |
| 26.15 | Lock-sector Password | 98 |
| 26.16 | Present-sector Password | 100 |
| 26.17 | Fast Read Single Block | 102 |
| 26.18 | Fast Inventory Initiated | 104 |
| 26.19 | Fast Initiate | 105 |
| 26.20 | Fast Read Multiple Block | 106 |
| 26.21 | Inventory Initiated | 108 |
| 26.22 | Initiate | 109 |
| 27 | Maximum rating | 110 |
| 28 | I²C DC and AC parameters | 111 |
| 29 | RF DC and AC parameters | 115 |
| 30 | Package mechanical data | 117 |
| 31 | Part numbering | 120 |
| Appendix A | Anticollision algorithm (informative) | 121 |
| A.1 | Algorithm for pulsed slots | 121 |
| Appendix B | CRC (informative) | 122 |
| B.1 | CRC error detection method | 122 |
| B.2 | CRC calculation example | 122 |
| Appendix C | Application family identifier (AFI) (informative) | 124 |
| | Revision history | 125 |

List of tables

| | | |
|-----------|---|----|
| Table 1. | Signal names | 13 |
| Table 2. | Device select code | 16 |
| Table 3. | Address most significant byte | 16 |
| Table 4. | Address least significant byte | 16 |
| Table 5. | Sector details | 19 |
| Table 6. | Sector Security Status Byte area | 22 |
| Table 7. | Sector security status byte organization | 22 |
| Table 8. | Read / Write protection bit setting. | 23 |
| Table 9. | Password Control bits. | 23 |
| Table 10. | Password system area | 23 |
| Table 11. | M24LR64-R sector security protection after power-up | 24 |
| Table 12. | M24LR64-R sector security protection after a valid presentation of password 1 | 24 |
| Table 13. | I2C_Write_Lock bit | 25 |
| Table 14. | System parameter sector | 26 |
| Table 15. | Operating modes | 30 |
| Table 16. | 10% modulation parameters | 39 |
| Table 17. | Response data rates. | 46 |
| Table 18. | UID format | 54 |
| Table 19. | CRC transmission rules | 56 |
| Table 20. | VCD request frame format | 57 |
| Table 21. | M24LR64-R Response frame format | 57 |
| Table 22. | M24LR64-R response depending on Request_flags | 60 |
| Table 23. | General request format. | 62 |
| Table 24. | Definition of request flags 1 to 4 | 62 |
| Table 25. | Request flags 5 to 8 when Bit 3 = 0 | 63 |
| Table 26. | Request flags 5 to 8 when Bit 3 = 1 | 63 |
| Table 27. | General response format | 64 |
| Table 28. | Definitions of response flags 1 to 8. | 64 |
| Table 29. | Response error code definition. | 65 |
| Table 30. | Inventory request format. | 66 |
| Table 31. | Example of the addition of 0-bits to an 11-bit mask value | 66 |
| Table 32. | Timing values | 72 |
| Table 33. | Command codes. | 73 |
| Table 34. | Inventory request format. | 74 |
| Table 35. | Inventory response format | 74 |
| Table 36. | Stay Quiet request format | 75 |
| Table 37. | Read Single Block request format | 76 |
| Table 38. | Read Single Block response format when Error_flag is NOT set | 76 |
| Table 39. | Sector security status | 76 |
| Table 40. | Read Single Block response format when Error_flag is set | 76 |
| Table 41. | Write Single Block request format | 78 |
| Table 42. | Write Single Block response format when Error_flag is NOT set | 78 |
| Table 43. | Write Single Block response format when Error_flag is set | 78 |
| Table 44. | Read Multiple Block request format | 80 |
| Table 45. | Read Multiple Block response format when Error_flag is NOT set. | 80 |
| Table 46. | Sector security status | 80 |
| Table 47. | Read Multiple Block response format when Error_flag is set | 81 |
| Table 48. | Select request format | 82 |

| | | |
|------------|---|-----|
| Table 49. | Select Block response format when Error_flag is NOT set. | 82 |
| Table 50. | Select response format when Error_flag is set. | 82 |
| Table 51. | Reset to Ready request format. | 83 |
| Table 52. | Reset to Ready response format when Error_flag is NOT set | 83 |
| Table 53. | Reset to ready response format when Error_flag is set | 83 |
| Table 54. | Write AFI request format. | 84 |
| Table 55. | Write AFI response format when Error_flag is NOT set | 84 |
| Table 56. | Write AFI response format when Error_flag is set | 84 |
| Table 57. | Lock AFI request format | 86 |
| Table 58. | Lock AFI response format when Error_flag is NOT set | 86 |
| Table 59. | Lock AFI response format when Error_flag is set. | 86 |
| Table 60. | Write DSFID request format | 88 |
| Table 61. | Write DSFID response format when Error_flag is NOT set | 88 |
| Table 62. | Write DSFID response format when Error_flag is set. | 88 |
| Table 63. | Lock DSFID request format | 90 |
| Table 64. | Lock DSFID response format when Error_flag is NOT set | 90 |
| Table 65. | Lock DSFID response format when Error_flag is set | 90 |
| Table 66. | Get System Info request format | 92 |
| Table 67. | Get System Info response format when Error_flag is NOT set. | 92 |
| Table 68. | Get System Info response format when Error_flag is set | 92 |
| Table 69. | Get Multiple Block Security Status request format | 94 |
| Table 70. | Get Multiple Block Security Status response format when Error_flag is NOT set | 94 |
| Table 71. | Sector security status | 94 |
| Table 72. | Get Multiple Block Security Status response format when Error_flag is set. | 95 |
| Table 73. | Write-sector Password request format | 96 |
| Table 74. | Write-sector Password response format when Error_flag is NOT set | 96 |
| Table 75. | Write-sector Password response format when Error_flag is set. | 96 |
| Table 76. | Lock-sector Password request format | 98 |
| Table 77. | Sector security status | 98 |
| Table 78. | Lock-sector Password response format when Error_flag is NOT set | 98 |
| Table 79. | Lock-sector Password response format when Error_flag is set | 98 |
| Table 80. | Present-sector Password request format | 100 |
| Table 81. | Present-sector Password response format when Error_flag is NOT set | 100 |
| Table 82. | Present-sector Password response format when Error_flag is set. | 100 |
| Table 83. | Fast Read Single Block request format | 102 |
| Table 84. | Fast Read Single Block response format when Error_flag is NOT set. | 102 |
| Table 85. | Sector security status | 102 |
| Table 86. | Fast Read Single Block response format when Error_flag is set | 102 |
| Table 87. | Fast Inventory Initiated request format | 104 |
| Table 88. | Fast Inventory Initiated response format | 104 |
| Table 89. | Fast Initiate request format | 105 |
| Table 90. | Fast Initiate response format | 105 |
| Table 91. | Fast Read Multiple Block request format | 106 |
| Table 92. | Fast Read Multiple Block response format when Error_flag is NOT set. | 106 |
| Table 93. | Sector security status if Option_flag is set | 106 |
| Table 94. | Fast Read Multiple Block response format when Error_flag is set | 107 |
| Table 95. | Inventory Initiated request format | 108 |
| Table 96. | Inventory Initiated response format | 108 |
| Table 97. | Initiate request format. | 109 |
| Table 98. | Initiate Initiated response format. | 109 |
| Table 99. | Absolute maximum ratings | 110 |
| Table 100. | I ² C operating conditions | 111 |

| | | |
|------------|---|-----|
| Table 101. | AC test measurement conditions | 111 |
| Table 102. | Input parameters | 111 |
| Table 103. | I ² C DC characteristics | 112 |
| Table 104. | I ² C AC characteristics | 113 |
| Table 105. | RF AC characteristics | 115 |
| Table 106. | RF DC characteristics | 116 |
| Table 107. | Operating conditions | 116 |
| Table 108. | SO8N – 8-lead plastic small outline, 150 mils body width, package data | 117 |
| Table 109. | UFDFPN8 (MLP8) – Ultra thin fine pitch dual flat package no lead 2 x 3 mm, package mechanical data | 118 |
| Table 110. | TSSOP8 – 8-lead thin shrink small outline, package mechanical data | 119 |
| Table 111. | Ordering information scheme | 120 |
| Table 112. | CRC definition | 122 |
| Table 113. | AFI coding | 124 |
| Table 114. | Document revision history | 125 |

List of figures

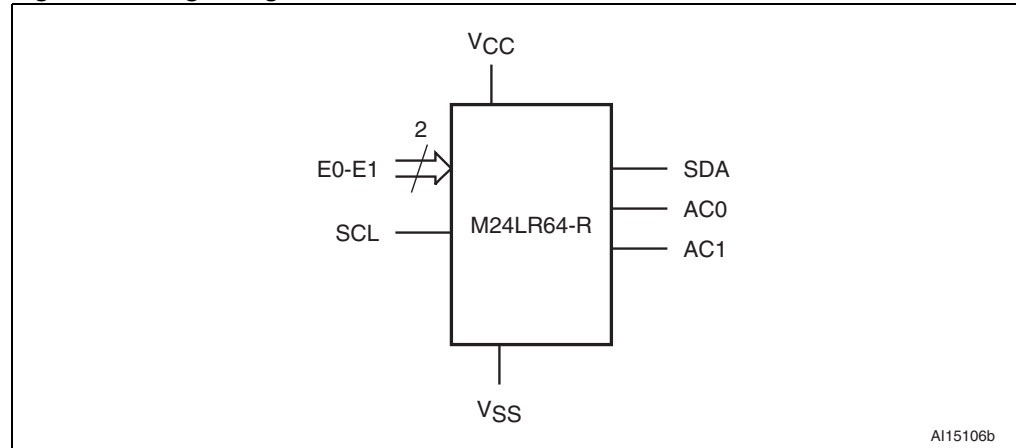
| | | |
|------------|--|----|
| Figure 1. | Logic diagram | 12 |
| Figure 2. | SO8 connections | 13 |
| Figure 3. | Device select code | 14 |
| Figure 4. | Maximum R_P value versus bus parasitic capacitance (C) for an I ² C bus | 15 |
| Figure 5. | I ² C bus protocol | 16 |
| Figure 6. | Block diagram | 17 |
| Figure 7. | Memory sector organization | 18 |
| Figure 8. | I ² C Present Password command | 27 |
| Figure 9. | I ² C Write Password command | 28 |
| Figure 10. | Write mode sequences with I2C_Write_Lock bit = 1 (data write inhibited) | 30 |
| Figure 11. | Write mode sequences with I2C_Write_Lock bit = 0 (data write enabled) | 32 |
| Figure 12. | Write cycle polling flowchart using ACK | 32 |
| Figure 13. | Read mode sequences | 34 |
| Figure 14. | 100% modulation waveform | 39 |
| Figure 15. | 10% modulation waveform | 40 |
| Figure 16. | 1 out of 256 coding mode | 41 |
| Figure 17. | Detail of a time period | 42 |
| Figure 18. | 1 out of 4 coding mode | 43 |
| Figure 19. | 1 out of 4 coding example | 43 |
| Figure 20. | SOF to select 1 out of 256 data coding mode | 44 |
| Figure 21. | SOF to select 1 out of 4 data coding mode | 44 |
| Figure 22. | EOF for either data coding mode | 45 |
| Figure 23. | Logic 0, high data rate | 47 |
| Figure 24. | Logic 0, high data rate x2 | 47 |
| Figure 25. | Logic 1, high data rate | 47 |
| Figure 26. | Logic 1, high data rate x2 | 47 |
| Figure 27. | Logic 0, low data rate | 48 |
| Figure 28. | Logic 0, low data rate x2 | 48 |
| Figure 29. | Logic 1, low data rate | 48 |
| Figure 30. | Logic 1, low data rate x2 | 48 |
| Figure 31. | Logic 0, high data rate | 49 |
| Figure 32. | Logic 1, high data rate | 49 |
| Figure 33. | Logic 0, low data rate | 49 |
| Figure 34. | Logic 1, low data rate | 49 |
| Figure 35. | Start of frame, high data rate, one subcarrier | 50 |
| Figure 36. | Start of frame, high data rate, one subcarrier x2 | 50 |
| Figure 37. | Start of frame, low data rate, one subcarrier | 50 |
| Figure 38. | Start of frame, low data rate, one subcarrier x2 | 51 |
| Figure 39. | Start of frame, high data rate, two subcarriers | 51 |
| Figure 40. | Start of frame, low data rate, two subcarriers | 51 |
| Figure 41. | End of frame, high data rate, one subcarriers | 52 |
| Figure 42. | End of frame, high data rate, one subcarriers x2 | 52 |
| Figure 43. | End of frame, low data rate, one subcarriers | 52 |
| Figure 44. | End of frame, low data rate, one subcarriers x2 | 52 |
| Figure 45. | End of frame, high data rate, two subcarriers | 53 |
| Figure 46. | End of frame, low data rate, two subcarriers | 53 |
| Figure 47. | M24LR64-R decision tree for AFI | 55 |
| Figure 48. | M24LR64-R protocol timing | 58 |

| | | |
|------------|---|-----|
| Figure 49. | M24LR64-R state transition diagram | 60 |
| Figure 50. | Principle of comparison between the mask, the slot number and the UID | 67 |
| Figure 51. | Description of a possible anticollision sequence | 70 |
| Figure 52. | Stay Quiet frame exchange between VCD and M24LR64-R | 75 |
| Figure 53. | Read Single Block frame exchange between VCD and M24LR64-R | 77 |
| Figure 54. | Write Single Block frame exchange between VCD and M24LR64-R | 79 |
| Figure 55. | Read Multiple Block frame exchange between VCD and M24LR64-R | 81 |
| Figure 56. | Select frame exchange between VCD and M24LR64-R | 82 |
| Figure 57. | Reset to Ready frame exchange between VCD and M24LR64-R | 83 |
| Figure 58. | Write AFI frame exchange between VCD and M24LR64-R | 85 |
| Figure 59. | Lock AFI frame exchange between VCD and M24LR64-R | 87 |
| Figure 60. | Write DSFID frame exchange between VCD and M24LR64-R | 89 |
| Figure 61. | Lock DSFID frame exchange between VCD and M24LR64-R | 91 |
| Figure 62. | Get System Info frame exchange between VCD and M24LR64-R | 93 |
| Figure 63. | Get Multiple Block Security Status frame exchange between VCD and M24LR64-R | 95 |
| Figure 64. | Write-sector Password frame exchange between VCD and M24LR64-R | 97 |
| Figure 65. | Lock-sector Password frame exchange between VCD and M24LR64-R | 99 |
| Figure 66. | Present-sector Password frame exchange between VCD and M24LR64-R | 101 |
| Figure 67. | Fast Read Single Block frame exchange between VCD and M24LR64-R | 103 |
| Figure 68. | Fast Initiate frame exchange between VCD and M24LR64-R | 105 |
| Figure 69. | Fast Read Multiple Block frame exchange between VCD and M24LR64-R | 107 |
| Figure 70. | Initiate frame exchange between VCD and M24LR64-R | 109 |
| Figure 71. | AC test measurement I/O waveform | 111 |
| Figure 72. | I ² C AC waveforms | 114 |
| Figure 73. | M24LR64-R synchronous timing, transmit and receive | 116 |
| Figure 74. | SO8N – 8-lead plastic small outline, 150 mils body width, package outline | 117 |
| Figure 75. | UFDFPN8 (MLP8) – Ultra thin fine pitch dual flat package no lead 2 x 3 mm, package outline | 118 |
| Figure 76. | TSSOP8 – 8-lead thin shrink small outline, package outline | 119 |

1 Description

The M24LR64-R device is a dual-interface, electrically erasable programmable memory (EEPROM). It features an I²C interface and can be operated from a V_{CC} power supply. It is also a contactless memory powered by the received carrier electromagnetic wave. The M24LR64-R is organized as 8192 × 8 bits in the I²C mode and as 2048 × 32 bits in the ISO 15693 and ISO 18000-3 mode 1 RF mode.

Figure 1. Logic diagram



I²C uses a two-wire serial interface, comprising a bidirectional data line and a clock line. The devices carry a built-in 4-bit device type identifier code (1010) in accordance with the I²C bus definition.

The device behaves as a slave in the I²C protocol, with all memory operations synchronized by the serial clock. Read and Write operations are initiated by a Start condition, generated by the bus master. The Start condition is followed by a device select code and Read/Write bit (RW) (as described in [Table 2](#)), terminated by an acknowledge bit.

When writing data to the memory, the device inserts an acknowledge bit during the 9th bit time, following the bus master's 8-bit transmission. When data is read by the bus master, the bus master acknowledges the receipt of the data byte in the same way. Data transfers are terminated by a Stop condition after an Ack for Write, and after a NoAck for Read.

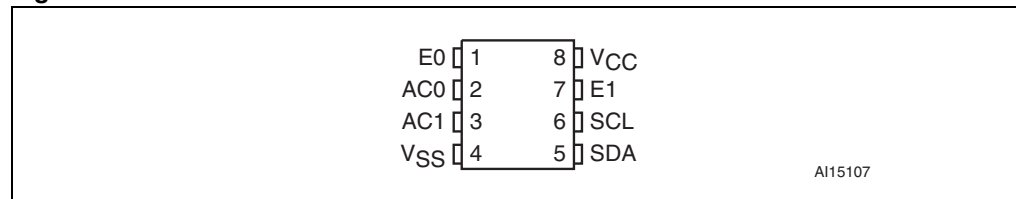
In the ISO15693/ISO18000-3 mode 1 RF mode, the M24LR64-R is accessed via the 13.56 MHz carrier electromagnetic wave on which incoming data are demodulated from the received signal amplitude modulation (ASK: amplitude shift keying). The received ASK wave is 10% or 100% modulated with a data rate of 1.6 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 M24LR64-R load variation using Manchester coding with one or two subcarrier frequencies at 423 kHz and 484 kHz. Data are transferred from the M24LR64-R at 6.6 Kbit/s in low data rate mode and 26 Kbit/s high data rate mode. The M24LR64-R supports the 53 Kbit/s in high data rate mode in one subcarrier frequency at 423 kHz.

The M24LR64-R follows the ISO 15693 and ISO 18000-3 mode 1 recommendation for radio-frequency power and signal interface.

Table 1. Signal names

| Signal name | Function | Direction |
|-----------------|----------------|-----------|
| E0, E1 | Chip Enable | Input |
| SDA | Serial Data | I/O |
| SCL | Serial Clock | Input |
| AC0, AC1 | Antenna coils | I/O |
| V _{CC} | Supply voltage | |
| V _{SS} | Ground | |

Figure 2. SO8 connections

1. See [Package mechanical data](#) section for package dimensions, and how to identify pin-1.

2 Signal description

2.1 Serial Clock (SCL)

This input signal is used to strobe all data in and out of the device. 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} . (*Figure 4* 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.

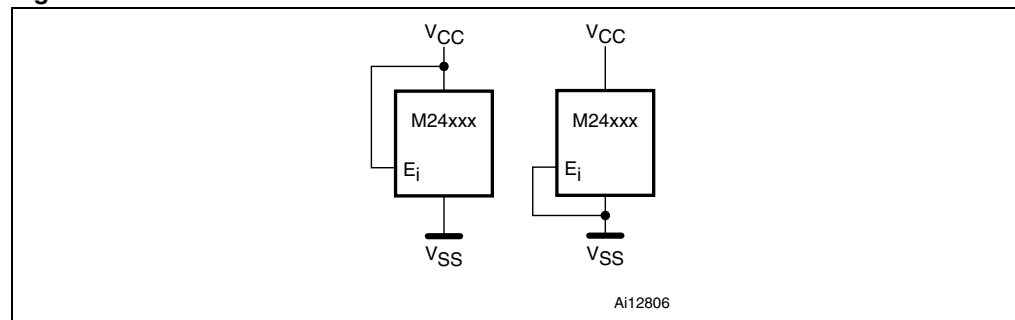
2.2 Serial Data (SDA)

This bidirectional signal is used to transfer data in or out of the device. 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 4* indicates how the value of the pull-up resistor can be calculated).

2.3 Chip Enable (E0, E1)

These input signals are used to set the value that is to be looked for on the two least significant bits (b2, b1) of the 7-bit device select code. These inputs must be tied to V_{CC} or V_{SS} , to establish the device select code as shown in *Figure 3*. When not connected (left floating), these inputs are read as low (0,0).

Figure 3. Device select code



2.4 Antenna coil (AC0, AC1)

These inputs are used to connect the device to an external coil. When correctly tuned, the coil is used to power and access the device using the ISO 15693 and ISO 18000-3 mode 1 protocols.

2.5 V_{SS} ground

V_{SS} is the reference for the V_{CC} supply voltage.

2.6 Supply voltage (V_{CC})

2.6.1 Operating supply voltage V_{CC}

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 100](#)). In order to secure 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 to 100 nF) 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).

2.6.2 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/ μ s.

2.6.3 Device reset

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 device does not respond to any 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 100](#)). 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.

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.

2.6.4 Power-down conditions

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).

Figure 4. Maximum R_P value versus bus parasitic capacitance (C) for an I²C bus

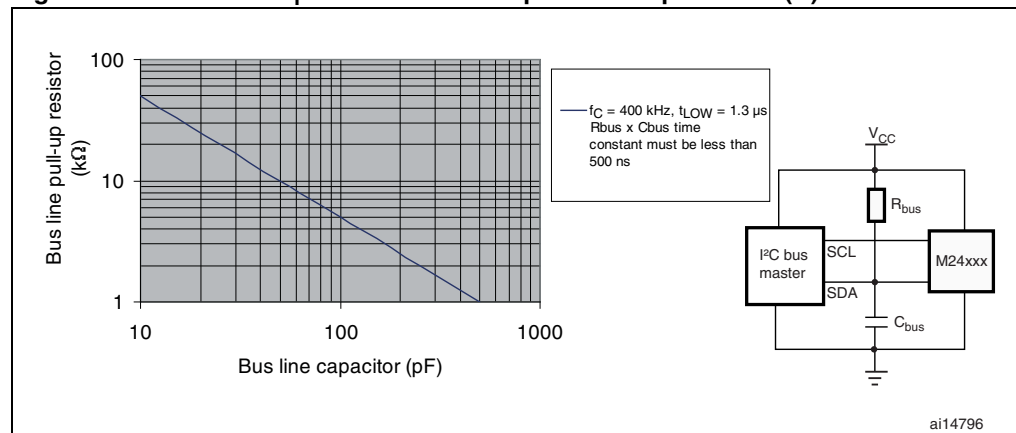


Figure 5. I²C bus protocol

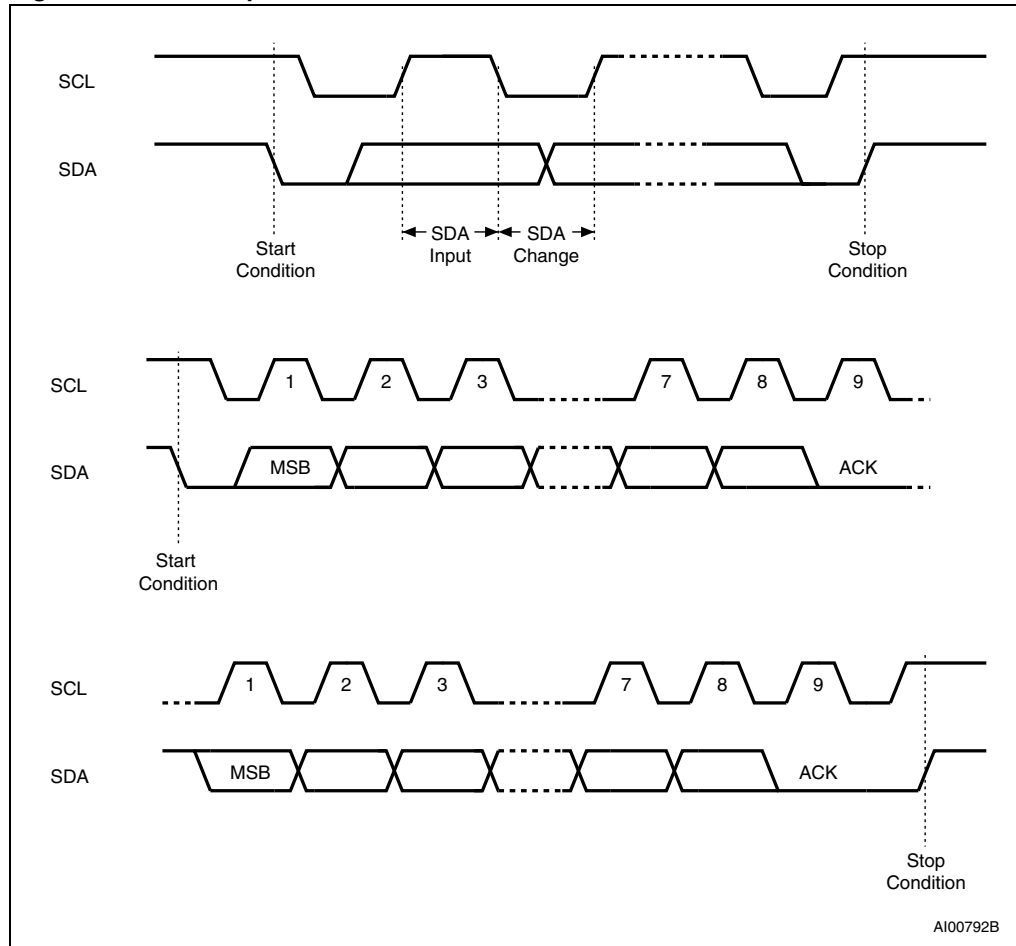


Table 2. Device select code

| | Device type identifier ⁽¹⁾ | | | | Chip Enable address ⁽²⁾ | | | R \bar{W} |
|--------------------|---------------------------------------|----|----|----|------------------------------------|----|----|-------------|
| | b7 | b6 | b5 | b4 | b3 | b2 | b1 | b0 |
| Device select code | 1 | 0 | 1 | 0 | E2 ⁽³⁾ | E1 | E0 | R \bar{W} |

1. The most significant bit, b7, is sent first.
2. E0 and E1 are compared against the respective external pins on the memory device.
3. E2 is not connected to any external pin. It is however used to address the M24LR64-R as described in [Section 3](#) and [Section 4](#).

Table 3. Address most significant byte

| | | | | | | | |
|-----|-----|-----|-----|-----|-----|----|----|
| b15 | b14 | b13 | b12 | b11 | b10 | b9 | b8 |
|-----|-----|-----|-----|-----|-----|----|----|

Table 4. Address least significant byte

| | | | | | | | |
|----|----|----|----|----|----|----|----|
| b7 | b6 | b5 | b4 | b3 | b2 | b1 | b0 |
|----|----|----|----|----|----|----|----|

3 User memory organization

The M24LR64-R is divided into 64 sectors of 32 blocks of 32 bits as shown in [Table 5](#). [Figure 7](#) shows the memory sector organization. Each sector can be individually read-and/or write-protected using a specific password command. Read and write operations are possible if the addressed data are not in a protected sector.

The M24LR64-R also has a 64-bit block that is used to store the 64-bit unique identifier (UID). The UID is compliant with the ISO 15963 description, and its value is used during the anticollision sequence (Inventory). This block is not accessible by the user and its value is written by ST on the production line.

The M24LR64-R includes an AFI register that stores the application family identifier, and a DSFID register that stores the data storage family identifier used in the anticollision algorithm.

The M24LR64-R has four additional 32-bit blocks that store an I²C password plus three RF password codes.

Figure 6. Block diagram

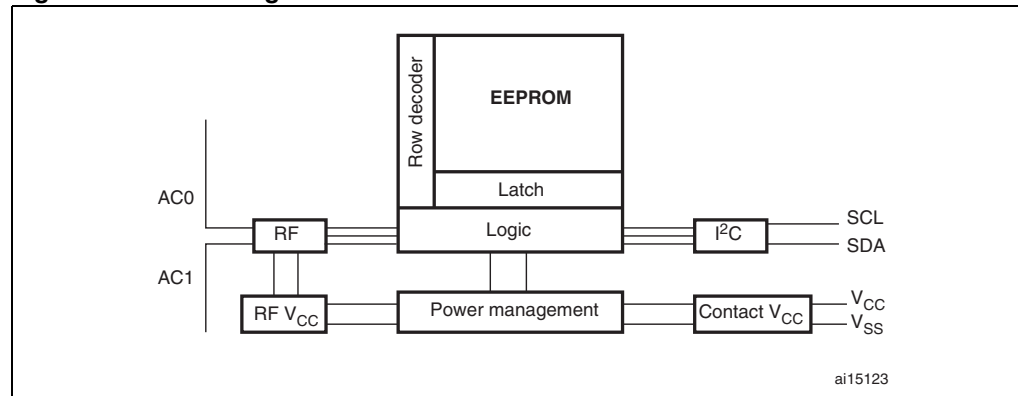


Figure 7. Memory sector organization

| Sector | Area | Sector security status |
|--------|----------------------|------------------------|
| 0 | 1 Kbit EEPROM sector | 5 bits |
| 1 | 1 Kbit EEPROM sector | 5 bits |
| 2 | 1 Kbit EEPROM sector | 5 bits |
| 3 | 1 Kbit EEPROM sector | 5 bits |
| 60 | 1 Kbit EEPROM sector | 5 bits |
| 61 | 1 Kbit EEPROM sector | 5 bits |
| 62 | 1 Kbit EEPROM sector | 5 bits |
| 63 | 1 Kbit EEPROM sector | 5 bits |
| | I2C Password | System |
| | RF Password 1 | System |
| | RF Password 2 | System |
| | RF Password 3 | System |
| | 8 bit DSFID | System |
| | 8 bit AFI | System |
| | 64 bit UID | System |

ai15124

Sector details

The M24LR64-R user memory is divided into 64 sectors. Each sector contains 1024 bits. The protection scheme is described in [Section 4: System memory area](#).

In RF mode, a sector provides 32 blocks of 32 bits. Each read and write access are done by block. Read and write block accesses are controlled by a Sector Security Status byte that defines the access rights to all the 32 blocks contained in the sector. If the sector is not protected, a Write command updates the complete 32 bits of the selected block.

In I²C mode, a sector provides 128 bytes that can be individually accessed in read and write modes. When protected by the corresponding I2C_Write_Lock bit, the entire sector is write-protected. To access the user memory, the device select code used for any I²C command must have the E2 Chip Enable address at 0.

Table 5. Sector details

| Sector number | RF block address | I ² C byte address | Bits [31:24] | Bits [23:16] | Bits [15:8] | Bits [7:0] |
|---------------|------------------|-------------------------------|--------------|--------------|-------------|------------|
| 0 | 0 | 0 | user | user | user | user |
| | 1 | 4 | user | user | user | user |
| | 2 | 8 | user | user | user | user |
| | 3 | 12 | user | user | user | user |
| | 4 | 16 | user | user | user | user |
| | 5 | 20 | user | user | user | user |
| | 6 | 24 | user | user | user | user |
| | 7 | 28 | user | user | user | user |
| | 8 | 32 | user | user | user | user |
| | 9 | 36 | user | user | user | user |
| | 10 | 40 | user | user | user | user |
| | 11 | 44 | user | user | user | user |
| | 12 | 48 | user | user | user | user |
| | 13 | 52 | user | user | user | user |
| | 14 | 56 | user | user | user | user |
| | 15 | 60 | user | user | user | user |
| | 16 | 64 | user | user | user | user |
| | 17 | 68 | user | user | user | user |
| | 18 | 72 | user | user | user | user |
| | 19 | 76 | user | user | user | user |
| | 20 | 80 | user | user | user | user |
| | 21 | 84 | user | user | user | user |
| | 22 | 88 | user | user | user | user |
| | 23 | 92 | user | user | user | user |
| | 24 | 96 | user | user | user | user |
| | 25 | 100 | user | user | user | user |
| | 26 | 104 | user | user | user | user |
| | 27 | 108 | user | user | user | user |
| | 28 | 112 | user | user | user | user |
| | 29 | 116 | user | user | user | user |
| | 30 | 120 | user | user | user | user |
| | 31 | 124 | user | user | user | user |

Table 5. Sector details (continued)

| Sector number | RF block address | I ² C byte address | Bits [31:24] | Bits [23:16] | Bits [15:8] | Bits [7:0] |
|---------------|------------------|-------------------------------|--------------|--------------|-------------|------------|
| 1 | 32 | 128 | user | user | user | user |
| | 33 | 132 | user | user | user | user |
| | 34 | 136 | user | user | user | user |
| | 35 | 140 | user | user | user | user |
| | 36 | 144 | user | user | user | user |
| | 37 | 148 | user | user | user | user |
| | 38 | 152 | user | user | user | user |
| | 39 | 156 | user | user | user | user |
| | ... | ... | ... | ... | ... | ... |
| ... | ... | ... | ... | ... | ... | |

Table 5. Sector details (continued)

| Sector number | RF block address | I ² C byte address | Bits [31:24] | Bits [23:16] | Bits [15:8] | Bits [7:0] |
|---------------|------------------|-------------------------------|--------------|--------------|-------------|------------|
| 63 | 2016 | 8064 | user | user | user | user |
| | 2017 | 8068 | user | user | user | user |
| | 2018 | 8072 | user | user | user | user |
| | 2019 | 8076 | user | user | user | user |
| | 2020 | 8080 | user | user | user | user |
| | 2021 | 8084 | user | user | user | user |
| | 2022 | 8088 | user | user | user | user |
| | 2023 | 8092 | user | user | user | user |
| | 2024 | 8096 | user | user | user | user |
| | 2025 | 8100 | user | user | user | user |
| | 2026 | 8104 | user | user | user | user |
| | 2027 | 8108 | user | user | user | user |
| | 2028 | 8112 | user | user | user | user |
| | 2029 | 8116 | user | user | user | user |
| | 2030 | 8120 | user | user | user | user |
| | 2031 | 8124 | user | user | user | user |
| | 2032 | 8128 | user | user | user | user |
| | 2033 | 8132 | user | user | user | user |
| | 2034 | 8136 | user | user | user | user |
| | 2035 | 8140 | user | user | user | user |
| | 2036 | 8144 | user | user | user | user |
| | 2037 | 8148 | user | user | user | user |
| | 2038 | 8152 | user | user | user | user |
| | 2039 | 8156 | user | user | user | user |
| | 2040 | 8160 | user | user | user | user |
| | 2041 | 8164 | user | user | user | user |
| | 2042 | 8168 | user | user | user | user |
| | 2043 | 8172 | user | user | user | user |
| | 2044 | 8176 | user | user | user | user |
| | 2045 | 8180 | user | user | user | user |
| 2046 | 8184 | user | user | user | user | |
| 2047 | 8188 | user | user | user | user | |

4 System memory area

4.1 M24LR64-R RF block security

The M24LR64-R provides a special protection mechanism based on passwords. Each memory sector of the M24LR64-R can be individually protected by one out of three available passwords, and each sector can also have Read/Write access conditions set.

Each memory sector of the M24LR64-R is assigned with a Sector security status byte including a Sector Lock bit, two Password Control bits and two Read/Write protection bits as shown in [Table 7](#). [Table 6](#) describes the organization of the Sector security status byte which can be read using the Read Single Block and Read Multiple Block commands with the Option_flag set to '1'.

On delivery, the default value of the SSS bytes is reset to 00h.

Table 6. Sector Security Status Byte area

| RF address | I ² C byte address | | Bits [31:24] | Bits [23:16] | Bits [15:8] | Bits [7:0] |
|------------|-------------------------------|----|--------------|--------------|-------------|------------|
| 0 | E2 = 1 | 0 | SSS 3 | SSS 2 | SSS 1 | SSS 0 |
| 128 | E2 = 1 | 4 | SSS 7 | SSS 6 | SSS 5 | SSS 4 |
| 256 | E2 = 1 | 8 | SSS 11 | SSS 10 | SSS 9 | SSS 8 |
| 384 | E2 = 1 | 12 | SSS 15 | SSS 14 | SSS 13 | SSS 12 |
| 512 | E2 = 1 | 16 | SSS 19 | SSS 18 | SSS 17 | SSS 16 |
| 640 | E2 = 1 | 20 | SSS 23 | SSS 22 | SSS 21 | SSS 20 |
| 768 | E2 = 1 | 24 | SSS 27 | SSS 26 | SSS 25 | SSS 24 |
| 896 | E2 = 1 | 28 | SSS 31 | SSS 30 | SSS 29 | SSS 28 |
| 1024 | E2 = 1 | 32 | SSS 35 | SSS 34 | SSS 33 | SSS 32 |
| 1152 | E2 = 1 | 36 | SSS 39 | SSS 38 | SSS 37 | SSS 36 |
| 1280 | E2 = 1 | 40 | SSS 43 | SSS 42 | SSS 41 | SSS 40 |
| 1408 | E2 = 1 | 44 | SSS 47 | SSS 46 | SSS 45 | SSS 44 |
| 1536 | E2 = 1 | 48 | SSS 51 | SSS 50 | SSS 49 | SSS 48 |
| 1664 | E2 = 1 | 52 | SSS 55 | SSS 54 | SSS 53 | SSS 52 |
| 1792 | E2 = 1 | 56 | SSS 59 | SSS 58 | SSS 57 | SSS 56 |
| 1920 | E2 = 1 | 60 | SSS 63 | SSS 62 | SSS 61 | SSS 60 |

Table 7. Sector security status byte organization

| b ₇ | b ₆ | b ₅ | b ₄ | b ₃ | b ₂ | b ₁ | b ₀ |
|----------------|----------------|----------------|-----------------------|----------------|------------------------------|----------------|----------------|
| 0 | 0 | 0 | Password Control bits | | Read / Write protection bits | | Sector Lock |

When the Sector Lock bit is set to '1', for instance by issuing a Lock-sector Password command, the 2 Read/Write protection bits (b₁, b₂) are used to set the Read/Write access of the sector as described in [Table 8](#).

Table 8. Read / Write protection bit setting

| Sector Lock | b ₂ , b ₁ | Sector access when password presented | | Sector access when password not presented | |
|-------------|---------------------------------|---------------------------------------|----------|---|----------|
| | | Read | Write | Read | Write |
| 0 | xx | Read | Write | Read | Write |
| 1 | 00 | Read | Write | Read | No Write |
| 1 | 01 | Read | Write | Read | Write |
| 1 | 10 | Read | Write | No Read | No Write |
| 1 | 11 | Read | No Write | No Read | No Write |

The next 2 bits of the Sector security status byte (b₃, b₄) are the Password Control bits. The value these two bits is used to link a password to the sector as defined in [Table 9](#).

Table 9. Password Control bits

| b ₄ , b ₃ | Password |
|---------------------------------|---|
| 00 | The sector is not protected by a Password |
| 01 | The sector is protected by the Password 1 |
| 10 | The sector is protected by the Password 2 |
| 11 | The sector is protected by the Password 3 |

The M24LR64-R password protection is organized around a dedicated set of commands plus a system area of three password blocks where the password values are stored. This system area is described in [Table 10](#).

Table 10. Password system area

| Add | 0 | 7 | 8 | 15 | 16 | 23 | 24 | 31 |
|-----|------------|---|---|----|----|----|----|----|
| 1 | Password 1 | | | | | | | |
| 2 | Password 2 | | | | | | | |
| 3 | Password 3 | | | | | | | |

The dedicated password commands are:

- **Write-sector Password:**

The Write-sector Password command is used to write a 32-bit block into the password system area. This command must be used to update password values. After the write cycle, the new password value is automatically activated. It is possible to modify a password value after issuing a valid Present-sector Password command. On delivery, the three default password values are set to 0000 0000h and are activated.

- **Lock-sector Password:**

The Lock-sector Password command is used to set the Sector security status byte of the selected sector. Bits b₄ to b₁ of the Sector security status byte are affected by the Lock-sector Password command. The Sector Lock bit, b₀, is set to '1' automatically. After issuing a Lock-sector Password command, the protection settings of the selected sector are activated. The protection of a locked block cannot be changed in RF mode. A Lock-sector Password command sent to a locked sector returns an error code.

- **Present-sector Password:**
The Present-sector Password command is used to present one of the three passwords to the M24LR64-R in order to modify the access rights of all the memory sectors linked to that password ([Table 8](#)) including the password itself. If the presented password is correct, the access rights remain activated until the tag is powered off or until a new Present-sector Password command is issued. If the presented password value is not correct, all the access rights of all the memory sectors are deactivated.
- **Sector security status byte area access conditions in I²C mode:**
In I²C mode, read access to the Sector security status byte area is always allowed. Write access depends on the correct presentation of the I²C password (see [I²C Present Password command description on page 26](#)).
To access the Sector security status byte area, the device select code used for any I²C command must have the E2 Chip Enable address at 1.
An I²C write access to a Sector security status byte re-initializes the RF access condition to the given memory sector.

4.2 Example of the M24LR64-R security protection

[Table 11](#) and [Table 12](#) show the sector security protections before and after a valid Present-sector Password command. [Table 11](#) shows the sector access rights of an M24LR64-R after power-up. After a valid Present-sector Password command with password 1, the memory sector access is changed as shown in [Table 12](#).

Table 11. M24LR64-R sector security protection after power-up

| Sector address | | | | Sector security status byte | | | | | |
|----------------|----------------------|---------|----------|--|----------------|----------------|----------------|----------------|----------------|
| | | | | b ₇ b ₆ b ₅ | b ₄ | b ₃ | b ₂ | b ₁ | b ₀ |
| 0 | Protection: Standard | Read | No Write | xxx | 0 | 0 | 0 | 0 | 1 |
| 1 | Protection: Pswd 1 | Read | No Write | xxx | 0 | 1 | 0 | 0 | 1 |
| 2 | Protection: Pswd 1 | Read | Write | xxx | 0 | 1 | 0 | 1 | 1 |
| 3 | Protection: Pswd 1 | No Read | No Write | xxx | 0 | 1 | 1 | 0 | 1 |
| 4 | Protection: Pswd 1 | No Read | No Write | xxx | 0 | 1 | 1 | 1 | 1 |

Table 12. M24LR64-R sector security protection after a valid presentation of password 1

| Sector address | | | | Sector security status byte | | | | | |
|----------------|----------------------|------|----------|--|----------------|----------------|----------------|----------------|----------------|
| | | | | b ₇ b ₆ b ₅ | b ₄ | b ₃ | b ₂ | b ₁ | b ₀ |
| 0 | Protection: Standard | Read | No Write | xxx | 0 | 0 | 0 | 0 | 1 |
| 1 | Protection: Pswd 1 | Read | Write | xxx | 0 | 1 | 0 | 0 | 1 |
| 2 | Protection: Pswd 1 | Read | Write | xxx | 0 | 1 | 0 | 1 | 1 |
| 3 | Protection: Pswd 1 | Read | Write | xxx | 0 | 1 | 1 | 0 | 1 |
| 4 | Protection: Pswd 1 | Read | No Write | xxx | 0 | 1 | 1 | 1 | 1 |

4.3 I2C_Write_Lock bit area

In the I²C mode only, it is possible to protect individual sectors against Write operations. This feature is controlled by the I2C_Write_Lock bits stored in the 8 bytes of the I2C_Write_Lock bit area starting from the location 2048 (see [Table 13](#)). Using these 64 bits, it is possible to write-protect all the 64 sectors of the M24LR64-R memory.

Each bit controls the I²C write access to a specific sector as shown in [Table 13](#). It is always possible to unprotect a sector in the I²C mode. When an I2C_Write_Lock bit is reset to 0, the corresponding sector is unprotected. When the bit is set to 1, the corresponding sector is write-protected.

In I²C mode, read access to the I2C_Write_Lock bit area is always allowed. Write access depends on the correct presentation of the I²C password.

To access the I2C_Write_Lock bit area, the device select code used for any I²C command must have the E2 Chip Enable address at 1.

On delivery, the default value of the 8 bytes of the I2C_Write_Lock bit area is reset to 00h.

Table 13. I2C_Write_Lock bit

| I ² C byte address | | Bits [31:24] | Bits [23:16] | Bits [15:8] | Bits [7:0] |
|-------------------------------|------|---------------|---------------|---------------|---------------|
| E2 = 1 | 2048 | sectors 31-24 | sectors 23-16 | sectors 15-8 | sectors 7-0 |
| E2 = 1 | 2052 | sectors 63-56 | sectors 55-48 | sectors 47-40 | sectors 39-32 |

4.4 System parameters

The M24LR64-R provides the system area required by the ISO 15693 RF protocol, as shown in [Table 14](#).

The first 32-bit block starting from I²C address 2304 stores the I²C password. This password is used to activate/deactivate the write protection of the protected sector in I²C mode. At power-on, all user memory sectors protected by the I2C_Write_Lock bits can be read but cannot be modified. To remove the write protection, it is necessary to use the I²C Present Password described in [Figure 8](#). When the password is correctly presented — that is, when all the presented bits correspond to the stored ones — it is also possible to modify the I²C password using the I²C Write Password command described in [Figure 9](#).

The next three 32-bit blocks store the three RF passwords. These passwords are neither read- nor write- accessible in the I²C mode.

The next 2 bytes are used to store the AFI, at I²C location 2322, and the DSFID, at I²C location 2323. These 2 values are used during the RF Inventory sequence. They are read-only in the I²C mode.

The next 8 bytes, starting from location 2324, store the 64-bit UID programmed by ST on the production line. Bytes at I²C locations 2332 to 2335 store the IC Ref and the Mem_Size data used by the RF Get_System_Info command. The UID, Mem_Size and IC Ref values are read-only data.

Table 14. System parameter sector

| RF address | I ² C byte address | | Bits [31:24] | Bits [23:16] | Bits [15:8] | Bits [7:0] |
|------------|-------------------------------|------|--|--------------|-------------|--------------|
| - | E2 = 1 | 2304 | I ² C password (default 0000 0000h) | | | |
| 1 | E2 = 1 | 2308 | RF password 1 (default 0000 0000h) | | | |
| 2 | E2 = 1 | 2312 | RF password 2 (default 0000 0000h) | | | |
| 3 | E2 = 1 | 2316 | RF password 3 (default 0000 0000h) | | | |
| - | E2 = 1 | 2320 | DSFID (FFh) | AFI (00h) | RFU (FFh) | RFU (FFh) |
| - | E2 = 1 | 2324 | UID | UID | UID | UID |
| - | E2 = 1 | 2328 | UID (E0h) | UID (02h) | UID | UID |
| - | E2 = 1 | 2332 | Mem_Size (03 07FFh) | | | IC Ref (2Ch) |

4.5 M24LR64-R I²C password security

The M24LR64-R controls I²C sector write access using the 32-bit-long I²C password and the 64-bit I2C_Write_Lock bit area. The I²C password value is managed using two I²C commands: I²C Present Password and I²C Write Password.

4.5.1 I²C Present Password command description

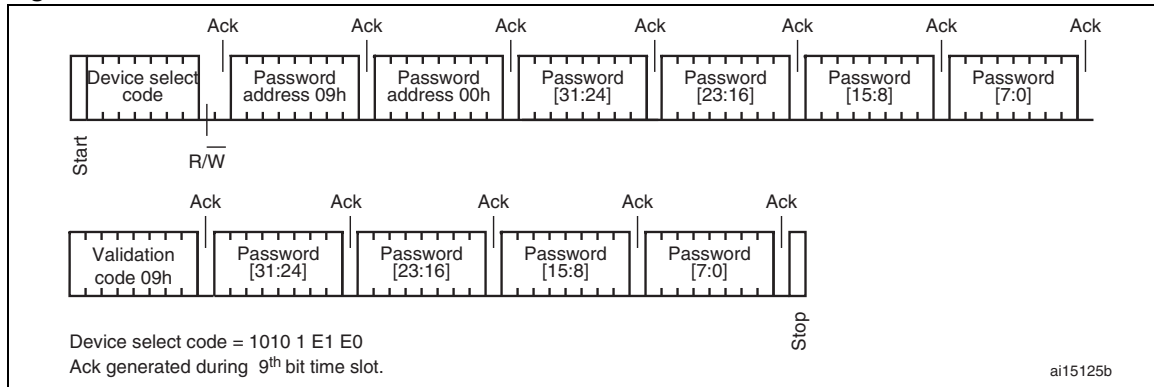
The I²C Present Password command is used in I²C mode to present the password to the M24LR64-R in order to modify the write access rights of all the memory sectors protected by the I2C_Write_Lock bits, including the password itself. If the presented password is correct, the access rights remain activated until the M24LR64-R is powered off or until a new I²C Present Password command is issued.

Following a Start condition, the bus master sends a device select code with the Read/Write bit (RW) reset to 0 and the Chip Enable bit E2 at 1. The device acknowledges this, as shown in [Figure 8](#), 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 4 password data bytes, the validation code, 09h, and a resend of the 4 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 32-bit password twice to prevent any data corruption during the sequence. If the two 32-bit passwords sent are not exactly the same, the M24LR64-R does not start the internal comparison.

When the bus master generates a Stop condition immediately after the Ack bit (during the “10th 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 M24LR64-R compares the 32 received data bits with the 32 bits of the stored I²C password. If the values match, the write access rights to all protected sectors are modified after the internal delay. If the values do not match, the protected sectors remains protected.

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

Figure 8. I²C Present Password command

4.5.2 I²C Write Password command description

The I²C Write Password command is used to write a 32-bit block into the M24LR64-R I²C password system area. This command is used in I²C mode to update the I²C password value. 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.

On delivery, the I²C default password value is set to 0000 0000h and is activated.

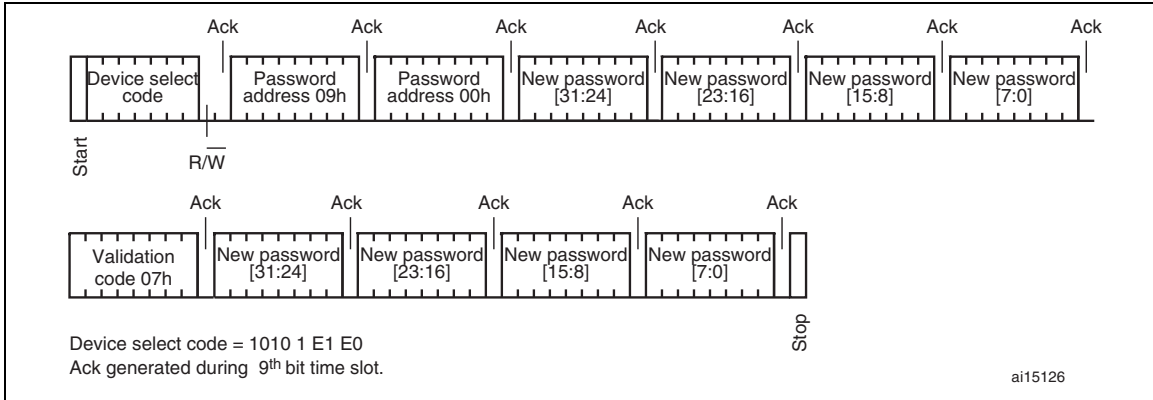
Following a Start condition, the bus master sends a device select code with the Read/Write bit ($\overline{R/W}$) reset to 0 and the Chip Enable bit E2 at 1. The device acknowledges this, as shown in [Figure 9](#), and waits for the two I²C password address bytes, 09h and 00h. The device responds to each address byte with an acknowledge bit, and then waits for the 4 password data bytes, the validation code, 07h, and a resend of the 4 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 32-bit password to prevent any data corruption during the write sequence. If the two 32-bit passwords sent are not exactly the same, the M24LR64-R does not modify the I²C password value.

When the bus master generates a Stop condition immediately after the Ack bit (during the 10th 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, Serial Data (SDA) is disabled internally, and the device does not respond to any requests.

Figure 9. I²C Write Password command



5 I²C device operation

The device supports the I²C protocol. This is summarized in [Figure 5](#). Any device that sends data on to the bus is defined to be a transmitter, and any device that reads the data to be 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 will also provide the serial clock for synchronization. The M24LR64-R device is always a slave in all communications.

5.1 Start condition

Start is identified by a falling edge of Serial Data (SDA) while 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) Serial Data (SDA) and Serial Clock (SCL) for a Start condition, and will not respond unless one is given.

5.2 Stop condition

Stop is identified by a rising edge of Serial Data (SDA) while 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.

5.3 Acknowledge bit (ACK)

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

5.4 Data Input

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

5.5 Memory addressing

To start 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 2](#) (on Serial Data (SDA), most significant bit first).

The device select code consists of a 4-bit device type identifier, and a 3-bit Chip Enable "Address" (E2, E1, E0). To address the memory array, the 4-bit device type identifier is 1010b.

Up to four memory devices can be connected on a single I²C bus. Each one is given a unique 2-bit code on the Chip Enable (E0, E1) inputs. When the device select code is received, the device only responds if the Chip Enable Address is the same as the value on the Chip Enable (E0, E1) inputs.

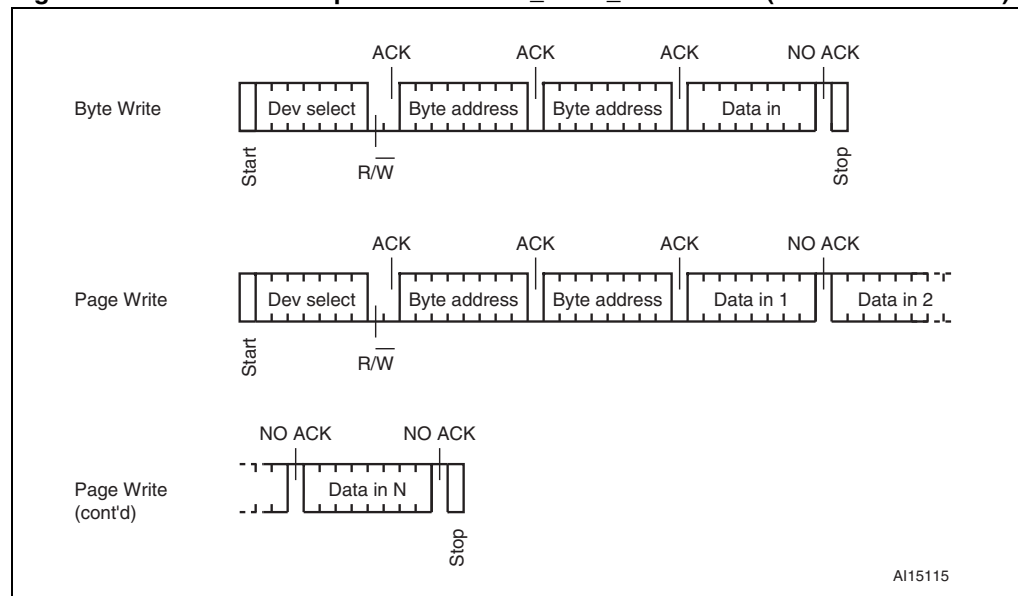
The 8th bit is the Read/Write bit (\overline{RW}). This bit is set to 1 for Read and 0 for Write operations.

If a match occurs on the device select code, the corresponding device gives an acknowledgment on Serial Data (SDA) during the 9th bit time. If the device does not match the device select code, it deselects itself from the bus, and goes into Standby mode.

Table 15. 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, $\overline{RW} = 0$ |
| Page Write | 0 | ≤ 4 bytes | Start, device select, $\overline{RW} = 0$ |

Figure 10. Write mode sequences with I2C_Write_Lock bit = 1 (data write inhibited)



5.6 Write operations

Following a Start condition the bus master sends a device select code with the Read/Write bit (\overline{RW}) reset to 0. The device acknowledges this, as shown in [Figure 11](#), and waits for two address bytes. The device responds to each address byte with an acknowledge bit, and then waits for the data byte.

Writing to the memory may be inhibited if the I2C_Write_Lock bit = 1. A Write instruction issued with the I2C_Write_Lock bit = 1 and with no I2C_Password presented, does not modify the memory contents, and the accompanying data bytes are *not* acknowledged, as shown in [Figure 10](#).

Each data byte in the memory has a 16-bit (two byte wide) address. The most significant byte ([Table 3](#)) is sent first, followed by the least significant byte ([Table 4](#)). Bits b15 to b0 form the address of the byte in memory.

When the bus master generates a Stop condition immediately after the Ack bit (in the “10th bit” time slot), either at the end of a Byte Write or a Page 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_{WW} , 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.

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

5.7 Byte Write

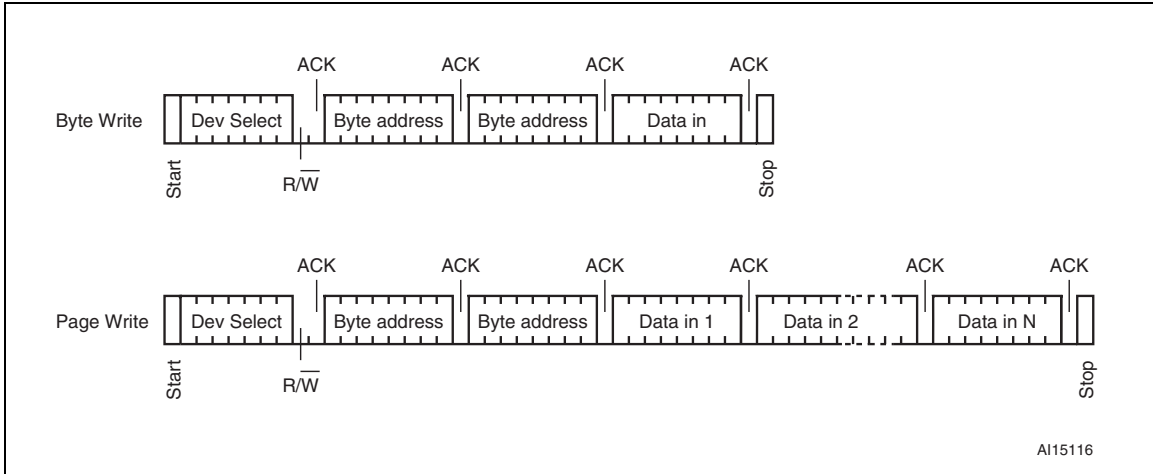
After the device select code and the address bytes, the bus master sends one data byte. If the addressed location is write-protected by the I2C_Write_Lock bit (= 1), the device replies with NoAck, and the location is not modified. If, instead, the addressed location is not Write-protected, the device replies with Ack. The bus master terminates the transfer by generating a Stop condition, as shown in [Figure 11](#).

5.8 Page Write

The Page Write mode allows up to 4 bytes to be written in a single Write cycle, provided that they are all located in the same “row” in the memory: that is, the most significant memory address bits (b12-b2) are the same. If more bytes are sent than will fit up to the end of the row, a condition known as ‘roll-over’ occurs. This should be avoided, as data starts to become overwritten in an implementation dependent way.

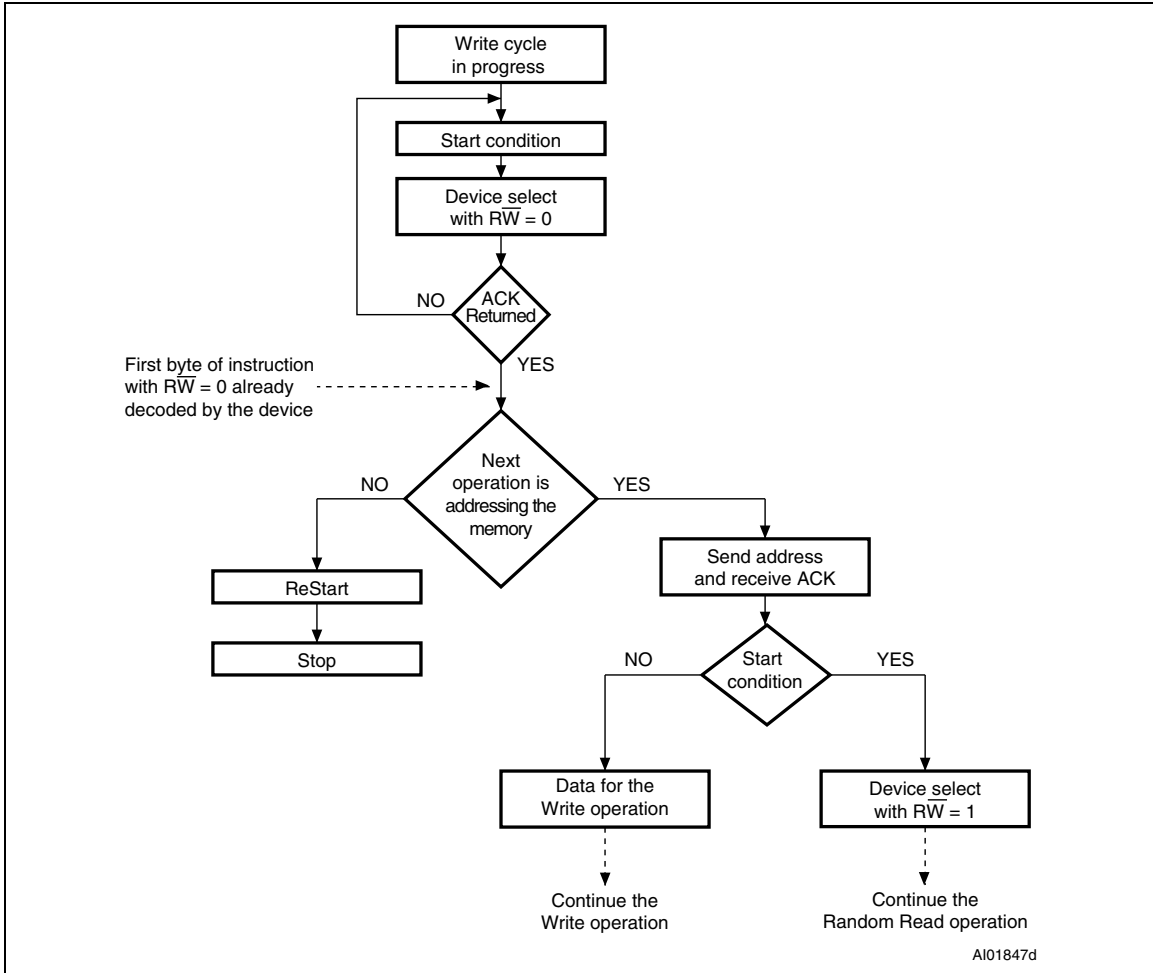
The bus master sends from 1 to 4 bytes of data, each of which is acknowledged by the device if the I2C_Write_Lock bit = 0 or the I2C_Password was correctly presented. If the I2C_Write_Lock_bit = 1 and the I2C_password is not presented, the contents of the addressed memory location are not modified, and each data byte is followed by a NoAck. After each byte is transferred, the internal byte address counter (inside the page) is incremented. The transfer is terminated by the bus master generating a Stop condition.

Figure 11. Write mode sequences with I2C_Write_Lock bit = 0 (data write enabled)



AI15116

Figure 12. Write cycle polling flowchart using ACK



AI01847d

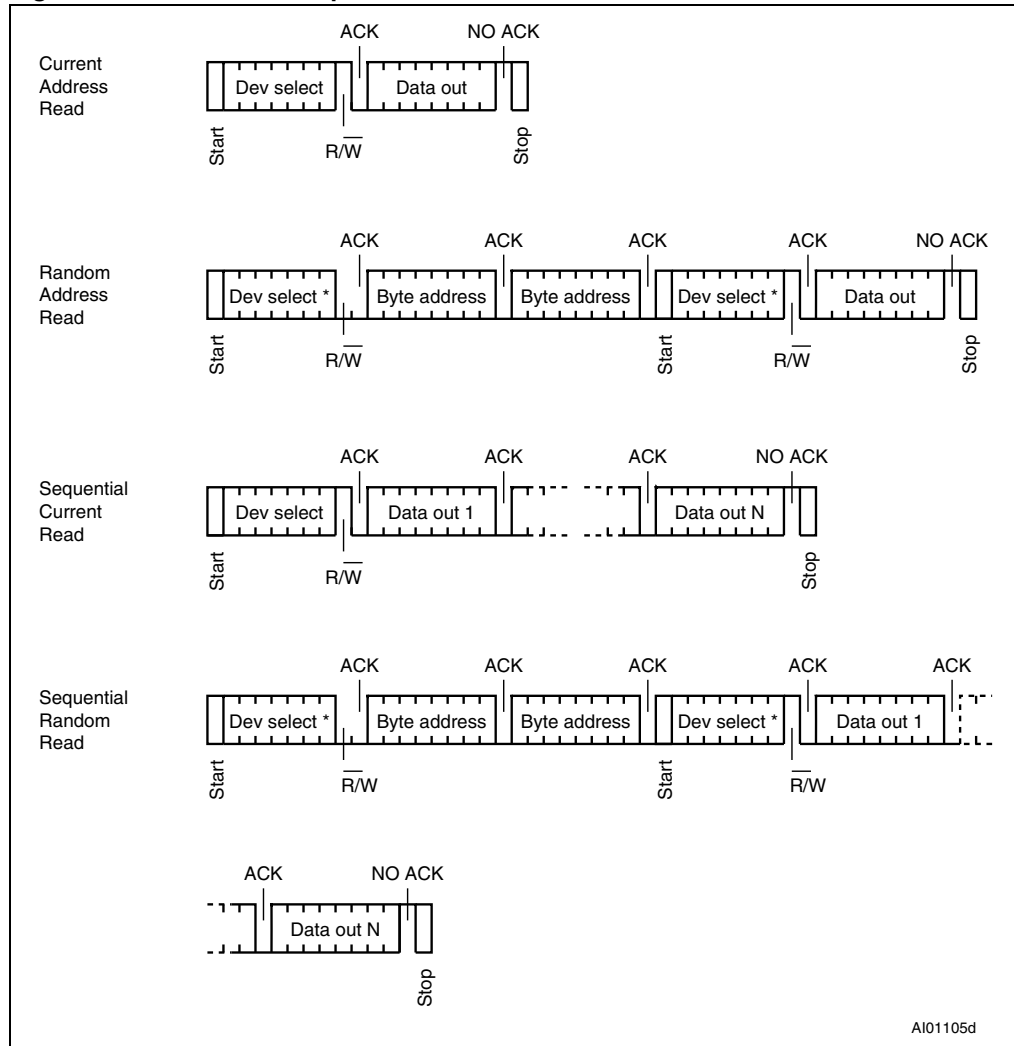
5.9 Minimizing system delays by polling on ACK

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 104](#), 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 12](#), is:

1. Initial condition: a write cycle is in progress.
2. Step 1: the bus master issues a Start condition followed by a device select code (the first byte of the new instruction).
3. Step 2: if the device is busy with the internal Write cycle, no Ack will be 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).

Figure 13. Read mode sequences



1. The seven most significant bits of the device select code of a Random Read (in the 1st and 4th bytes) must be identical.

5.10 Read operations

Read operations are performed independently of the state of the I2C_Write_Lock bit.

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.

5.11 Random Address Read

A dummy Write is first performed to load the address into this address counter (as shown in [Figure 13](#)) 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 (\overline{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.

5.12 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 (\overline{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 13](#), *without* acknowledging the byte.

5.13 Sequential Read

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 13](#).

The output data comes from consecutive addresses, with the internal address counter automatically incremented after each byte output. After the last memory address, the address counter 'rolls-over', and the device continues to output data from memory address 00h.

5.14 Acknowledge in Read mode

For all Read commands, the device waits, after each byte read, for an acknowledgment during the 9th 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 Initial delivery state

The device is delivered with all bits in the memory array set to 1 (each byte contains FFh).

7 RF device operation

The M24LR64-R is divided into 64 sectors of 32 blocks of 32 bits as shown in [Table 5](#). Each sector can be individually read- and/or write-protected using a specific lock or password command.

Read and Write operations are possible if the addressed block is not protected. During a Write, the 32 bits of the block are replaced by the new 32-bit value.

The M24LR64-R also has a 64-bit block that is used to store the 64-bit unique identifier (UID). The UID is compliant with the ISO 15963 description, and its value is used during the anticollision sequence (Inventory). This block is not accessible by the user and its value is written by ST on the production line.

The M24LR64-R also includes an AFI register in which the application family identifier is stored, and a DSFID register in which the data storage family identifier used in the anticollision algorithm is stored. The M24LR64-R has three additional 32-bit blocks in which the password codes are stored.

7.1 Commands

The M24LR64-R supports the following commands:

- **Inventory**, used to perform the anticollision sequence.
- **Stay Quiet**, used to put the M24LR64-R in quiet mode, where it does not respond to any inventory command.
- **Select**, used to select the M24LR64-R. After this command, the M24LR64-R processes all Read/Write commands with Select_flag set.
- **Reset To Ready**, used to put the M24LR64-R in the ready state.
- **Read Block**, used to output the 32 bits of the selected block and its locking status.
- **Write Block**, used to write the 32-bit value in the selected block, provided that it is not locked.
- **Read Multiple Blocks**, used to read the selected blocks and send back their value.
- **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 Info**, used to provide the system information value
- **Get Multiple Block Security Status**, used to send the security status of the selected block.
- **Initiate**, used to trigger the tag response to the Inventory Initiated sequence.
- **Inventory Initiated**, used to perform the anticollision sequence triggered by the Initiate command.
- **Write-sector Password**, used to write the 32 bits of the selected password.
- **Lock-sector Password**, used to write the Sector security status bits of the selected sector.
- **Present-sector Password**, enables the user to present a password to unprotect the user blocks linked to this password.
- **Fast Initiate**, used to trigger the tag response to the Inventory Initiated sequence.
- **Fast Inventory Initiated**, used to perform the anticollision sequence triggered by the Initiate command.
- **Fast Read Single Block**, used to output the 32 bits of the selected block and its locking status.
- **Fast Read Multiple Blocks**, used to read the selected blocks and send back their value.

7.2 Initial dialog for vicinity cards

The dialog between the vicinity coupling device or VCD (commonly the “RF reader”) and the vicinity integrated circuit card or VICC (M24LR64-R) takes place as follows:

- activation of the M24LR64-R by the RF operating field of the VCD
- transmission of a command by the VCD
- transmission of a response by the M24LR64-R

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).

7.2.1 Power transfer

Power is transferred to the M24LR64-R by radio frequency at 13.56 MHz via coupling antennas in the M24LR64-R and the VCD. The RF operating field of the VCD is transformed on the M24LR64-R antenna to an AC Voltage which is rectified, filtered and internally regulated. The amplitude modulation (ASK) on this received signal is demodulated by the ASK demodulator.

7.2.2 Frequency

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

7.2.3 Operating field

The M24LR64-R operates continuously between the minimum and maximum values of the electromagnetic field H defined in [Table 105](#). The VCD has to generate a field within these limits.

8 Communication signal from VCD to M24LR64-R

Communications between the VCD and the M24LR64-R takes place using the modulation principle of ASK (Amplitude Shift Keying). Two modulation indexes are used, 10% and 100%. The M24LR64-R decodes both. The VCD determines which index is used.

The modulation index is defined as $[a - b]/[a + b]$ where a is the peak signal amplitude and b , the minimum signal amplitude of the carrier frequency.

Depending on the choice made by the VCD, a "pause" will be created as described in [Figure 14](#) and [Figure 15](#).

The M24LR64-R is operational for any degree of modulation index from between 10% and 30%.

Figure 14. 100% modulation waveform

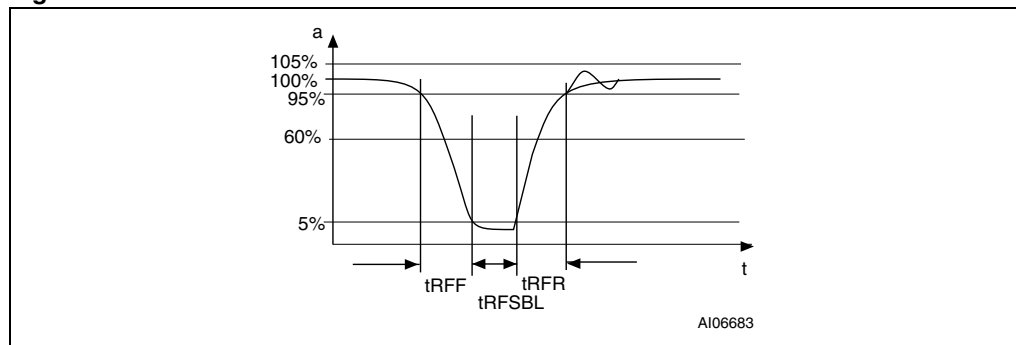
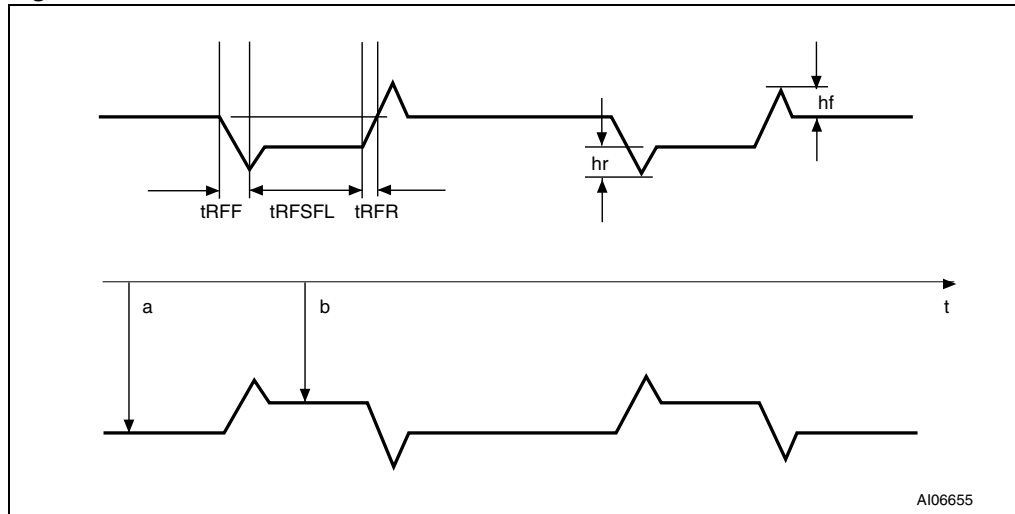


Table 16. 10% modulation parameters

| Symbol | Parameter definition | Value |
|--------|----------------------|-------|
| hr | $0.1 \times (a - b)$ | max |
| hf | $0.1 \times (a - b)$ | max |

Figure 15. 10% modulation waveform



9 Data rate and data coding

The data coding implemented in the M24LR64-R uses pulse position modulation. Both data coding modes that are described in the ISO15693 are supported by the M24LR64-R. The selection is made by the VCD and indicated to the M24LR64-R within the start of frame (SOF).

9.1 Data coding mode: 1 out of 256

The value of one single byte is represented by the position of one pause. The position of the pause on 1 of 256 successive time periods of $18.88 \mu\text{s}$ ($256/f_C$), determines the value of the byte. In this case the transmission of one byte takes 4.833 ms and the resulting data rate is 1.65 Kbits/s ($f_C/8192$).

Figure 16 illustrates this pulse position modulation technique. In this figure, data E1h (225 decimal) is sent by the VCD to the M24LR64-R.

The pause occurs during the second half of the position of the time period that determines the value, as shown in *Figure 17*.

A pause during the first period transmits the data value 00h. A pause during the last period transmit the data value FFh (255 decimal).

Figure 16. 1 out of 256 coding mode

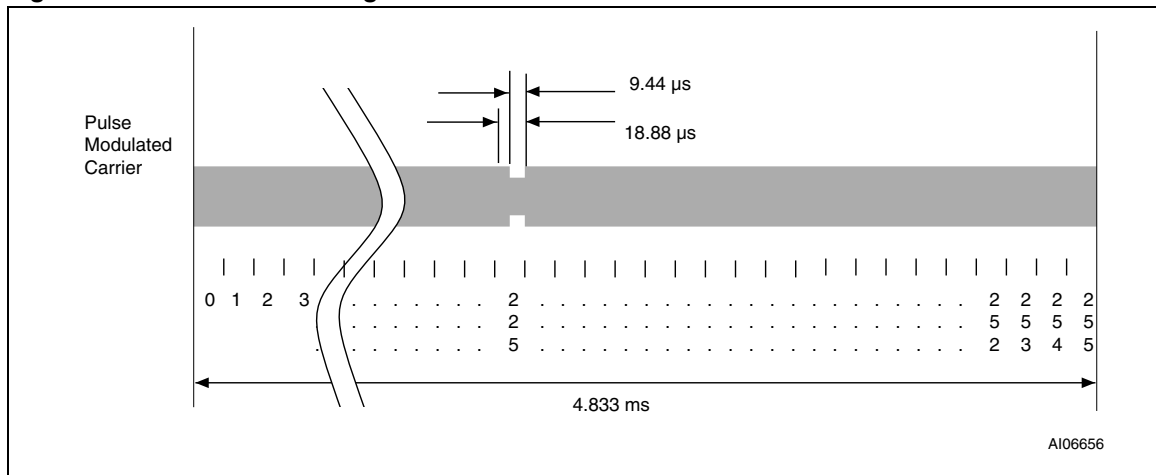
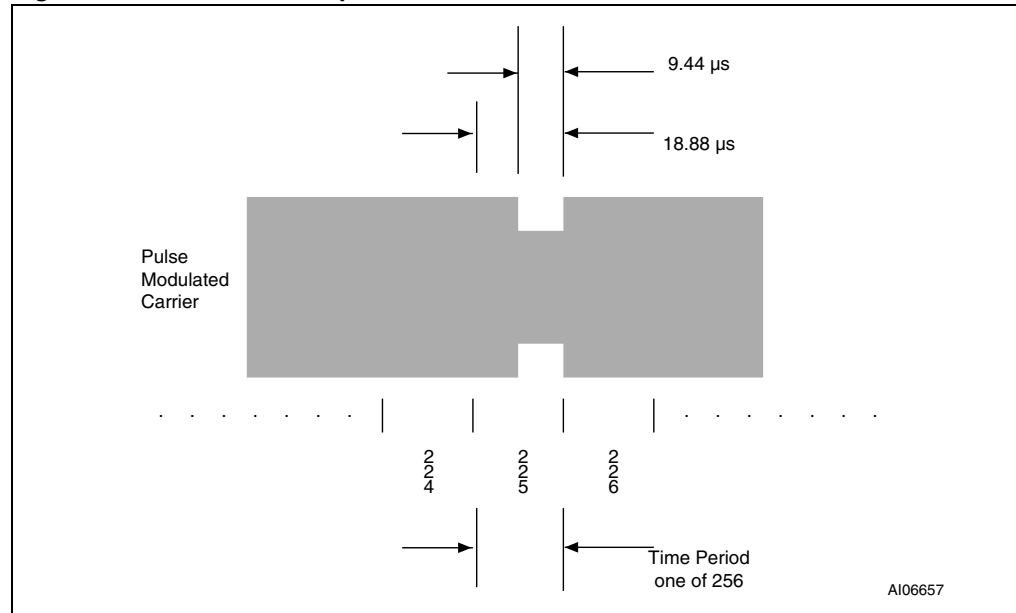


Figure 17. Detail of a time period



9.2 Data coding mode: 1 out of 4

The value of 2 bits is represented by the position of one pause. The position of the pause on 1 of 4 successive time periods of $18.88 \mu\text{s}$ ($256/f_C$), determines the value of the 2 bits. Four successive pairs of bits form a byte, where the least significant pair of bits is transmitted first.

In this case the transmission of one byte takes $302.08 \mu\text{s}$ and the resulting data rate is 26.48 Kbits/s ($f_C/512$). [Figure 18](#) illustrates the 1 out of 4 pulse position technique and coding. [Figure 19](#) shows the transmission of E1h (225d - 1110 0001b) by the VCD.

Figure 18. 1 out of 4 coding mode

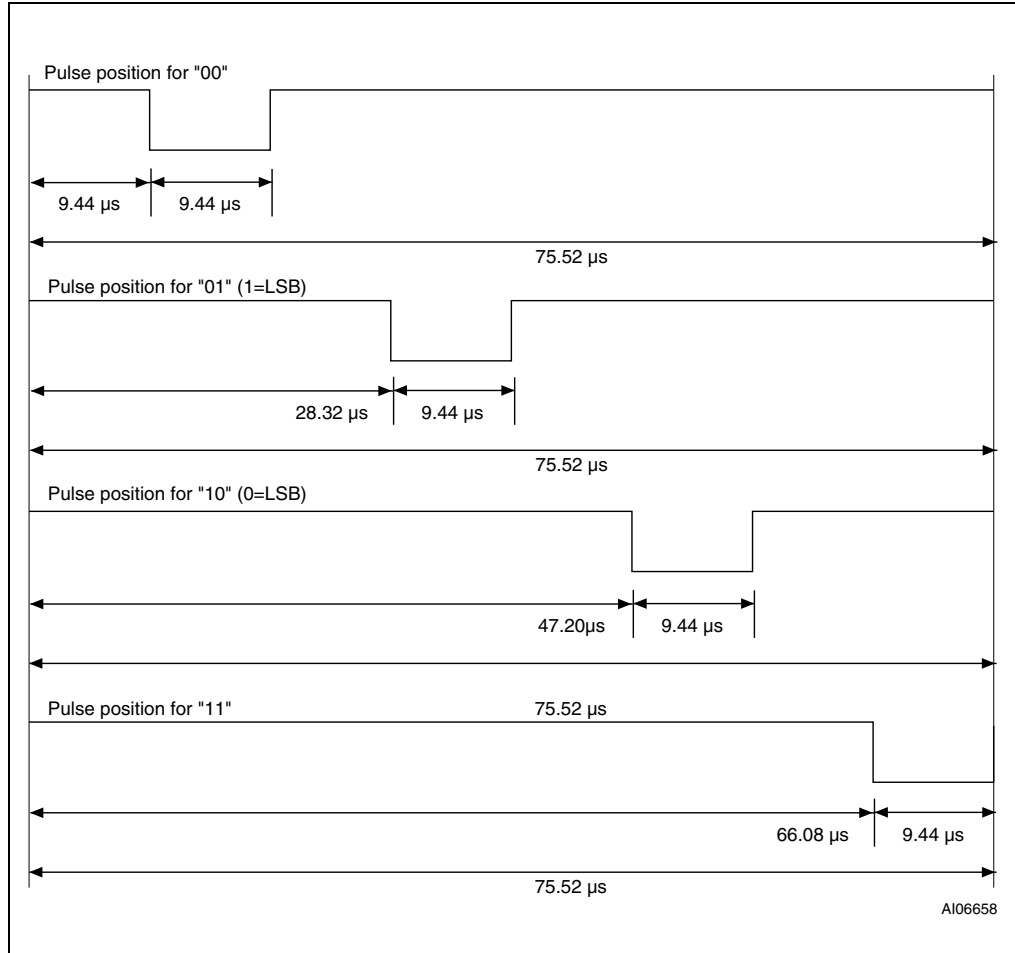
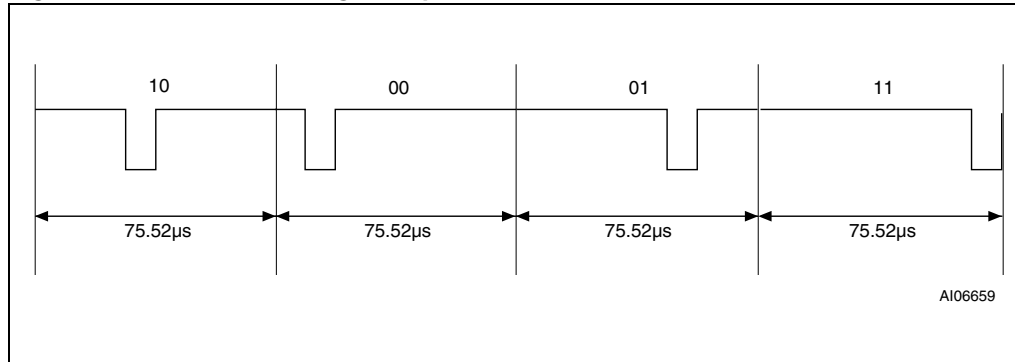


Figure 19. 1 out of 4 coding example



9.3 VCD to M24LR64-R frames

Frames are delimited by a start of frame (SOF) and an end of frame (EOF). They are implemented using code violation. Unused options are reserved for future use.

The M24LR64-R is ready to receive a new command frame from the VCD 311.5 μs (t_2) after sending a response frame to the VCD.

The M24LR64-R takes a power-up time of 0.1 ms after being activated by the powering field. After this delay, the M24LR64-R is ready to receive a command frame from the VCD.

9.4 Start of frame (SOF)

The SOF defines the data coding mode the VCD is to use for the following command frame. The SOF sequence described in [Figure 20](#) selects the 1 out of 256 data coding mode. The SOF sequence described in [Figure 21](#) selects the 1 out of 4 data coding mode. The EOF sequence for either coding mode is described in [Figure 22](#).

Figure 20. SOF to select 1 out of 256 data coding mode

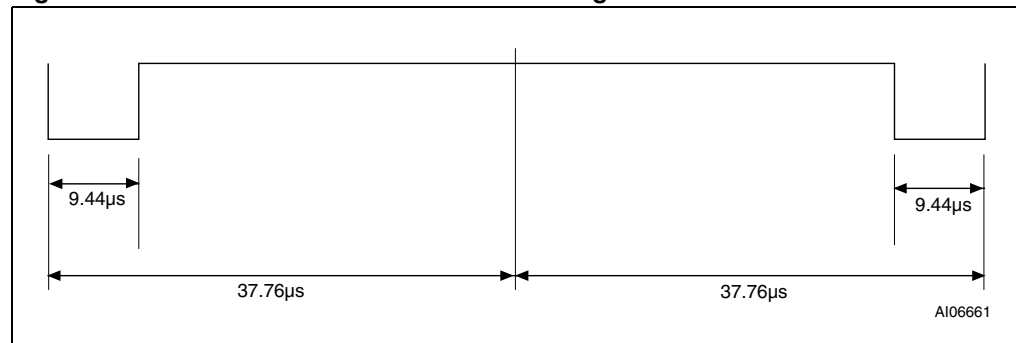


Figure 21. SOF to select 1 out of 4 data coding mode

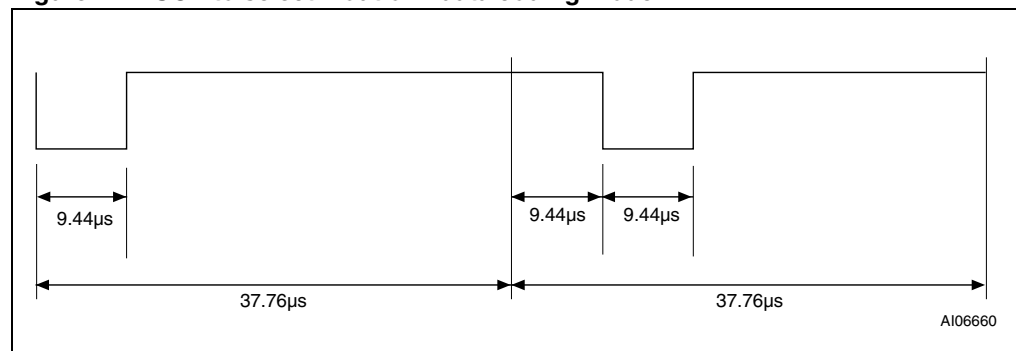
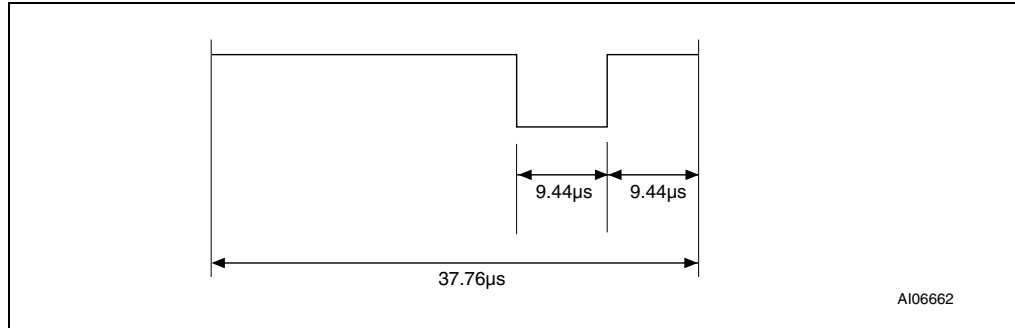


Figure 22. EOF for either data coding mode



10 Communications signal from M24LR64-R to VCD

The M24LR64-R has several modes defined for some parameters, owing to which it can operate in different noise environments and meet different application requirements.

10.1 Load modulation

The M24LR64-R is capable of communication to the VCD via an inductive coupling area whereby the carrier is loaded to generate a subcarrier with frequency f_S . The subcarrier is generated by switching a load in the M24LR64-R.

The load-modulated amplitude received on the VCD antenna must be of at least 10mV when measured as described in the test methods defined in International Standard ISO10373-7.

10.2 Subcarrier

The M24LR64-R supports the one-subcarrier and two-subcarrier response formats. These formats are selected by the VCD using the first bit in the protocol header. When one subcarrier is used, the frequency f_{S1} of the subcarrier load modulation is 423.75 kHz ($f_C/32$). When two subcarriers are used, the frequency f_{S1} is 423.75 kHz ($f_C/32$), and frequency f_{S2} is 484.28 kHz ($f_C/28$). When using the two-subcarrier mode, the M24LR64-R generates a continuous phase relationship between f_{S1} and f_{S2} .

10.3 Data rates

The M24LR64-R can respond using the low or the high data rate format. The selection of the data rate is made by the VCD using the second bit in the protocol header. It also supports the x2 mode available on all the Fast commands. [Table 17](#) shows the different data rates produced by the M24LR64-R using the different response format combinations.

Table 17. Response data rates

| Data rate | | One subcarrier | Two subcarriers |
|-----------|-------------------|-----------------------------|----------------------------|
| Low | Standard commands | 6.62 Kbit/s ($f_C/2048$) | 6.67 Kbit/s ($f_C/2032$) |
| | Fast commands | 13.24 Kbit/s ($f_C/1024$) | not applicable |
| High | Standard commands | 26.48 Kbit/s ($f_C/512$) | 26.69 Kbit/s ($f_C/508$) |
| | Fast commands | 52.97 Kbit/s ($f_C/256$) | not applicable |

11 Bit representation and coding

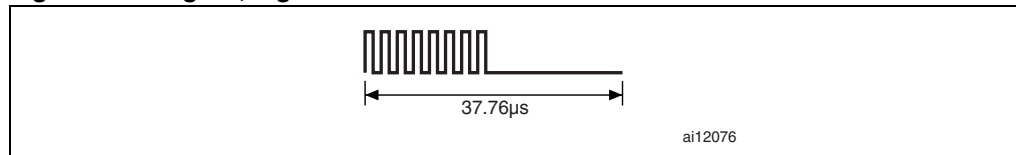
Data bits are encoded using Manchester coding, according to the following schemes. For the low data rate, same subcarrier frequency or frequencies is/are used, in this case the number of pulses is multiplied by 4 and all times will increase by this factor. For the Fast commands using one subcarrier, all pulse numbers and times are divided by 2.

11.1 Bit coding using one subcarrier

11.1.1 High data rate

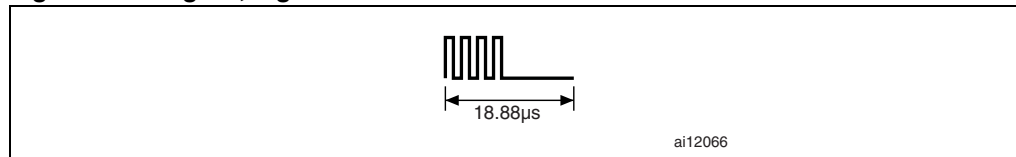
A logic 0 starts with 8 pulses at 423.75 kHz ($f_C/32$) followed by an unmodulated time of 18.88 μs as shown in [Figure 23](#).

Figure 23. Logic 0, high data rate



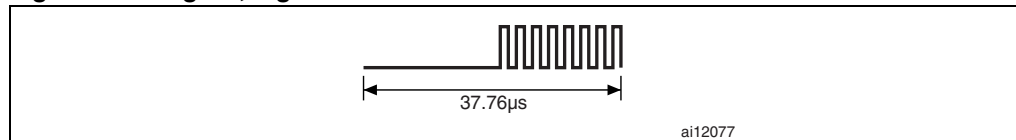
For the fast commands, a logic 0 starts with 4 pulses at 423.75 kHz ($f_C/32$) followed by an unmodulated time of 9.44 μs as shown in [Figure 24](#).

Figure 24. Logic 0, high data rate x2



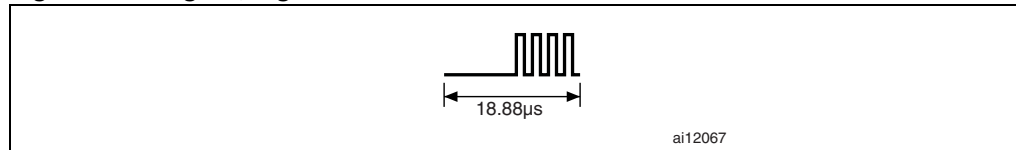
A logic 1 starts with an unmodulated time of 18.88 μs followed by 8 pulses at 423.75 kHz ($f_C/32$) as shown in [Figure 25](#).

Figure 25. Logic 1, high data rate



For the Fast commands, a logic 1 starts with an unmodulated time of 9.44 μs followed by 4 pulses of 423.75 kHz ($f_C/32$) as shown in [Figure 26](#).

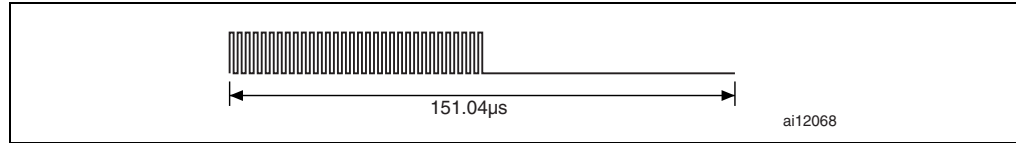
Figure 26. Logic 1, high data rate x2



11.1.2 Low data rate

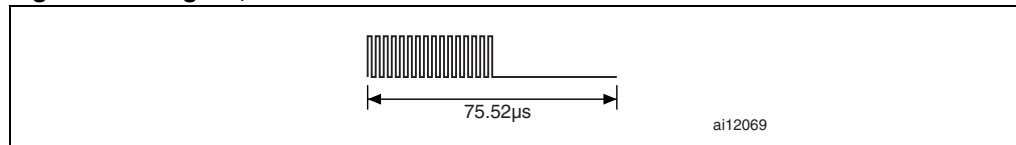
A logic 0 starts with 32 pulses at 423.75 kHz ($f_C/32$) followed by an unmodulated time of 75.52 μs as shown in [Figure 27](#).

Figure 27. Logic 0, low data rate



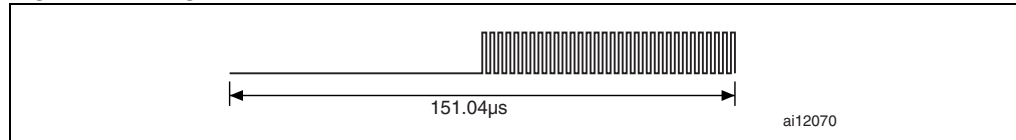
For the Fast commands, a logic 0 starts with 16 pulses at 423.75 kHz ($f_C/32$) followed by an unmodulated time of 37.76 μs as shown in [Figure 28](#).

Figure 28. Logic 0, low data rate x2



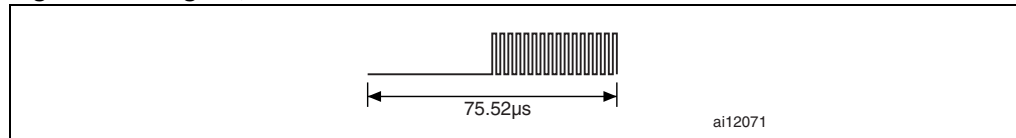
A logic 1 starts with an unmodulated time of 75.52 μs followed by 32 pulses at 423.75 kHz ($f_C/32$) as shown in [Figure 29](#).

Figure 29. Logic 1, low data rate



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 29](#).

Figure 30. Logic 1, low data rate x2

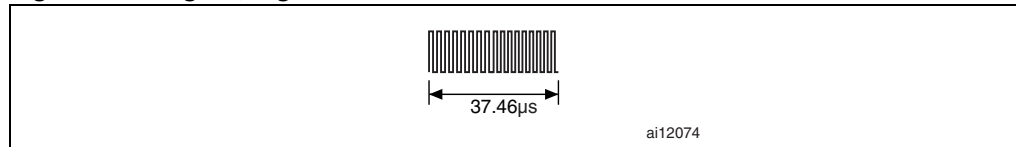


11.2 Bit coding using two subcarriers

11.3 High data rate

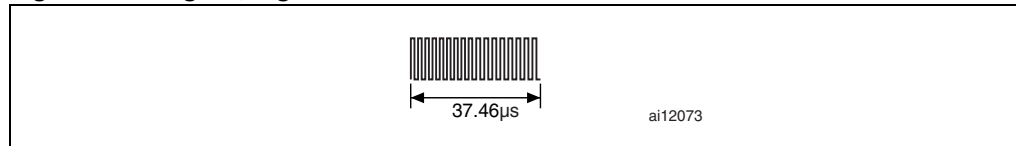
A logic 0 starts with 8 pulses at 423.75 kHz ($f_C/32$) followed by 9 pulses at 484.28 kHz ($f_C/28$) as shown in [Figure 31](#). For the Fast commands, the x2 mode is not available.

Figure 31. Logic 0, high data rate



A logic 1 starts with 9 pulses at 484.28 kHz ($f_C/28$) followed by 8 pulses at 423.75 kHz ($f_C/32$) as shown in [Figure 32](#). For the Fast commands, the x2 mode is not available.

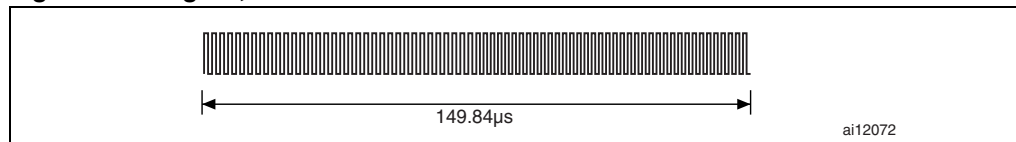
Figure 32. Logic 1, high data rate



11.4 Low data rate

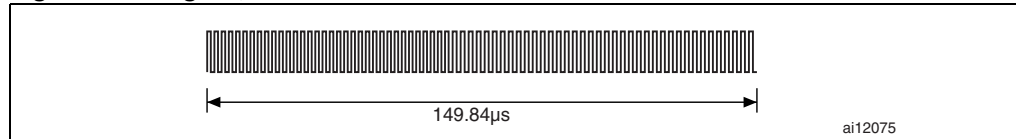
A logic 0 starts with 32 pulses at 423.75 kHz ($f_C/32$) followed by 36 pulses at 484.28 kHz ($f_C/28$) as shown in [Figure 33](#). For the Fast commands, the x2 mode is not available.

Figure 33. Logic 0, low data rate



A logic 1 starts with 36 pulses at 484.28 kHz ($f_C/28$) followed by 32 pulses at 423.75 kHz ($f_C/32$) as shown in [Figure 34](#). For the Fast commands, the x2 mode is not available.

Figure 34. Logic 1, low data rate



12 M24LR64-R to VCD frames

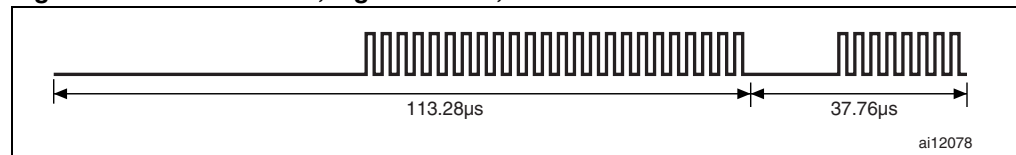
Frames are delimited by an SOF and an EOF. They are implemented using code violation. Unused options are reserved for future use. For the low data rate, the same subcarrier frequency or frequencies is/are used. In this case the number of pulses is multiplied by 4. For the Fast commands using one subcarrier, all pulse numbers and times are divided by 2.

12.1 SOF when using one subcarrier

12.2 High data rate

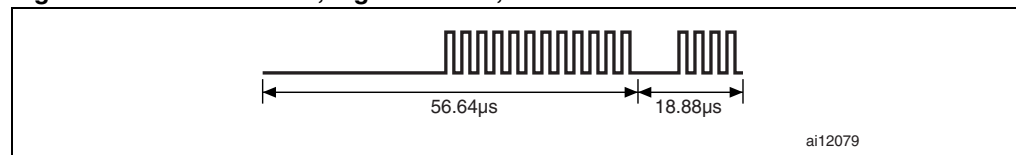
The SOF includes an unmodulated time of 56.64 μs , followed by 24 pulses at 423.75 kHz ($f_C/32$), and a logic 1 that consists of an unmodulated time of 18.88 μs followed by 8 pulses at 423.75 kHz as shown in [Figure 35](#).

Figure 35. Start of frame, high data rate, one subcarrier



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 4 pulses at 423.75 kHz as shown in [Figure 36](#).

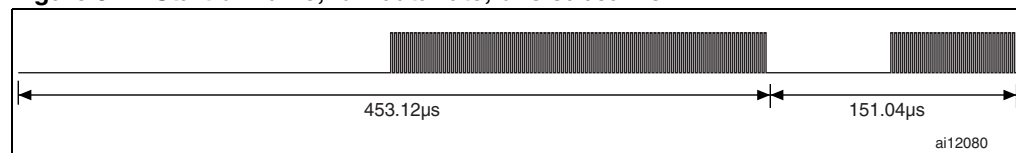
Figure 36. Start of frame, high data rate, one subcarrier x2



12.3 Low data rate

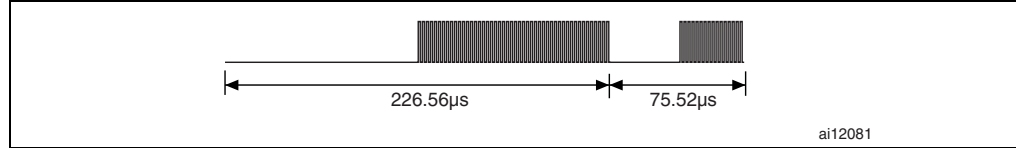
The SOF comprises an unmodulated time of 226.56 μs , followed by 96 pulses at 423.75 kHz ($f_C/32$), and a logic 1 that consists of an unmodulated time of 75.52 μs followed by 32 pulses at 423.75 kHz as shown in [Figure 37](#).

Figure 37. Start of frame, low data rate, one subcarrier



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 38](#).

Figure 38. Start of frame, low data rate, one subcarrier x2



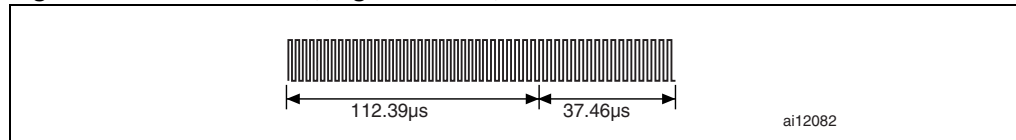
12.4 SOF when using two subcarriers

12.5 High data rate

The SOF comprises 27 pulses at 484.28 kHz ($f_C/28$), followed by 24 pulses at 423.75 kHz ($f_C/32$), and a logic 1 that includes 9 pulses at 484.28 kHz followed by 8 pulses at 423.75 kHz as shown in [Figure 39](#).

For the Fast commands, the x2 mode is not available.

Figure 39. Start of frame, high data rate, two subcarriers

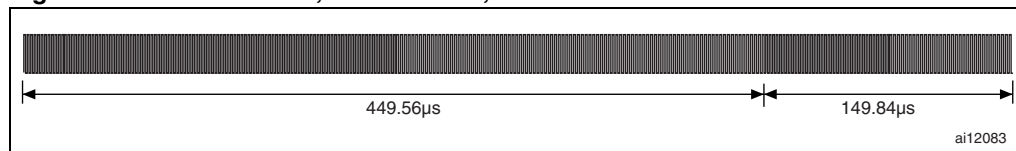


12.6 Low data rate

The SOF comprises 108 pulses at 484.28 kHz ($f_C/28$), followed by 96 pulses at 423.75 kHz ($f_C/32$), and a logic 1 that includes 36 pulses at 484.28 kHz followed by 32 pulses at 423.75 kHz as shown in [Figure 40](#).

For the Fast commands, the x2 mode is not available.

Figure 40. Start of frame, low data rate, two subcarriers

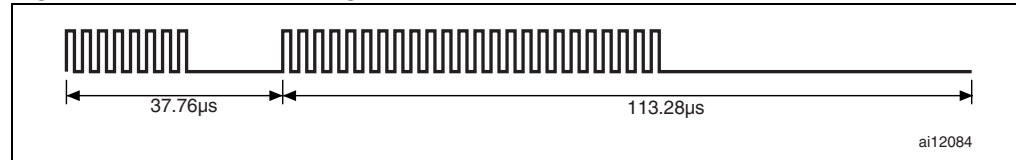


12.7 EOF when using one subcarrier

12.8 High data rate

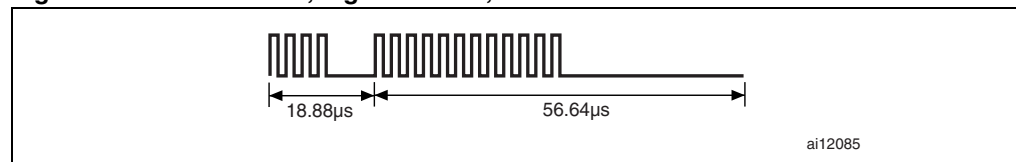
The EOF comprises a logic 0 that includes 8 pulses at 423.75 kHz and an unmodulated time of 18.88 μs , followed by 24 pulses at 423.75 kHz ($f_C/32$), and by an unmodulated time of 56.64 μs as shown in [Figure 41](#).

Figure 41. End of frame, high data rate, one subcarriers



For the Fast commands, the EOF comprises a logic 0 that includes 4 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 42](#).

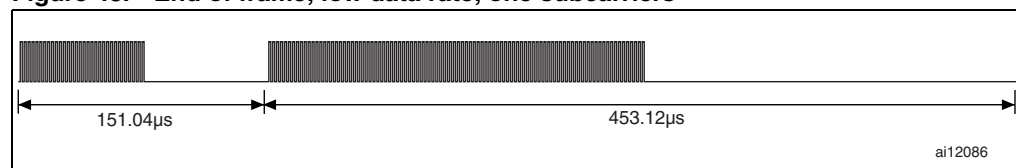
Figure 42. End of frame, high data rate, one subcarriers x2



12.9 Low data rate

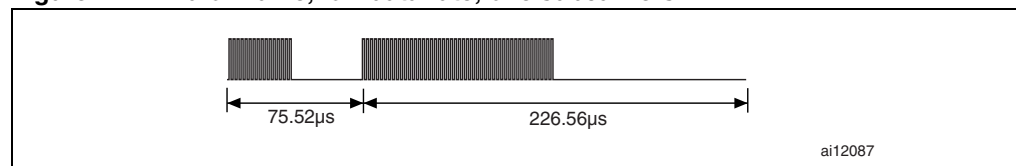
The EOF comprises a logic 0 that includes 32 pulses at 423.75 kHz and an unmodulated time of 75.52 μs , followed by 96 pulses at 423.75 kHz ($f_C/32$) and an unmodulated time of 226.56 μs as shown in [Figure 43](#).

Figure 43. End of frame, low data rate, one subcarriers



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 44](#).

Figure 44. End of frame, low data rate, one subcarriers x2



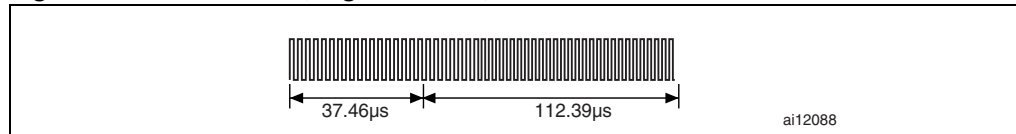
12.10 EOF when using two subcarriers

12.11 High data rate

The EOF comprises a logic 0 that includes 8 pulses at 423.75 kHz and 9 pulses at 484.28 kHz, followed by 24 pulses at 423.75 kHz ($f_D/32$) and 27 pulses at 484.28 kHz ($f_D/28$) as shown in [Figure 45](#).

For the Fast commands, the x2 mode is not available.

Figure 45. End of frame, high data rate, two subcarriers

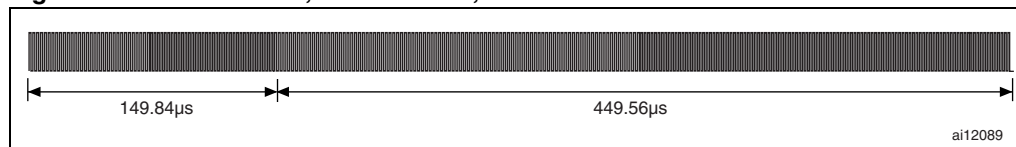


12.12 Low data rate

The EOF comprises a logic 0 that includes 32 pulses at 423.75 kHz and 36 pulses at 484.28 kHz, followed by 96 pulses at 423.75 kHz ($f_D/32$) and 108 pulses at 484.28 kHz ($f_D/28$) as shown in [Figure 46](#).

For the Fast commands, the x2 mode is not available.

Figure 46. End of frame, low data rate, two subcarriers



13 Unique identifier (UID)

The M24LR64-R 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:

- 8 MSBs with a value of E0h
- The IC manufacturer code of ST 02h, on 8 bits (ISO/IEC 7816-6/AM1)
- a unique serial number on 48 bits

Table 18. UID format

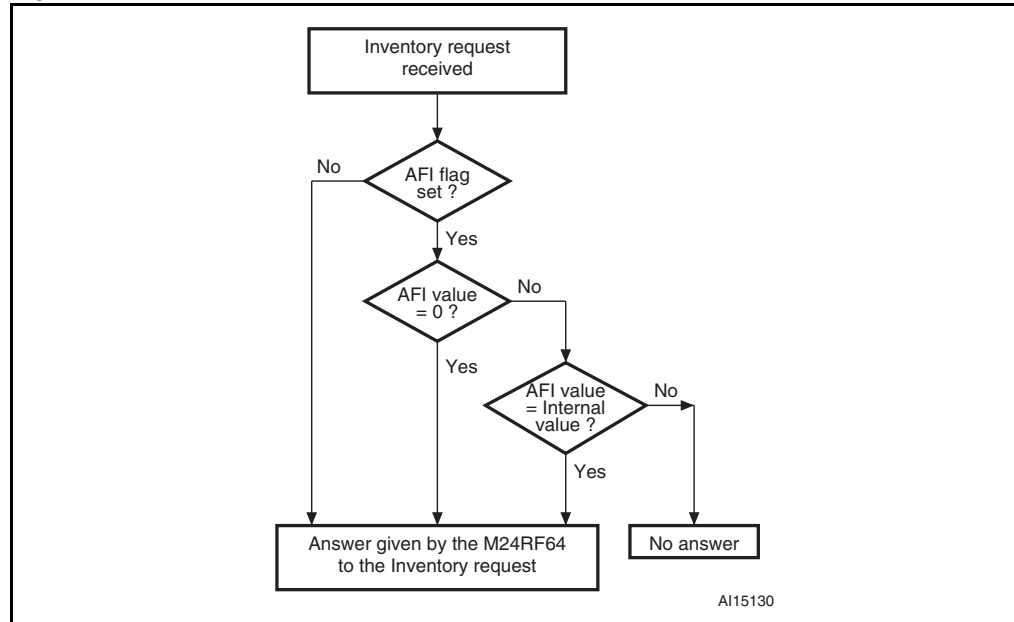
| MSB | | | | LSB | |
|------|----|------|----|----------------------|---|
| 63 | 56 | 55 | 48 | 47 | 0 |
| 0xE0 | | 0x02 | | Unique serial number | |

With the UID each M24LR64-R can be addressed uniquely and individually during the anticollision loop and for one-to-one exchanges between a VCD and an M24LR64-R.

14 Application family identifier (AFI)

The AFI (application family identifier) represents the type of application targeted by the VCD and is used to identify, among all the M24LR64-Rs present, only the M24LR64-Rs that meet the required application criteria.

Figure 47. M24LR64-R decision tree for AFI



The AFI is programmed by the M24LR64-R issuer (or purchaser) in the AFI register. Once programmed and Locked, it can no longer be modified.

The most significant nibble of the AFI is used to code one specific or all application families.

The least significant nibble of the AFI is used to code one specific or all application subfamilies. Subfamily codes different from 0 are proprietary.

(See ISO 15693-3 documentation)

15 Data storage format identifier (DSFID)

The data storage format identifier indicates how the data is structured in the M24LR64-R memory. The logical organization of data can be known instantly using the DSFID. It can be programmed and locked using the Write DSFID and Lock DSFID commands.

15.1 CRC

The CRC used in the M24LR64-R is calculated as per the definition in ISO/IEC 13239. The initial register contents are all ones: "FFFF".

The two-byte CRC are appended to each request and response, within each frame, before the EOF. The CRC is calculated on all the bytes after the SOF up to the CRC field.

Upon reception of a request from the VCD, the M24LR64-R verifies that the CRC value is valid. If it is invalid, the M24LR64-R discards the frame and does not answer to the VCD.

Upon reception of a Response from the M24LR64-R, it is recommended that the VCD verifies whether the CRC value is valid. If it is invalid, actions to be performed are left to the discretion of the VCD designer.

The CRC is transmitted least significant byte first. Each byte is transmitted least significant bit first.

Table 19. CRC transmission rules

| LSByte | | MSByte | |
|-----------------|-------|-----------------|-------|
| LSBit | MSBit | LSBit | MSBit |
| CRC 16 (8 bits) | | CRC 16 (8 bits) | |

16 M24LR64-R protocol description

The transmission protocol (or simply protocol) defines the mechanism used to exchange instructions and data between the VCD and the M24LR64-R, in both directions. It is based on the concept of “VCD talks first”.

This means that an M24LR64-R will 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 M24LR64-R
- a response from the M24LR64-R to the VCD

Each request and each response are contained in a frame. The frame delimiters (SOF, EOF) are described in [Section 12: M24LR64-R to VCD frames](#).

Each request consists of:

- a request SOF (see [Figure 20](#) and [Figure 21](#))
- flags
- a command code
- parameters, depending on the command
- application data
- a 2-byte CRC
- a request EOF (see [Figure 22](#))

Each response consists of:

- an answer SOF (see [Figure 35](#) to [Figure 40](#))
- flags
- parameters, depending on the command
- application data
- a 2-byte CRC
- an answer EOF (see [Figure 41](#) to [Figure 46](#))

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, each byte is transmitted least significant bit (LSBit) first.

The setting of the flags indicates the presence of the optional fields. When the flag is set (to one), the field is present. When the flag is reset (to zero), the field is absent.

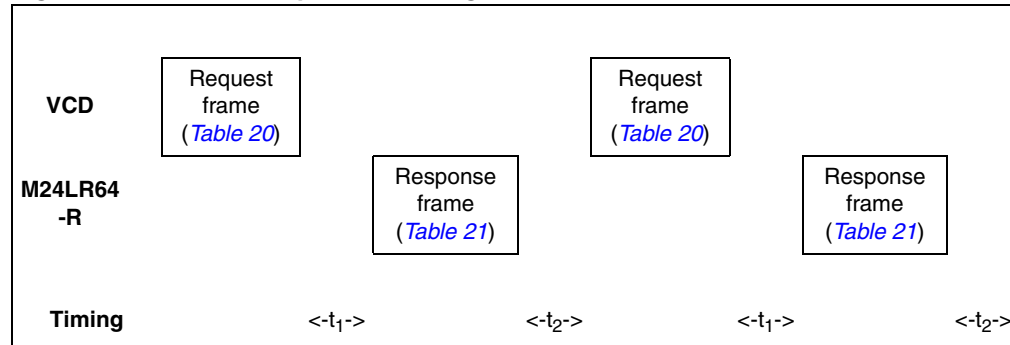
Table 20. VCD request frame format

| | | | | | | |
|-------------|---------------|--------------|------------|------|------------|-------------|
| Request SOF | Request_flags | Command code | Parameters | Data | 2-byte CRC | Request EOF |
|-------------|---------------|--------------|------------|------|------------|-------------|

Table 21. M24LR64-R Response frame format

| | | | | | |
|--------------|----------------|------------|------|------------|--------------|
| Response SOF | Response_flags | Parameters | Data | 2-byte CRC | Response EOF |
|--------------|----------------|------------|------|------------|--------------|

Figure 48. M24LR64-R protocol timing



17 M24LR64-R states

An M24LR64-R can be in one of 4 states:

- Power-off
- Ready
- Quiet
- Selected

Transitions between these states are specified in [Figure 49: M24LR64-R state transition diagram](#) and [Table 22: M24LR64-R response depending on Request_flags](#).

17.1 Power-off state

The M24LR64-R is in the Power-off state when it does not receive enough energy from the VCD.

17.2 Ready state

The M24LR64-R is in the Ready state when it receives enough energy from the VCD. When in the Ready state, the M24LR64-R answers any request where the Select_flag is not set.

17.3 Quiet state

When in the Quiet state, the M24LR64-R answers any request except for Inventory requests with the Address_flag set.

17.4 Selected state

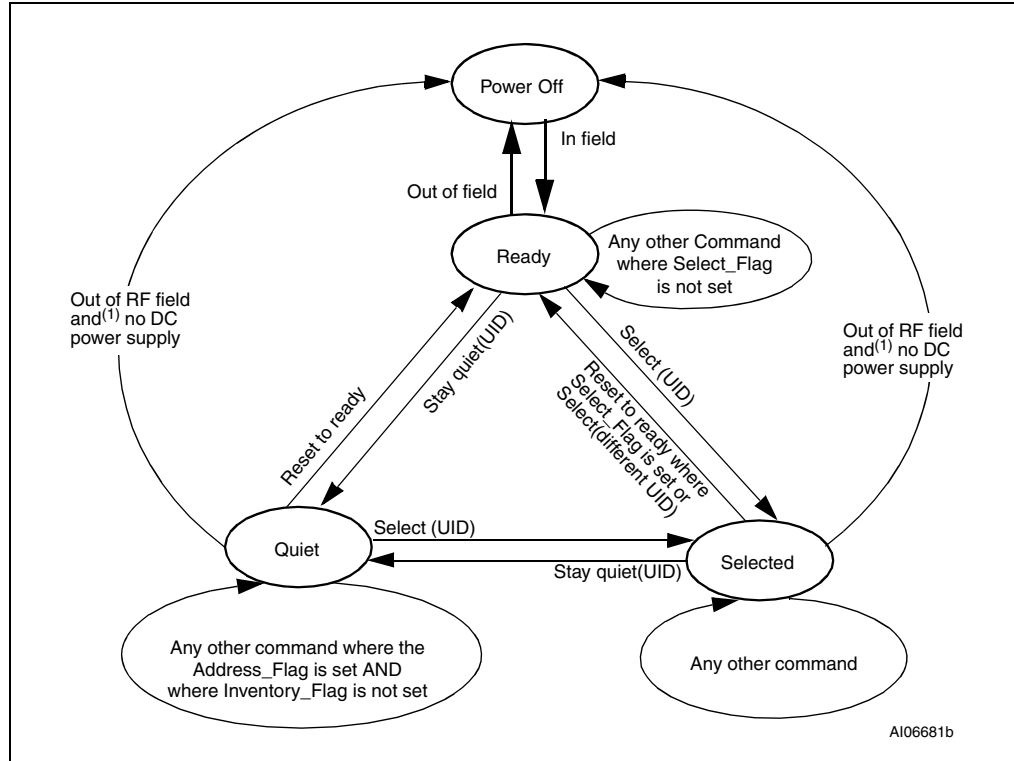
In the Selected state, the M24LR64-R answers any request in all modes (see [Section 18: Modes](#)):

- 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 22. M24LR64-R response depending on Request_flags

| Flags | Address_flag | | Select_flag | |
|--|----------------|--------------------|---------------|-------------------|
| | 1 Addressed | 0 Non addressed | 1 Selected | 0 Non selected |
| M24LR64-R in Ready or Selected state (Devices in Quiet state do not answer) | | X | | X |
| M24LR64-R in Selected state | | X | X | |
| M24LR64-R in Ready, Quiet or Selected state (the device which matches the UID) | X | | | X |
| Error (03h) | X | | X | |

Figure 49. M24LR64-R state transition diagram



1. The M24LR64-R returns to the "Power Off" state only when both conditions are met: the V_{CC} pin is not supplied (0 V or HiZ) and the tag is out of the RF field. Please refer to application note AN3057 for more information.
2. The intention of the state transition method is that only one M24LR64-R should be in the selected state at a time.

18 Modes

The term “mode” refers to the mechanism used in a request to specify the set of M24LR64-Rs that will answer the request.

18.1 Addressed mode

When the `Address_flag` is set to 1 (Addressed mode), the request contains the Unique ID (UID) of the addressed M24LR64-R.

Any M24LR64-R that receives a request with the `Address_flag` set to 1 compares the received Unique ID to its own. If it matches, then the M24LR64-R 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.

18.2 Non-addressed mode (general request)

When the `Address_flag` is cleared to 0 (Non-Addressed mode), the request does not contain a Unique ID. Any M24LR64-R receiving a request with the `Address_flag` cleared to 0 executes it and returns a response to the VCD as specified in the command description.

18.3 Select mode

When the `Select_flag` is set to 1 (Select mode), the request does not contain an M24LR64-R Unique ID. The M24LR64-R 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 M24LR64-Rs in the Selected state answer a request where the `Select_flag` set to 1.

The system design ensures in theory that only one M24LR64-R can be in the Select state at a time.

19 Request format

The request consists of:

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

Table 23. General request format

| | | | | | | |
|-------------|---------------|--------------|------------|------|-----|-------------|
| S O F | Request_flags | Command code | Parameters | Data | CRC | E O F |
|-------------|---------------|--------------|------------|------|-----|-------------|

19.1 Request flags

In a request, the “flags” field specifies the actions to be performed by the M24LR64-R and whether corresponding fields are present or not.

The flags field consists of eight bits. The bit 3 (Inventory_flag) of the request flag defines the contents of the 4 MSBs (bits 5 to 8). When bit 3 is reset (0), bits 5 to 8 define the M24LR64-R selection criteria. When bit 3 is set (1), bits 5 to 8 define the M24LR64-R Inventory parameters.

Table 24. 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 M24LR64-R |
| | | 1 | Two subcarrier are used by the M24LR64-R |
| 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 25 |
| | | 1 | The meaning of flags 5 to 8 is described in Table 26 |
| Bit 4 | Protocol_extension_flag | 0 | No Protocol format extension |
| | | 1 | Protocol format extension |

1. Subcarrier_flag refers to the M24LR64-R-to-VCD communication.

2. Data_rate_flag refers to the M24LR64-R-to-VCD communication

Table 25. Request flags 5 to 8 when Bit 3 = 0

| Bit No | Flag | Level | Description |
|--------|-----------------------------|-------|---|
| Bit 5 | Select flag ⁽¹⁾ | 0 | Request is executed by any M24LR64-R according to the setting of Address_flag |
| | | 1 | Request is executed only by the M24LR64-R in Selected state |
| Bit 6 | Address flag ⁽¹⁾ | 0 | Request is not addressed. UID field is not present. The request is executed by all M24LR64-Rs. |
| | | 1 | Request is addressed. UID field is present. The request is executed only by the M24LR64-R 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 26. Request flags 5 to 8 when Bit 3 = 1

| Bit No | 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 | |

20 Response format

The response consists of:

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

Table 27. General response format

| | | | | | |
|-------------|----------------|------------|------|-----|-------------|
| S O F | Response_flags | Parameters | Data | CRC | E O F |
|-------------|----------------|------------|------|-----|-------------|

20.1 Response flags

In a response, the flags indicate how actions have been performed by the M24LR64-R and whether corresponding fields are present or not. The response flags consist of eight bits.

Table 28. Definitions of response flags 1 to 8

| Bit No | 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 | |

20.2 Response error code

If the Error_flag is set by the M24LR64-R in the response, the Error code field is present and provides information about the error that occurred.

Error codes not specified in [Table 29](#) are reserved for future use.

Table 29. Response error code definition

| Error code | Meaning |
|------------|---|
| 02h | The command is not recognized, for example a format error occurred |
| 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 read-protected |

21 Anticollision

The purpose of the anticollision sequence is to inventory the M24LR64-Rs present in the VCD field using their unique ID (UID).

The VCD is the master of communications with one or several M24LR64-Rs. It initiates M24LR64-R communication by issuing the Inventory request.

The M24LR64-R sends its response in the determined slot or does not respond.

21.1 Request parameters

When issuing the Inventory Command, the VCD:

- sets the Nb_slots_flag as desired
- adds the mask length and the mask value after the command field
- The mask length is the number of significant bits of the mask value.
- The mask value is contained in an integer number of bytes. The mask length indicates the number of significant bits. LSB is transmitted first
- If the mask length is not a multiple of 8 (bits), as many 0-bits as required will be added to the mask value MSB so that the mask value is contained in an integer number of bytes
- The next field starts at the next byte boundary.

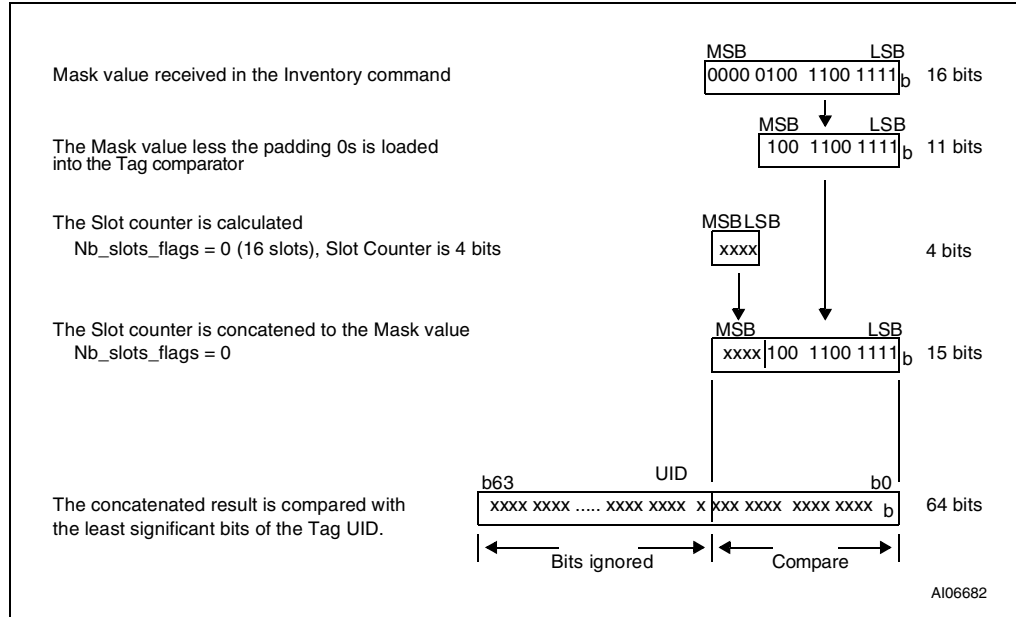
Table 30. Inventory request format

| MSB | | | | | | | LSB |
|-----|---------------|---------|--------------|-------------|--------------|---------|-----|
| SOF | Request_flags | Command | Optional AFI | Mask length | Mask value | CRC | EOF |
| | 8 bits | 8 bits | 8 bits | 8 bits | 0 to 8 bytes | 16 bits | |

In the example of the [Table 31](#) and [Figure 50](#), the mask length is 11 bits. Five 0-bits are added to the mask value MSB. The 11-bit Mask and the current slot number are compared to the UID.

Table 31. Example of the addition of 0-bits to an 11-bit mask value

| (b ₁₅) MSB | LSB (b ₀) |
|------------------------|-----------------------|
| 0000 0 | 100 1100 1111 |
| 0-bits added | 11-bit mask value |

Figure 50. Principle of comparison between the mask, the slot number and the UID

The AFI field is present if the AFI_flag is set.

The pulse is generated according to the definition of the EOF in ISO/IEC 15693-2.

The first slot starts immediately after the reception of the request EOF. To switch to the next slot, the VCD sends an EOF.

The following rules and restrictions apply:

- if no M24LR64-R answer is detected, the VCD may switch to the next slot by sending an EOF,
- if one or more M24LR64-R answers are detected, the VCD waits until the complete frame has been received before sending an EOF for switching to the next slot.

22 Request processing by the M24LR64-R

Upon reception of a valid request, the M24LR64-R performs the following algorithm:

- NbS is the total number of slots (1 or 16)
- SN is the current slot number (0 to 15)
- LSB (value, n) function returns the n Less Significant Bits of value
- MSB (value, n) function returns the n Most Significant Bits of value
- “&” is the concatenation operator
- Slot_Frame is either an SOF or an EOF

```
SN = 0
if (Nb_slots_flag)
  then NbS = 1
       SN_length = 0
       endif
  else NbS = 16
       SN_length = 4
       endif

label1:
if LSB(UID, SN_length + Mask_length) =
  LSB(SN, SN_length) & LSB(Mask, Mask_length)
  then answer to inventory request
  endif

wait (Slot_Frame)

if Slot_Frame = SOF
  then Stop Anticollision
       decode/process request
       exit
  endif

if Slot_Frame = EOF
  if SN < NbS-1
    then SN = SN + 1
         goto label1
         exit
    endif
  endif
```

23 Explanation of the possible cases

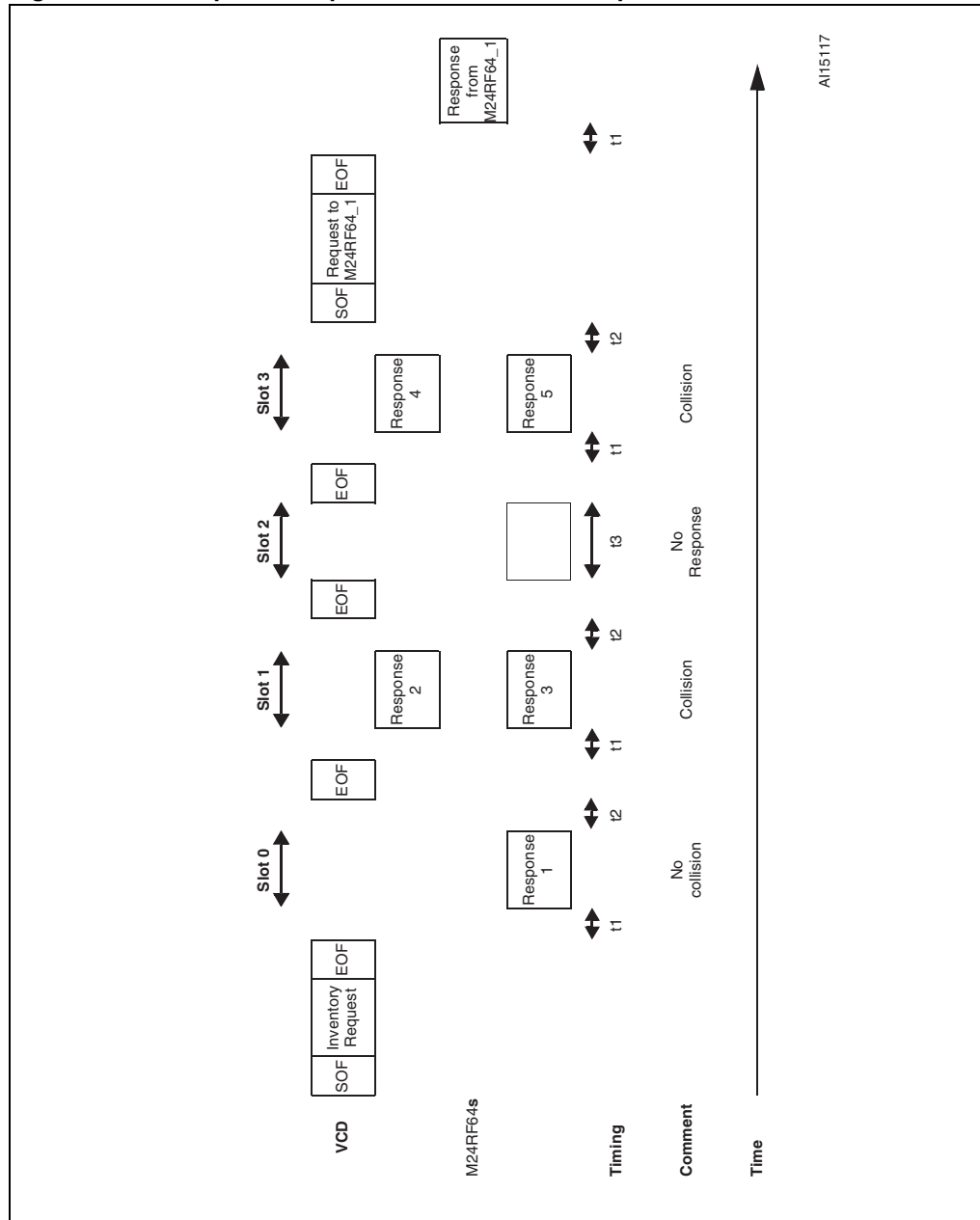
Figure 51 summarizes the main possible cases that can occur during an anticollision sequence when the slot number is 16.

The different steps are:

- The VCD sends an Inventory request, in a frame terminated by an EOF. The number of slots is 16.
- M24LR64-R_1 transmits its response in Slot 0. It is the only one to do so, therefore no collision occurs and its UID is received and registered by the VCD;
- The VCD sends an EOF in order to switch to the next slot.
- In slot 1, two M24LR64-Rs, M24LR64-R_2 and M24LR64-R_3 transmit a response, thus generating a collision. The VCD records the event and remembers that a collision was detected in Slot 1.
- The VCD sends an EOF in order to switch to the next slot.
- In Slot 2, no M24LR64-R transmits a response. Therefore the VCD does not detect any M24LR64-R SOF and decides to switch to the next slot by sending an EOF.
- In slot 3, there is another collision caused by responses from M24LR64-R_4 and M24LR64-R_5
- The VCD then decides to send a request (for instance a Read Block) to M24LR64-R_1 whose UID has already been correctly received.
- All M24LR64-Rs detect an SOF and exit the anticollision sequence. They process this request and since the request is addressed to M24LR64-R_1, only M24LR64-R_1 transmits a response.
- All M24LR64-Rs are ready to receive another request. If it is an Inventory command, the slot numbering sequence restarts from 0.

Note: *The decision to interrupt the anticollision sequence is made by the VCD. It could have continued to send EOFs until Slot 16 and only then sent the request to M24LR64-R_1.*

Figure 51. Description of a possible anticollision sequence



24 Inventory Initiated command

The M24LR64-R provides a special feature to improve the inventory time response of moving tags using the `Initiate_flag` value. This flag, controlled by the `Initiate` command, allows tags to answer to `Inventory Initiated` commands.

For applications in which multiple tags are moving in front of a reader, it is possible to miss tags using the standard inventory command. The reason is that the inventory sequence has to be performed on a global tree search. For example, a tag with a particular UID value may have to wait the run of a long tree search before being inventoried. If the delay is too long, the tag may be out of the field before it has been detected.

Using the `Initiate` command, the inventory sequence is optimized. When multiple tags are moving in front of a reader, the ones which are within the reader field will be initiated by the `Initiate` command. In this case, a small batch of tags will answer to the `Inventory Initiated` command which will optimize the time necessary to identify all the tags. When finished, the reader has to issue a new `Initiate` command in order to initiate a new small batch of tags which are new inside the reader field.

It is also possible to reduce the inventory sequence time using the `Fast Initiate` and `Fast Inventory Initiated` commands. These commands allow the M24LR64-Rs to increase their response data rate by a factor of 2, up to 53 Kbit/s.

25 Timing definition

25.1 t_1 : M24LR64-R response delay

Upon detection of the rising edge of the EOF received from the VCD, the M24LR64-R waits for a time t_{1nom} before transmitting its response to a VCD request or before switching to the next slot during an inventory process. Values of t_1 are given in [Table 32](#). The EOF is defined in [Figure 22 on page 45](#).

25.2 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 M24LR64-R responses have been received during an Inventory command. It starts from the reception of the EOF from the M24LR64-Rs.

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 M24LR64-R.

t_2 is also the time after which the VCD may send a new request to the M24LR64-R as described in [Table 48: M24LR64-R protocol timing](#).

Values of t_2 are given in [Table 32](#).

25.3 t_3 : VCD new request delay in the absence of a response from the M24LR64-R

t_3 is the time after which the VCD may send an EOF to switch to the next slot when no M24LR64-R 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 M24LR64-R.

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

- If this EOF is 100% modulated, the VCD waits a time at least equal to t_{3min} before sending a new EOF.
- If this EOF is 10% modulated, the VCD waits a time at least equal to the sum of t_{3min} + the M24LR64-R nominal response time (which depends on the M24LR64-R data rate and subcarrier modulation mode) before sending a new EOF.

Table 32. Timing values⁽¹⁾

| | Minimum (min) values | Nominal (nom) values | Maximum (max) values |
|-------|----------------------------------|----------------------|----------------------|
| t_1 | 318.6 μ s | 320.9 μ s | 323.3 μ s |
| t_2 | 309.2 μ s | No t_{nom} | No t_{max} |
| t_3 | $t_{1max}^{(2)} + t_{SOF}^{(3)}$ | No t_{nom} | No t_{max} |

1. The tolerance of specific timings is $\pm 32/f_C$.

2. t_{1max} does not apply for write alike requests. Timing conditions for write alike requests are defined in the command description.

3. t_{SOF} is the time taken by the M24LR64-R to transmit an SOF to the VCD. t_{SOF} depends on the current data rate: High data rate or Low data rate.

26 Commands codes

The M24LR64-R supports the commands described in this section. Their codes are given in [Table 33](#).

Table 33. Command codes

| Command code standard | Function |
|-----------------------|---------------------|
| 01h | Inventory |
| 02h | Stay Quiet |
| 20h | Read Single Block |
| 21h | Write Single Block |
| 23h | Read Multiple Block |
| 25h | Select |
| 26h | Reset to Ready |
| 27h | Write AFI |
| 28h | Lock AFI |
| 29h | Write DSFID |
| 2Ah | Lock DSFID |
| 2Bh | Get System Info |

| Command code custom | Function |
|---------------------|------------------------------------|
| 2Ch | Get Multiple Block Security Status |
| B1h | Write-sector Password |
| B2h | Lock-sector Password |
| B3h | Present-sector Password |
| C0h | Fast Read Single Block |
| C1h | Fast Inventory Initiated |
| C2h | Fast Initiate |
| C3h | Fast Read Multiple Block |
| D1h | Inventory Initiated |
| D2h | Initiate |

26.1 Inventory

When receiving the Inventory request, the M24LR64-R runs the anticollision sequence. The `Inventory_flag` is set to 1. The meaning of flags 5 to 8 is shown in [Table 26: Request flags 5 to 8 when Bit 3 = 1](#).

The request contains:

- the flags,
- the Inventory command code (see [Table 33: Command codes](#))
- the AFI if the AFI flag is set
- the mask length
- the mask value
- the CRC

The M24LR64-R does not generate any answer in case of error.

Table 34. 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 35. 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 M24LR64-R response, it waits 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:

$$t_{3\min} = 4384/f_C (323.3\mu\text{s}) + t_{\text{SOF}}$$
- If the VCD sends a 10% modulated EOF, the minimum value of t_3 is:

$$t_{3\min} = 4384/f_C (323.3\mu\text{s}) + t_{\text{NRT}}$$

where:

- t_{SOF} is the time required by the M24LR64-R to transmit an SOF to the VCD
- t_{NRT} is the nominal response time of the M24LR64-R

t_{NRT} and t_{SOF} are dependent on the M24LR64-R-to-VCD data rate and subcarrier modulation mode.

26.2 Stay Quiet

Command code = 0x02

On receiving the Stay Quiet command, the M24LR64-R 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 M24LR64-R does not process any request if the Inventory_flag is set,
- the M24LR64-R processes any Addressed request

The M24LR64-R exits the Quiet State when:

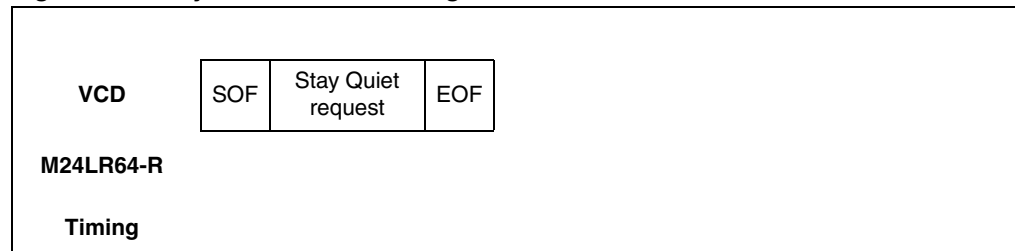
- it is reset (power off),
- receiving a Select request. It then goes to the Selected state,
- receiving a Reset to Ready request. It then goes to the Ready state.

Table 36. 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 52. Stay Quiet frame exchange between VCD and M24LR64-R



26.3 Read Single Block

On receiving the Read Single Block command, the M24LR64-R reads the requested block and sends back its 32-bit value in the response. The Protocol_extention_flag should be set to 1 for the M24LR64-R to operate correctly. If the Protocol_extention_flag is at 0, the M24LR64-R answers with an error code. The Option_flag is supported.

Table 37. Read Single Block request format

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

1. Gray means that the field is optional.

Request parameters:

- Option_flag
- UID (optional)
- Block number

Table 38. Read Single Block response format when Error_flag is NOT set

| Response SOF | Response flags | Sector security status ⁽¹⁾ | Data | CRC16 | Response EOF |
|--------------|----------------|---------------------------------------|---------|---------|--------------|
| | 8 bits | 8 bits | 32 bits | 16 bits | |

1. Gray means that the field is optional.

Response parameters:

- Sector security status if Option_flag is set (see [Table 39: Sector security status](#))
- 4 bytes of block data

Table 39. Sector 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 | | | password control bits | | Read / Write protection bits | | 0: Current sector not locked 1: Current sector locked |

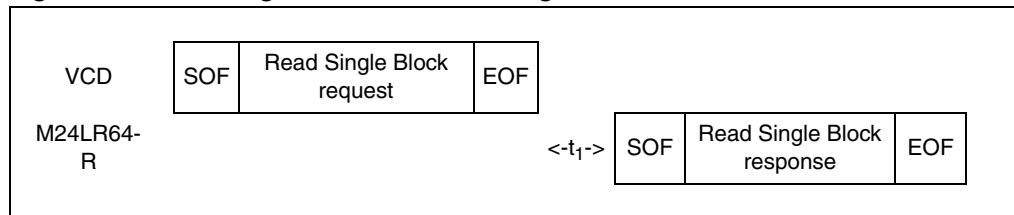
Table 40. Read Single Block response format when Error_flag is set

| 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: the 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 53. Read Single Block frame exchange between VCD and M24LR64-R



26.4 Write Single Block

On receiving the Write Single Block command, the M24LR64-R writes the data contained in the request to the requested block and reports whether the write operation was successful in the response. The Protocol_extention_flag should be set to 1 for the M24LR64-R to operate correctly. If the Protocol_extention_flag is at 0, the M24LR64-R answers with an error code. The Option_flag is supported.

During the RF write cycle W_t , there should be no modulation (neither 100% nor 10%). Otherwise, the M24LR64-R may not program correctly the data into the memory. The W_t time is equal to $t_{1nom} + 18 \times 302 \mu s$.

Table 41. 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 | 16 bits | 32 bits | 16 bits | |

1. Gray means that the field is optional.

Request parameters:

- UID (optional)
- Block number
- Data

Table 42. 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 send back after the writing cycle.

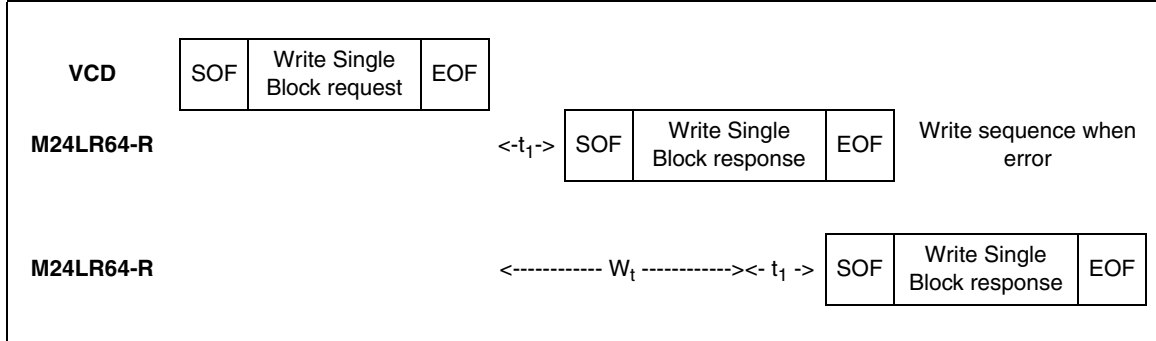
Table 43. Write Single Block response format when Error_flag is set

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

Response parameter:

- Error code as Error_flag is set:
 - 03h: the 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 54. Write Single Block frame exchange between VCD and M24LR64-R



26.5 Read Multiple Block

When receiving the Read Multiple Block command, the M24LR64-R reads the selected blocks and sends back their value in multiples of 32 bits in the response. The blocks are numbered from '00h to '7FFh' 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, 7 blocks are read. The maximum number of blocks is fixed at 32 assuming that they are all located in the same sector. If the number of blocks overlaps sectors, the M24LR64-R returns an error code. The Protocol_extention_flag should be set to 1 for the M24LR64-R to operate correctly. If the Protocol_extention_flag is at 0, the M24LR64-R answers with an error code. The Option_flag is supported.

Table 44. 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 | 16 bits | 8 bits | 16 bits | |

1. Gray means that the field is optional.

Request parameters:

- Option_flag
- UID (optional)
- First block number
- Number of blocks

Table 45. Read Multiple Block response format when Error_flag is NOT set

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

1. Gray means that the field is optional.

2. Repeated as needed.

Response parameters:

- Sector security status if Option_flag is set (see [Table 46: Sector security status](#))
- N blocks of data

Table 46. Sector security status

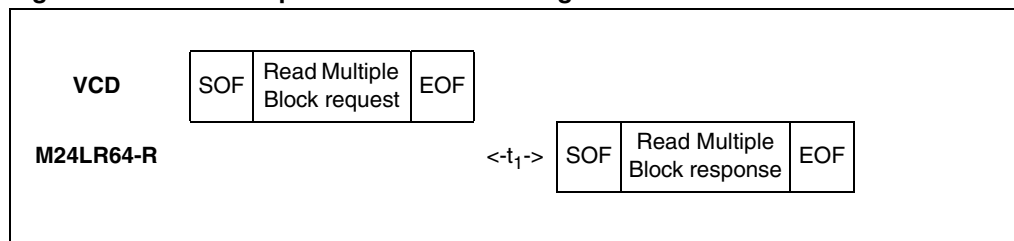
| b ₇ | b ₆ | b ₅ | b ₄ | b ₃ | b ₂ | b ₁ | b ₀ |
|-----------------------------------|----------------|----------------|-----------------------|----------------|------------------------------|----------------|--|
| Reserved for future use. All at 0 | | | password control bits | | Read / Write protection bits | | 0: Current sector not locked 1: Current sector locked |

Table 47. 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: the 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 55. Read Multiple Block frame exchange between VCD and M24LR64-R

26.6 Select

When receiving the Select command:

- if the UID is equal to its own UID, the M24LR64-R enters or stays in the Selected state and sends a response.
- if the UID does not match its own, the selected M24LR64-R returns to the Ready state and does not send a response.

The M24LR64-R 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 M24LR64-R remains in its current state.

Table 48. Select request format

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

Request parameter:

- UID

Table 49. 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.

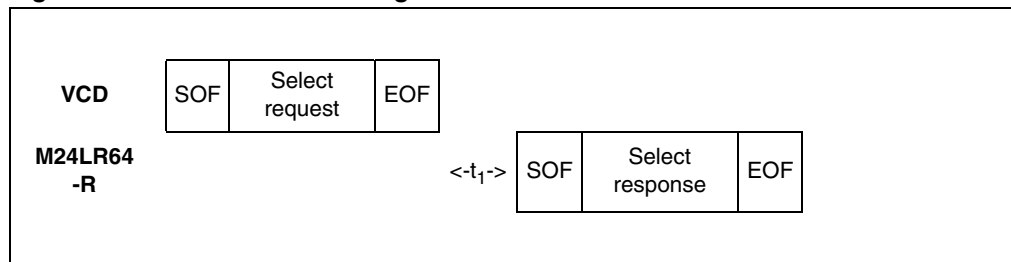
Table 50. Select response format when Error_flag is set

| 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 56. Select frame exchange between VCD and M24LR64-R



26.7 Reset to Ready

On receiving a Reset to Ready command, the M24LR64-R returns to the Ready state if no error occurs. In the Addressed mode, the M24LR64-R answers an error code only if the UID is equal to its own UID. If not, no response is generated.

Table 51. 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 means that the field is optional.

Request parameter:

- UID (optional)

Table 52. 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

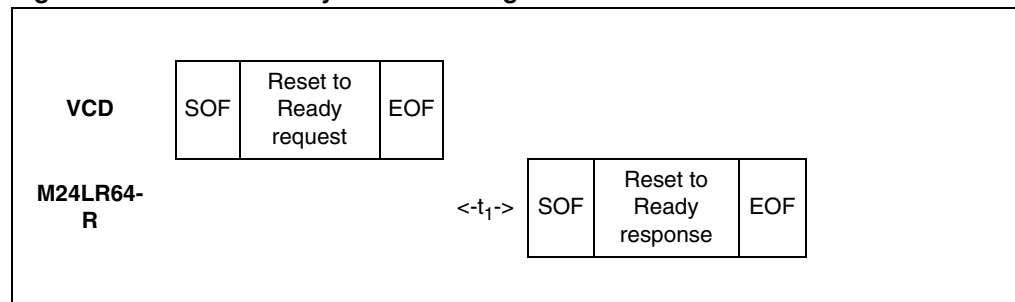
Table 53. Reset to ready response format when Error_flag is set

| 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 57. Reset to Ready frame exchange between VCD and M24LR64-R



26.8 Write AFI

On receiving the Write AFI request, the M24LR64-R programs the 8-bit AFI value to its memory. The Option_flag is supported.

During the RF write cycle W_t , there should be no modulation (neither 100% nor 10%). Otherwise, the M24LR64-R may not write correctly the AFI value into the memory. The W_t time is equal to $t_{1nom} + 18 \times 302 \mu s$.

Table 54. 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 means that the field is optional.

Request parameter:

- UID (optional)
- AFI

Table 55. 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.

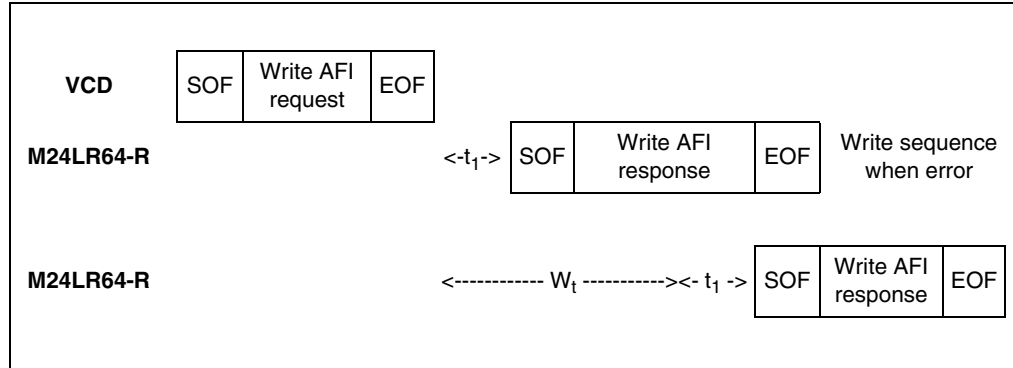
Table 56. Write AFI response format when Error_flag is set

| 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: the 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 58. Write AFI frame exchange between VCD and M24LR64-R



26.9 Lock AFI

On receiving the Lock AFI request, the M24LR64-R locks the AFI value permanently. The Option_flag is supported.

During the RF write cycle W_t , there should be no modulation (neither 100% nor 10%). Otherwise, the M24LR64-R may not Lock correctly the AFI value in memory. The W_t time is equal to $t_{1nom} + 18 \times 302 \mu s$.

Table 57. Lock AFI request format

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

1. Gray means that the field is optional.

Request parameter:

- UID (optional)

Table 58. 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

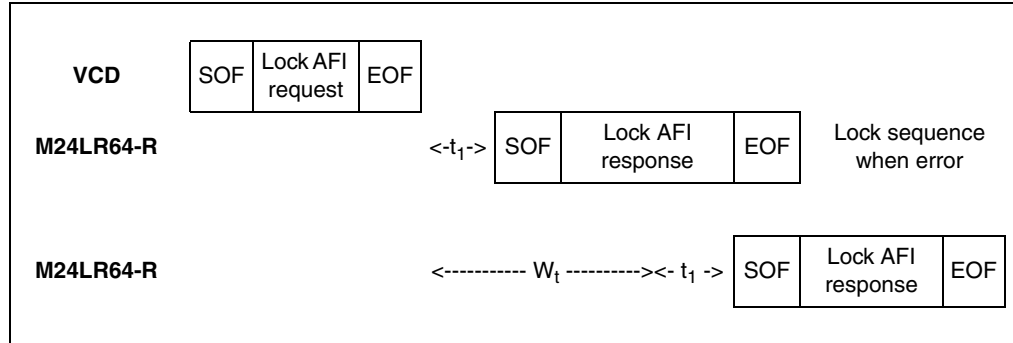
Table 59. Lock AFI response format when Error_flag is set

| 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: the 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 59. Lock AFI frame exchange between VCD and M24LR64-R



26.10 Write DSFID

On receiving the Write DSFID request, the M24LR64-R programs the 8-bit DSFID value to its memory. The Option_flag is supported.

During the RF write cycle W_t , there should be no modulation (neither 100% nor 10%). Otherwise, the M24LR64-R may not write correctly the DSFID value in memory. The W_t time is equal to $t_{1nom} + 18 \times 302 \mu s$.

Table 60. 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 means that the field is optional.

Request parameter:

- UID (optional)
- DSFID

Table 61. 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

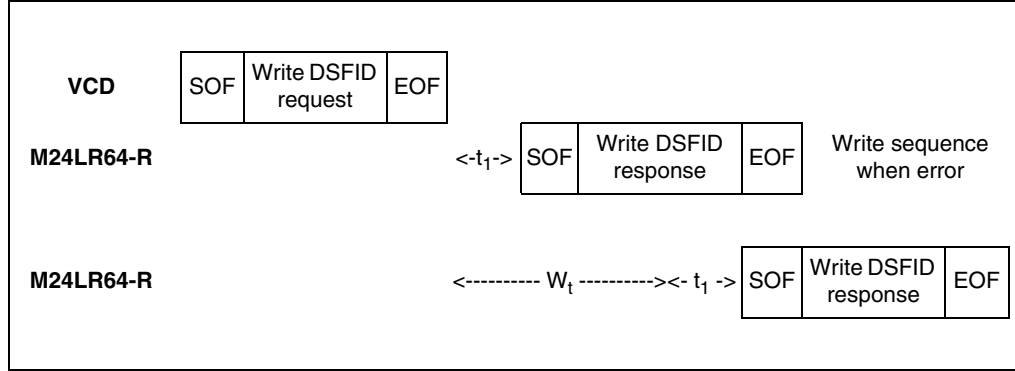
Table 62. Write DSFID response format when Error_flag is set

| 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: the 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 60. Write DSFID frame exchange between VCD and M24LR64-R



26.11 Lock DSFID

On receiving the Lock DSFID request, the M24LR64-R locks the DSFID value permanently. The Option_flag is supported.

During the RF write cycle W_t , there should be no modulation (neither 100% nor 10%). Otherwise, the M24LR64-R may not lock correctly the DSFID value in memory. The W_t time is equal to $t_{1nom} + 18 \times 302 \mu s$.

Table 63. Lock DSFID request format

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

1. Gray means that the field is optional.

Request parameter:

- UID (optional)

Table 64. 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.

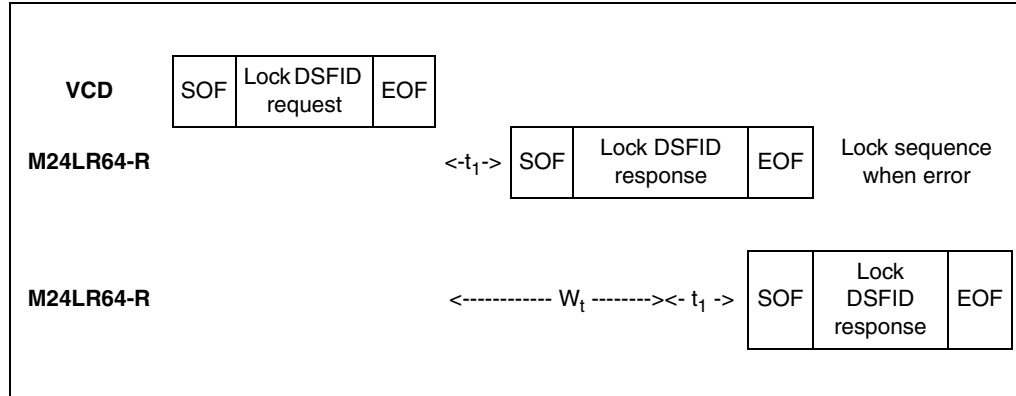
Table 65. Lock DSFID response format when Error_flag is set

| 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: the 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 61. Lock DSFID frame exchange between VCD and M24LR64-R



26.12 Get System Info

When receiving the Get System Info command, the M24LR64-R sends back its information data in the response. The Option_flag is supported and must be reset to 0. The Get System Info can be issued in both Addressed and Non Addressed modes.

The Protocol_extention_flag should be set to 1 for the M24LR64-R to operate correctly. If the Protocol_extention_flag is at 0, the M24LR64-R answers with an error code.

Table 66. 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 means that the field is optional.

Request parameter:

- UID (optional)

Table 67. Get System Info response format when Error_flag is NOT set

| Response SOF | Response_flags | Information flags | UID | DSFID | AFI | Memory Size | IC reference | CRC16 | Response EOF |
|--------------|----------------|-------------------|---------|--------|--------|-------------|--------------|---------|--------------|
| | 00h | 0Fh | 64 bits | 8 bits | 8 bits | 0307FFh | 2Ch | 16 bits | |

Response parameters:

- Information flags set to 0Fh. DSFID, AFI, Memory Size and IC reference fields are present
- UID code on 64 bits
- DSFID value
- AFI value
- Memory size. The M24LR64-R provides 2048 blocks (07FFh) of 4 byte (03h)
- IC reference. Only the 6 MSB are significant.

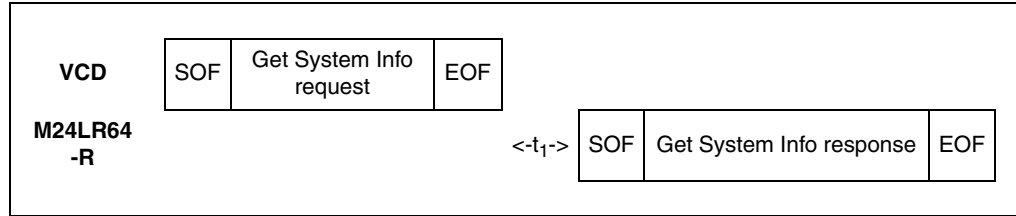
Table 68. Get System Info response format when Error_flag is set

| 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: other error

Figure 62. Get System Info frame exchange between VCD and M24LR64-R



26.13 Get Multiple Block Security Status

When receiving the Get Multiple Block Security Status command, the M24LR64-R sends back the sector security status. The blocks are numbered from '00h to '07FFh' 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 7 blocks.

The Protocol_extention_flag should be set to 1 for the M24LR64-R to operate correctly. If the Protocol_extention_flag is at 0, the M24LR64-R answers with an error code.

During the M24LR64-R response, if the internal block address counter reaches 07FFh, it rolls over to 0000h and the Sector Security Status bytes for that location are sent back to the reader.

Table 69. 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 | 16 bits | 16 bits | 16 bits | |

1. Gray means that the field is optional.

Request parameter:

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

Table 70. Get Multiple Block Security Status response format when Error_flag is NOT set

| Response SOF | Response _flags | Sector security status | CRC16 | Response EOF |
|--------------|-----------------|------------------------|---------|--------------|
| | 8 bits | 8 bits ⁽¹⁾ | 16 bits | |

1. Repeated as needed.

Response parameters:

- Sector security status (see [Table 71: Sector security status](#))

Table 71. Sector 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 | | | password control bits | Read / Write protection bits | | 0: Current sector not locked 1: Current sector locked | |

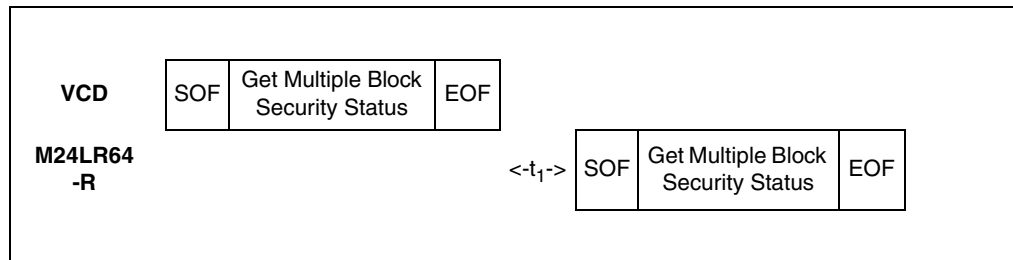
Table 72. Get Multiple Block Security Status response format when Error_flag is set

| 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 63. Get Multiple Block Security Status frame exchange between VCD and M24LR64-R



26.14 Write-sector Password

On receiving the Write-sector Password command, the M24LR64-R uses the data contained in the request to write the password and reports whether the operation was successful in the response. The Option_flag is supported.

During the RF write cycle time, W_t , there must be no modulation at all (neither 100% nor 10%). Otherwise, the M24LR64-R may not correctly program the data into the memory. The W_t time is equal to $t_{1nom} + 18 \times 302 \mu s$. 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 M24LR64-R power-down.

Table 73. Write-sector Password request format

| Request SOF | Request_flags | Write-sector Password | IC Mfg code | UID ⁽¹⁾ | Password number | Data | CRC16 | Request EOF |
|-------------|---------------|-----------------------|-------------|--------------------|-----------------|---------|---------|-------------|
| | 8 bits | B1h | 02h | 64 bits | 8 bits | 32 bits | 16 bits | |

1. Gray means that the field is optional.

Request parameter:

- UID (optional)
- Password number (01h = Pswd1, 02h = Pswd2, 03h = Pswd3, other = Error)
- Data

Table 74. Write-sector Password response format when Error_flag is NOT set

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

Response parameter:

- 32-bit password value. The response is sent back after the write cycle.

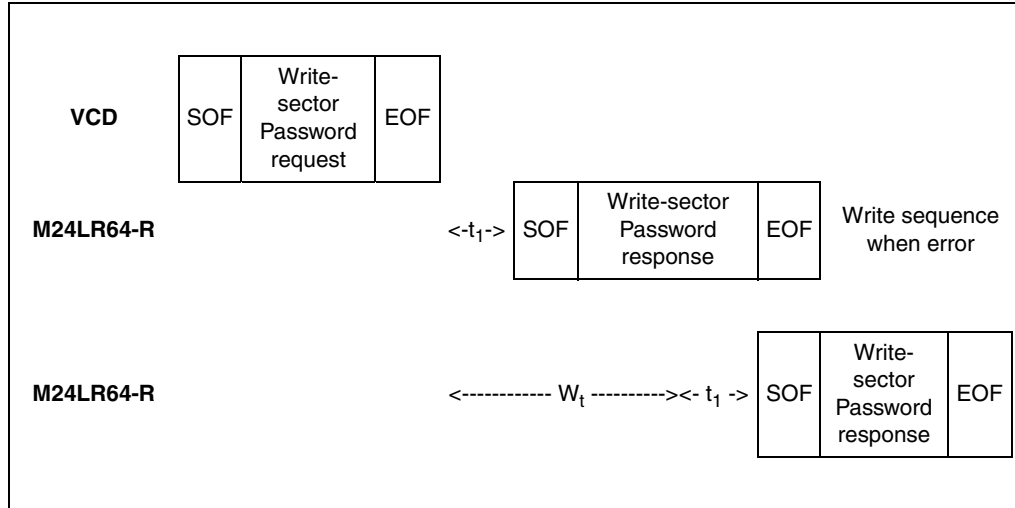
Table 75. Write-sector Password response format when Error_flag is set

| 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: the command is not recognized, for example: a format error occurred
 - 03h: the 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 64. Write-sector Password frame exchange between VCD and M24LR64-R



26.15 Lock-sector Password

On receiving the Lock-sector Password command, the M24LR64-R sets the access rights and permanently locks the selected sector. The Option_flag is supported.

A sector is selected by giving the address of one of its blocks in the Lock-sector Password request (Sector number field). For example, addresses 0 to 31 are used to select sector 0 and addresses 32 to 63 are used to select sector 1. Care must be taken when issuing the Lock-sector Password command as all the blocks belonging to the same sector are automatically locked by a single command.

The Protocol_extention_flag should be set to 1 for the M24LR64-R to operate correctly. If the Protocol_extention_flag is at 0, the M24LR64-R answers with an error code.

During the RF write cycle W_t , there should be no modulation (neither 100% nor 10%) otherwise, the M24LR64-R may not correctly lock the memory block.

The W_t time is equal to $t_{1nom} + 18 \times 302 \mu s$.

Table 76. Lock-sector Password request format

| Request SOF | Request _flags | Lock-sector Password | IC Mfg code | UID ⁽¹⁾ | Sector number | Sector security status | CRC16 | Request EOF |
|-------------|----------------|----------------------|-------------|--------------------|---------------|------------------------|---------|-------------|
| | 8 bits | B2h | 02h | 64 bits | 16 bits | 8 bits | 16 bits | |

1. Gray means that the field is optional.

Request parameters:

- (optional) UID
- Sector number
- Sector security status (refer to [Table 77](#))

Table 77. Sector security status

| b ₇ | b ₆ | b ₅ | b ₄ | b ₃ | b ₂ | b ₁ | b ₀ |
|----------------|----------------|----------------|-----------------------|----------------|------------------------------|----------------|----------------|
| 0 | 0 | 0 | password control bits | | Read / Write protection bits | | 1 |

Table 78. Lock-sector Password response format when Error_flag is NOT set

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

Response parameter:

- No parameter.

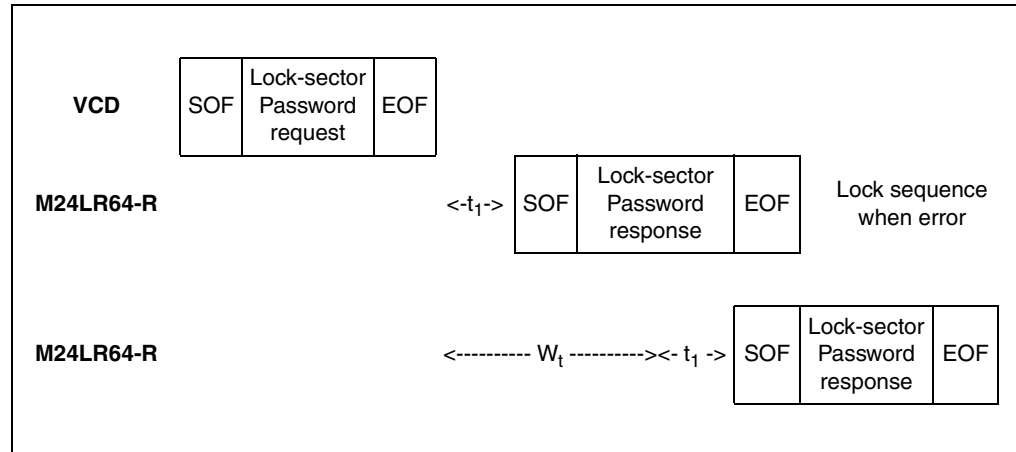
Table 79. Lock-sector Password response format when Error_flag is set

| 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: the command is not recognized, for example: a format error occurred
 - 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
 - 14h: the specified block was not successfully locked

Figure 65. Lock-sector Password frame exchange between VCD and M24LR64-R



26.16 Present-sector Password

On receiving the Present-sector Password command, the M24LR64-R compares the requested password with the data contained in the request and reports whether the operation has been successful in the response. The Option_flag is supported. During the comparison cycle equal to W_t , there should be no modulation (neither 100% nor 10%) otherwise, the M24LR64-R the Password value may not be correctly compared. The W_t time is equal to $t_{1nom} + 18 \times 302 \mu s$. After a successful command, the access to all the memory blocks linked to the password is changed as described in [Section 4.1: M24LR64-R RF block security](#).

Table 80. Present-sector Password request format

| Request SOF | Request _flags | Present-sector Password | IC Mfg code | UID ⁽¹⁾ | Password number | Data | CRC16 | Request EOF |
|-------------|----------------|-------------------------|-------------|--------------------|-----------------|---------|---------|-------------|
| | 8 bits | B3h | 02h | 64 bits | 8 bits | 32 bits | 16 bits | |

1. Gray means that the field is optional.

Request parameter:

- UID (optional)
- Password Number (0x01 = Pswd1, 0x02 = Pswd2, 0x03 = Pswd3, other = Error)
- Data

Table 81. Present-sector 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 send back after the write cycle.

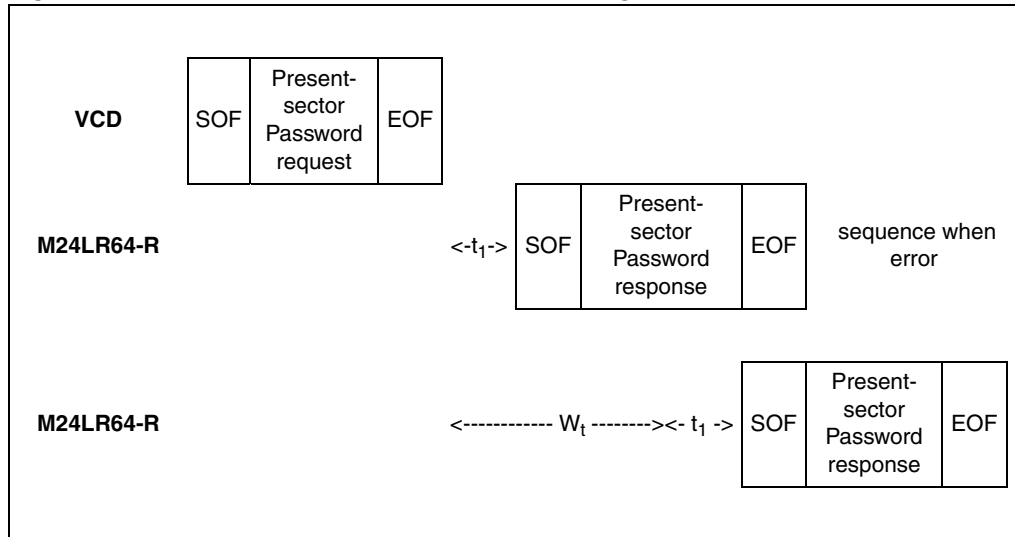
Table 82. Present-sector Password response format when Error_flag is set

| 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: the command is not recognized, for example: a format error occurred
 - 03h: the option is not supported
 - 0Fh: error with no information given
 - 10h: the specified block is not available

Figure 66. Present-sector Password frame exchange between VCD and M24LR64-R



26.17 Fast Read Single Block

On receiving the Fast Read Single Block command, the M24LR64-R reads the requested block and sends back its 32-bit value in the response. The Option_flag is supported. The data rate of the response is multiplied by 2.

The Protocol_extention_flag should be set to 1 for the M24LR64-R to operate correctly. If the Protocol_extention_flag is at 0, the M24LR64-R answers with an error code.

Table 83. Fast Read Single Block request format

| Request SOF | Request flags | Fast Read Single Block | ICMfg code | UID ⁽¹⁾ | Block number | CRC16 | Request EOF |
|-------------|---------------|------------------------|------------|--------------------|--------------|---------|-------------|
| | 8 bits | C0h | 02h | 64 bits | 16 bits | 16 bits | |

1. Gray means that the field is optional.

Request parameters:

- Option_flag
- UID (optional)
- Block number

Table 84. Fast Read Single Block response format when Error_flag is NOT set

| Response SOF | Response flags | Sector security status ⁽¹⁾ | Data | CRC16 | Response EOF |
|--------------|----------------|---------------------------------------|---------|---------|--------------|
| | 8 bits | 8 bits | 32 bits | 16 bits | |

1. Gray means that the field is optional.

Response parameters:

- Sector security status if Option_flag is set (see [Table 85](#))
- 4 bytes of block data

Table 85. Sector security status

| b ₇ | b ₆ | b ₅ | b ₄ | b ₃ | b ₂ | b ₁ | b ₀ |
|------------------------------------|----------------|----------------|-----------------------|----------------|------------------------------|----------------|--|
| Reserved for future used. All at 0 | | | password control bits | | Read / Write protection bits | | 0: Current sector not locked 1: Current sector locked |

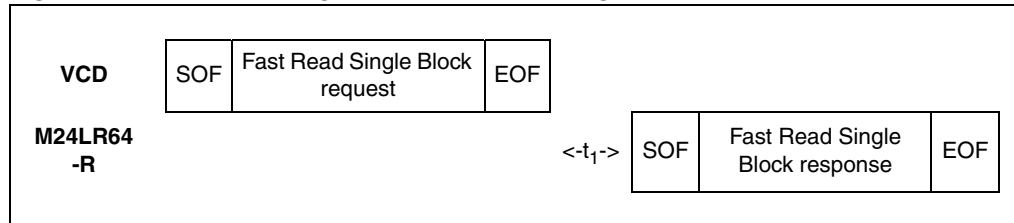
Table 86. Fast Read Single Block response format when Error_flag is set

| 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: the command is not recognized, for example: a format error occurred
 - 03h: the 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 67. Fast Read Single Block frame exchange between VCD and M24LR64-R



26.18 Fast Inventory Initiated

Before receiving the Fast Inventory Initiated command, the M24LR64-R must have received an Initiate or a Fast Initiate command in order to set the Initiate_flag. If not, the M24LR64-R does not answer to the Fast Inventory Initiated command.

On receiving the Fast Inventory Initiated request, the M24LR64-R runs the anticollision sequence. The Inventory_flag must be set to 1. The meaning of flags 5 to 8 is shown in [Table 26: Request flags 5 to 8 when Bit 3 = 1](#). The data rate of the response is multiplied by 2.

The request contains:

- the flags,
- the Inventory command code
- the AFI if the AFI flag is set
- the mask length
- the mask value
- the CRC

The M24LR64-R does not generate any answer in case of error.

Table 87. Fast Inventory Initiated request format

| Request SOF | Request _flags | Fast Inventory Initiated | IC Mfg code | Optional AFI | Mask length | Mask value | CRC16 | Request EOF |
|-------------|----------------|--------------------------|-------------|--------------|-------------|-------------|---------|-------------|
| | 8 bits | C1h | 02h | 8 bits | 8 bits | 0 - 64 bits | 16 bits | |

The Response contains:

- the flags
- the Unique ID

Table 88. Fast Inventory Initiated 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 M24LR64-R response, it waits 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:

$$t_{3\min} = 4384/f_C (323.3\mu\text{s}) + t_{\text{SOF}}$$
- If the VCD sends a 10% modulated EOF, the minimum value of t_3 is:

$$t_{3\min} = 4384/f_C (323.3\mu\text{s}) + t_{\text{NRT}}$$

where:

- t_{SOF} is the time required by the M24LR64-R to transmit an SOF to the VCD
- t_{NRT} is the nominal response time of the M24LR64-R

t_{NRT} and t_{SOF} are dependent on the M24LR64-R-to-VCD data rate and subcarrier modulation mode.

26.19 Fast Initiate

On receiving the Fast Initiate command, the M24LR64-R will set the internal `Initiate_flag` and send back a response only if it is in the Ready state. The command has to be issued in the Non Addressed mode only (`Select_flag` is reset to 0 and `Address_flag` is reset to 0). If an error occurs, the M24LR64-R does not generate any answer. The `Initiate_flag` is reset after a power off of the M24LR64-R. The data rate of the response is multiplied by 2.

The request contains:

- No data

Table 89. Fast Initiate request format

| Request SOF | Request_flags | Fast Initiate | IC Mfg Code | CRC16 | Request EOF |
|-------------|---------------|---------------|-------------|---------|-------------|
| | 8 bits | C2h | 02h | 16 bits | |

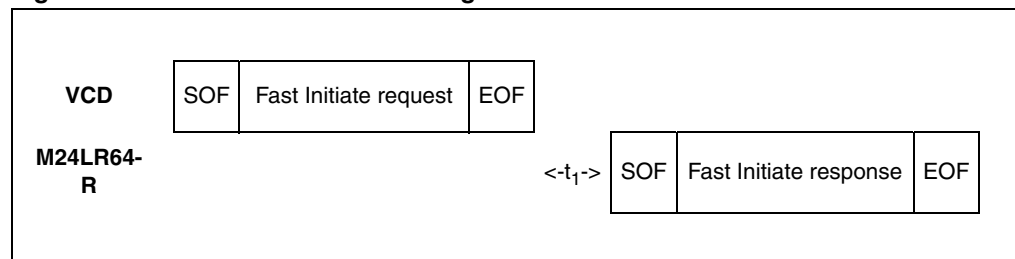
The response contains:

- the flags
- the Unique ID

Table 90. Fast Initiate response format

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

Figure 68. Fast Initiate frame exchange between VCD and M24LR64-R



26.20 Fast Read Multiple Block

On receiving the Fast Read Multiple Block command, the M24LR64-R reads the selected blocks and sends back their value in multiples of 32 bits in the response. The blocks are numbered from '00h to '7FFh' 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, 7 blocks are read. The maximum number of blocks is fixed to 32 assuming that they are all located in the same sector. If the number of blocks overlaps sectors, the M24LR64-R returns an error code.

The Protocol_extention_flag should be set to 1 for the M24LR64-R to operate correctly. If the Protocol_extention_flag is at 0, the M24LR64-R answers with an error code.

The Option_flag is supported. The data rate of the response is multiplied by 2.

Table 91. 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 | 16 bits | 8 bits | 16 bits | |

1. Gray means that the field is optional.

Request parameters:

- Option_flag
- UID (Optional)
- First block number
- Number of blocks

Table 92. Fast Read Multiple Block response format when Error_flag is NOT set

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

1. Gray means that the field is optional.

2. Repeated as needed.

Response parameters:

- Sector security status if Option_flag is set (see [Table 93: Sector security status if Option_flag is set](#))
- N block of data

Table 93. Sector security status if Option_flag is set

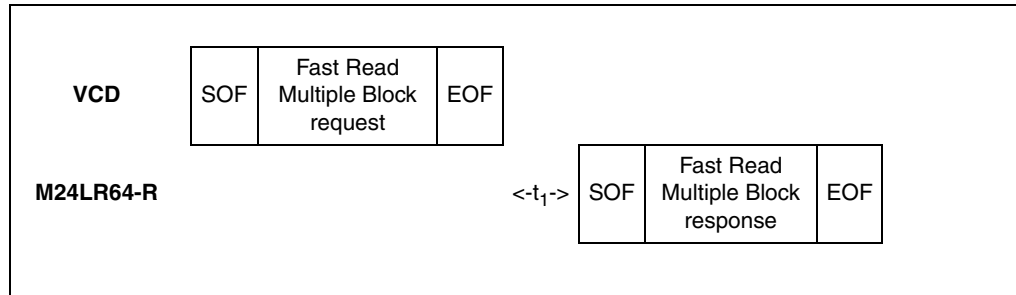
| b ₇ | b ₆ | b ₅ | b ₄ | b ₃ | b ₂ | b ₁ | b ₀ |
|--------------------------------------|----------------|-----------------------|----------------|------------------------------|----------------|--|----------------|
| Reserved for future use. All at 0 | | password control bits | | Read / Write protection bits | | 0: Current sector not locked 1: Current sector locked | |

Table 94. 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:
 - 0Fh: other error
 - 10h: block address not available

Figure 69. Fast Read Multiple Block frame exchange between VCD and M24LR64-R

26.21 Inventory Initiated

Before receiving the Inventory Initiated command, the M24LR64-R must have received an Initiate or a Fast Initiate command in order to set the Initiate_ flag. If not, the M24LR64-R does not answer to the Inventory Initiated command.

On receiving the Inventory Initiated request, the M24LR64-R runs the anticollision sequence. The Inventory_ flag must be set to 1. The meaning of flags 5 to 8 is given in [Table 26: Request flags 5 to 8 when Bit 3 = 1](#).

The request contains:

- the flags,
- the Inventory Command code
- the AFI if the AFI flag is set
- the mask length
- the mask value
- the CRC

The M24LR64-R does not generate any answer in case of error.

Table 95. Inventory Initiated request format

| Request SOF | Request _flags | Inventory Initiated | IC Mfg code | Optional AFI | Mask length | Mask value | CRC16 | Request EOF |
|-------------|----------------|---------------------|-------------|--------------|-------------|-------------|---------|-------------|
| | 8 bits | D1h | 02h | 8 bits | 8 bits | 0 - 64 bits | 16 bits | |

The response contains:

- the flags
- the Unique ID

Table 96. Inventory Initiated 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 M24LR64-R response, it waits 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:
 $t_{3min} = 4384/f_C (323.3\mu s) + t_{SOF}$
- If the VCD sends a 10% modulated EOF, the minimum value of t_3 is:
 $t_{3min} = 4384/f_C (323.3\mu s) + t_{NRT}$

where:

- t_{SOF} is the time required by the M24LR64-R to transmit an SOF to the VCD
- t_{NRT} is the nominal response time of the M24LR64-R

t_{NRT} and t_{SOF} are dependent on the M24LR64-R-to-VCD data rate and subcarrier modulation mode.

26.22 Initiate

On receiving the Initiate command, the M24LR64-R will set the internal Initiate_flag and send back a response only if it is in the ready state. The command has to be issued in the Non Addressed mode only (Select_flag is reset to 0 and Address_flag is reset to 0). If an error occurs, the M24LR64-R does not generate any answer. The Initiate_flag is reset after a power off of the M24LR64-R.

The request contains:

- No data

Table 97. Initiate request format

| Request SOF | Request_flags | Initiate | IC Mfg code | CRC16 | Request EOF |
|-------------|---------------|----------|-------------|---------|-------------|
| | 8 bits | D2h | 02h | 16 bits | |

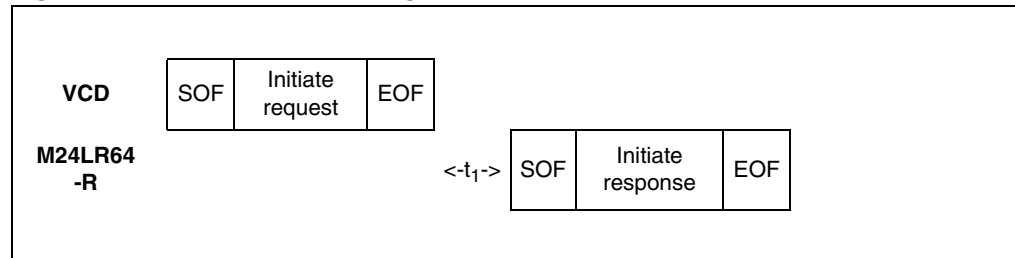
The response contains:

- the flags
- the Unique ID

Table 98. Initiate Initiated response format

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

Figure 70. Initiate frame exchange between VCD and M24LR64-R



27 Maximum rating

Stressing the device above the rating listed in the absolute maximum ratings table 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 device reliability. Refer also to the STMicroelectronics SURE Program and other relevant quality documents.

Table 99. Absolute maximum ratings

| Symbol | Parameter | | Min. | Max. | Unit |
|--------------------------------------|---|--------------------------------|----------------------------|------|--------|
| T_A | Ambient operating temperature | | -40 | 85 | °C |
| T_{STG} , h_{STG} , t_{STG} | Storage conditions | Wafer | 15 | 25 | °C |
| | | | | 23 | months |
| | | | kept in its antistatic bag | | |
| T_{STG} | Storage temperature | UFDFPN8 (MLP8), SO8, TSSOP8 | -65 | 150 | °C |
| T_{LEAD} | Lead temperature during soldering | UFDFPN8 (MLP8), SO8, TSSOP8 | see note (1) | | °C |
| V_{IO} | I_2C input or output range | | -0.50 | 6.0 | V |
| V_{CC} | I_2C supply voltage | | -0.50 | 6.0 | V |
| I_{CC} | RF supply current on AC0 / AC1 | | -20 | 20 | mA |
| V_{MAX} | RF input voltage on AC0 / AC1 | | -7 | 7 | V |
| V_{ESD} | Electrostatic discharge voltage (human body model) ⁽²⁾ | AC0, AC1 | -800 | 800 | V |
| | | Other pads | -3000 | 3000 | |
| | Electrostatic discharge voltage (Machine model) | | -100 | 100 | |

1. 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.

2. AEC-Q100-002 (compliant with JEDEC Std JESD22-A114A, C1 = 100 pF, R1 = 1500 Ω , R2 = 500 Ω)

28 I²C DC and AC parameters

This section summarizes the operating and measurement conditions, and the DC and AC characteristics of the device in I²C 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 100. I²C 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 101. AC test measurement conditions

| Symbol | Parameter | Min. | Max. | Unit |
|----------------|--|--|------|------|
| C _L | Load capacitance | 100 | | pF |
| | Input rise and fall times | | 50 | ns |
| | Input levels | 0.2V _{CC} to 0.8V _{CC} | | V |
| | Input and output timing reference levels | 0.3V _{CC} to 0.7V _{CC} | | V |

Figure 71. AC test measurement I/O waveform

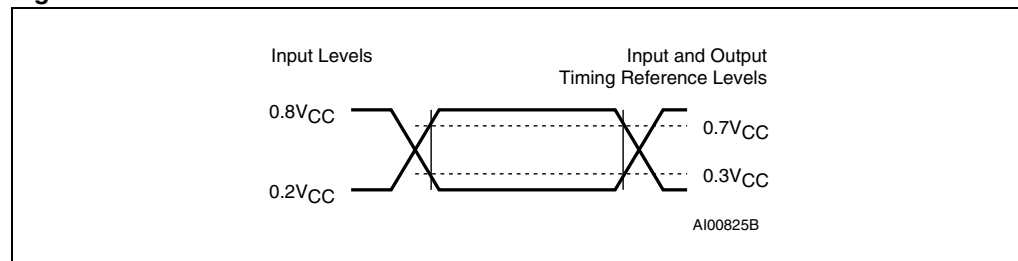


Table 102. Input parameters

| 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 103. I²C DC characteristics

| Symbol | Parameter | Test condition | Min. | Max. | Unit |
|------------------|--|---|---------------------|---------------------|------|
| I _{LI} | Input leakage current (SCL, SDA, E1, E0) | V _{IN} = V _{SS} or V _{CC} device in Standby mode | | ± 2 | μA |
| I _{LO} | Output leakage current | SDA in Hi-Z, external voltage applied on SDA: V _{SS} or V _{CC} | | ± 2 | μA |
| I _{CC} | Supply current (Read) | V _{CC} = 1.8 V, f _c = 400 kHz (rise/fall time < 50 ns) | | 100 | μA |
| | | V _{CC} = 2.5 V, f _c = 400 kHz (rise/fall time < 50 ns) | | 200 | |
| | | V _{CC} = 5.5 V, f _c = 400 kHz (rise/fall time < 50 ns) | | 400 | |
| I _{CC0} | Supply current (Write) | During t _W , 1.8 V < V _{CC} < 5.5 V | | 600 ⁽¹⁾ | μA |
| I _{CC1} | Standby supply current | V _{IN} = V _{SS} or V _{CC} , 1.8 V < V _{CC} < 3.6 V | | 30 | μA |
| | | V _{IN} = V _{SS} or V _{CC} , 3.6 V < V _{CC} < 5.5 V | | 40 | μA |
| V _{IL} | Input low voltage (SDA, SCL) | V _{CC} ≥ 2.5 V | -0.45 | 0.3V _{CC} | V |
| | | 1.8 V ≤ V _{CC} < 2.5 V | -0.45 | 0.25V _{CC} | |
| V _{IH} | Input high voltage (SDA, SCL) | V _{CC} ≥ 2.5 V | 0.7V _{CC} | V _{CC} +1 | V |
| | | 1.8 V ≤ V _{CC} < 2.5 V | 0.75V _{CC} | V _{CC} +1 | |
| V _{OL} | Output low voltage | I _{OL} = 2.1 mA, V _{CC} = 1.8 V or I _{OL} = 3 mA, V _{CC} = 5.5 V | | 0.4 | V |

1. Characterized value, not tested in production.

Table 104. I²C AC characteristics

| Test conditions specified in Table 100 | | | | | |
|--|---------------------|--|------|------|------|
| Symbol | Alt. | Parameter | Min. | Max. | Unit |
| f _C | f _{SCL} | Clock frequency | | 400 | kHz |
| t _{CHCL} | t _{HIGH} | Clock pulse width high | 600 | | ns |
| t _{CLCH} | t _{LOW} | Clock pulse width low | 1300 | | ns |
| t _{XH1XH2} ⁽¹⁾ | t _R | Input signal rise time | 20 | 300 | ns |
| t _{XL1XL2} ⁽¹⁾ | t _F | Input signal fall time | 20 | 300 | ns |
| t _{DL1DL2} | t _F | SDA (out) fall time | 20 | 100 | ns |
| t _{DXCX} | t _{SU:DAT} | Data in set up time | 100 | | ns |
| t _{CLDX} | t _{HD:DAT} | Data in hold time | 0 | | ns |
| t _{CLQX} | t _{DH} | Data out hold time | 100 | | ns |
| t _{CLQV} ⁽²⁾⁽³⁾ | t _{AA} | Clock low to next data valid (access time) | 100 | 900 | ns |
| t _{CHDX} ⁽⁴⁾ | t _{SU:STA} | Start condition set up time | 600 | | ns |
| t _{DLCL} | t _{HD:STA} | Start condition hold time | 600 | | ns |
| t _{CHDH} | t _{SU:STO} | Stop condition set up time | 600 | | ns |
| t _{DHDL} | t _{BUF} | Time between Stop condition and next Start condition | 1300 | | ns |
| t _W | | I ² C write time | | 5 | ms |

1. Values recommended by the I²C-bus Fast-Mode specification.
2. To avoid spurious Start and Stop conditions, a minimum delay is placed between SCL=1 and the falling or rising edge of SDA.
3. 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} × C_{bus} time constant is less than 500 ns (as specified in [Figure 4](#)).
4. For a reStart condition, or following a write cycle.

29 RF DC and AC 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 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 105. RF AC 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 | $T_A = 0\text{ }^{\circ}\text{C}$ to $50\text{ }^{\circ}\text{C}$ | 150 | | 5000 | mA/m |
| H_Extended | Operating field in extended temperature range | $T_A = -40\text{ }^{\circ}\text{C}$ to $85\text{ }^{\circ}\text{C}$ | 150 | | 3500 | mA/m |
| MI_CARRIER | 10% carrier modulation index | $MI=(A-B)/(A+B)$ | 10 | | 30 | % |
| t_{RFR} , t_{RFF} | 10% rise and fall time | | 0.5 | | 3.0 | μs |
| t_{RFSBL} | 10% minimum pulse width for bit | | 7.1 | | 9.44 | μs |
| MI_CARRIER | 100% carrier modulation index | $MI=(A-B)/(A+B)$ | 95 | | 100 | % |
| t_{RFR} , t_{RFF} | 100% rise and fall time | | 0.5 | | 3.5 | μs |
| t_{RFSBL} | 100% minimum pulse width for bit | | 7.1 | | 9.44 | μs |
| t_{JIT} | Bit pulse jitter | | -2 | | +2 | μs |
| $t_{MIN\ CD}$ | Minimum time from carrier generation to first data | From H-field min | | 0.1 | 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 M24LR64-R response | $4224/F_S$ | 318.6 | 320.9 | 323.3 | μs |
| t_2 | Time between commands | $4224/F_S$ | 309 | 311.5 | 314 | μs |
| W_t | RF write time (including internal Verify) | | | 5.75 | | ms |

- $T_A = -40$ to $85\text{ }^{\circ}\text{C}$.
- All timing measurements were performed between $0\text{ }^{\circ}\text{C}$ and $50\text{ }^{\circ}\text{C}$ on a reference antenna with the following characteristics:
 External size: 75 mm x 48 mm
 Number of turns: 6
 Width of conductor: 1 mm
 Space between 2 conductors: 0.4 mm
 Value of the tuning capacitor: 28.5 pF (M24LR64-R-W4)
 Value of the coil: 4.3 μH
 Tuning frequency: 13.8 MHz.

Table 106. RF DC characteristics⁽¹⁾

| Symbol | Parameter | Test conditions | Min. | Typ. | Max. | Unit |
|-----------|--------------------------------|------------------------|-------------------------|------|------|---------------|
| V_{CC} | Limited voltage | | | | 2.0 | V |
| V_{RET} | Retromodulated induced voltage | ISO10373-7 | 10 | | | mV |
| I_{CC} | Supply current | Read | $V_{CC} = 2.0\text{ V}$ | | 50 | μA |
| | | Write | $V_{CC} = 2.0\text{ V}$ | | 150 | μA |
| C_{TUN} | Internal tuning capacitor | $f = 13.56\text{ MHz}$ | | 28.5 | | pF |

1. $T_A = -40$ to $85\text{ }^\circ\text{C}$.

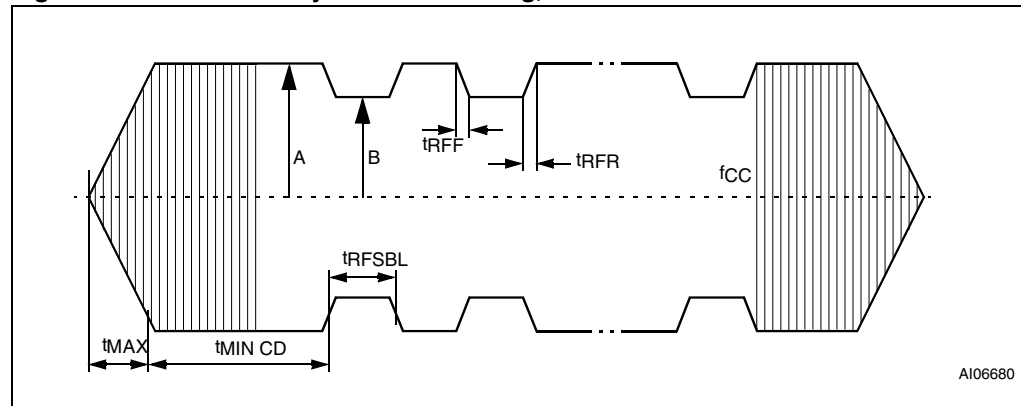
Table 107. Operating conditions

| Symbol | Parameter | Min. | Max. | Unit |
|--------|-------------------------------|------|------|------------------|
| T_A | Ambient operating temperature | -40 | 85 | $^\circ\text{C}$ |

Figure 73 shows an ASK modulated signal, from the VCD to the M24LR64-R. The test condition for the AC/DC parameters are:

- Close coupling condition with tester antenna (1mm)
- M24LR64-R performance measured at the tag antenna

Figure 73. M24LR64-R synchronous timing, transmit and receive

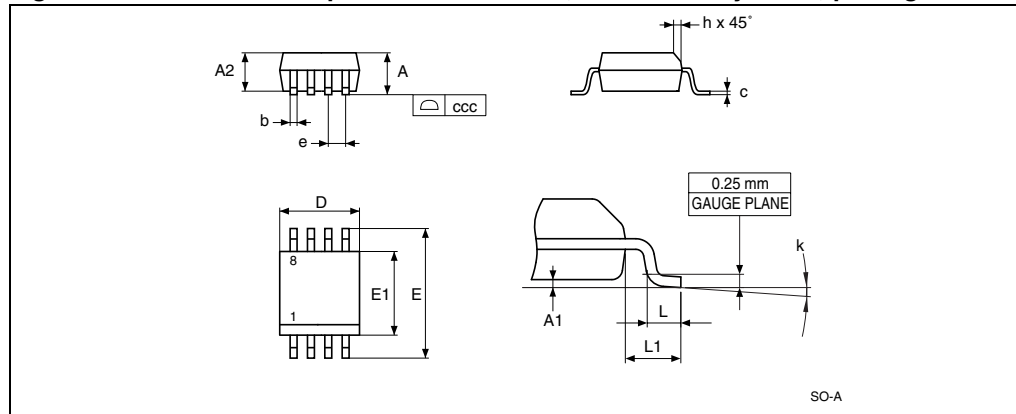


AI06680

30 Package mechanical data

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.

Figure 74. SO8N – 8-lead plastic small outline, 150 mils body width, package outline



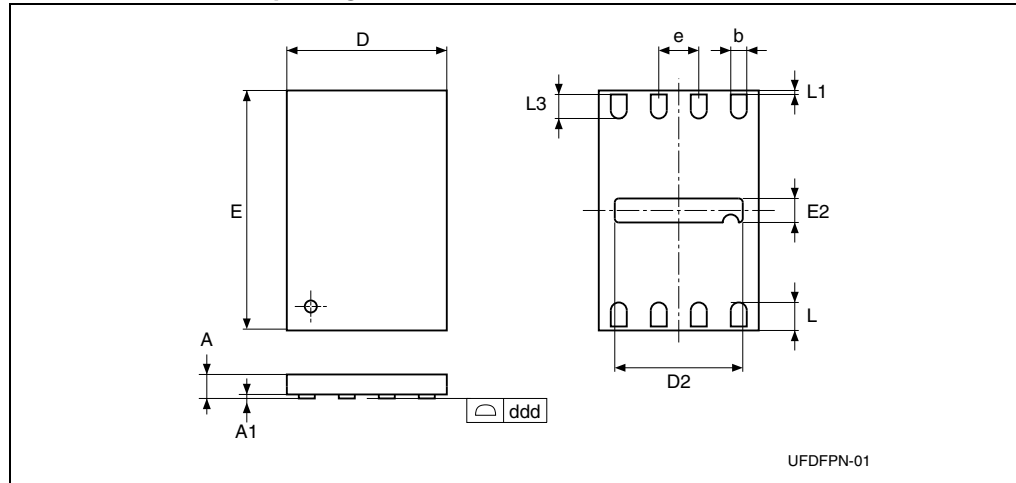
1. Drawing is not to scale.

Table 108. SO8N – 8-lead plastic small outline, 150 mils body width, package data

| Symbol | millimeters | | | inches ⁽¹⁾ | | |
|--------|-------------|------|------|-----------------------|--------|--------|
| | Typ | Min | Max | Typ | Min | Max |
| A | | | 1.75 | | | 0.0689 |
| A1 | | 0.10 | 0.25 | | 0.0039 | 0.0098 |
| A2 | | 1.25 | | | 0.0492 | |
| b | | 0.28 | 0.48 | | 0.0110 | 0.0189 |
| c | | 0.17 | 0.23 | | 0.0067 | 0.0091 |
| ccc | | | 0.10 | | | 0.0039 |
| D | 4.90 | 4.80 | 5.00 | 0.1929 | 0.1890 | 0.1969 |
| E | 6.00 | 5.80 | 6.20 | 0.2362 | 0.2283 | 0.2441 |
| E1 | 3.90 | 3.80 | 4.00 | 0.1535 | 0.1496 | 0.1575 |
| e | 1.27 | – | – | 0.0500 | – | – |
| h | | 0.25 | 0.50 | | | |
| k | | 0° | 8° | | 0° | 8° |
| L | | 0.40 | 1.27 | | 0.0157 | 0.0500 |
| L1 | 1.04 | | | 0.0410 | | |

1. Values in inches are converted from mm and rounded to 4 decimal digits.

**Figure 75. UFDFPN8 (MLP8) – Ultra thin fine pitch dual flat package no lead
2 x 3 mm, package outline**



1. Drawing is not to scale.

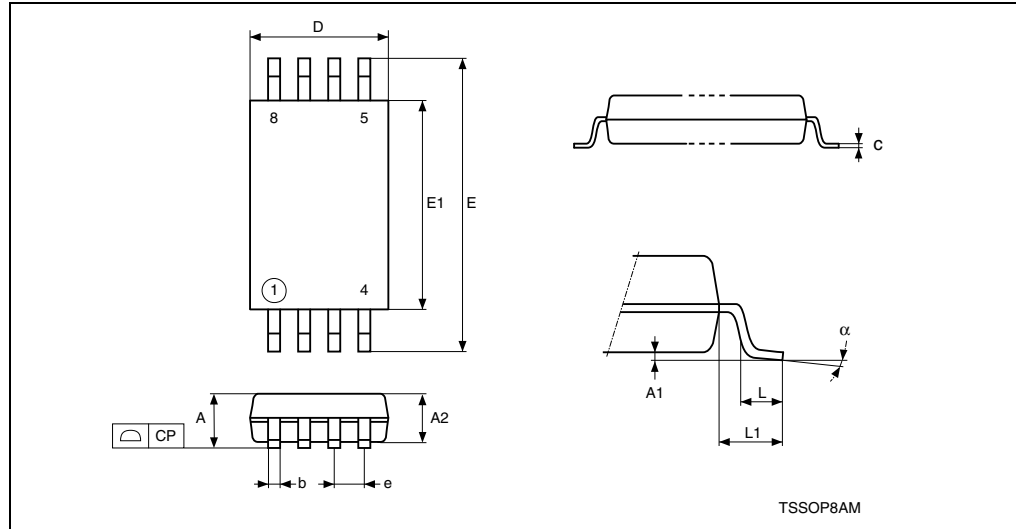
**Table 109. UFDFPN8 (MLP8) – Ultra thin fine pitch dual flat package no lead
2 x 3 mm, package mechanical data**

| Symbol | millimeters | | | inches ⁽¹⁾ | | |
|--------------------|-------------|------|------|-----------------------|--------|--------|
| | Typ | Min | Max | Typ | Min | Max |
| A | 0.55 | 0.45 | 0.6 | 0.0217 | 0.0177 | 0.0236 |
| A1 | 0.02 | 0 | 0.05 | 0.0008 | 0 | 0.002 |
| b | 0.25 | 0.2 | 0.3 | 0.0098 | 0.0079 | 0.0118 |
| D | 2 | 1.9 | 2.1 | 0.0787 | 0.0748 | 0.0827 |
| D2 | 1.6 | 1.5 | 1.7 | 0.063 | 0.0591 | 0.0669 |
| E | 3 | 2.9 | 3.1 | 0.1181 | 0.1142 | 0.122 |
| E2 | 0.2 | 0.1 | 0.3 | 0.0079 | 0.0039 | 0.0118 |
| e | 0.5 | - | - | 0.0197 | - | - |
| L | 0.45 | 0.4 | 0.5 | 0.0177 | 0.0157 | 0.0197 |
| L1 | | | 0.15 | | | 0.0059 |
| L3 | | 0.3 | | | 0.0118 | |
| ddd ⁽²⁾ | | 0.08 | | | 0.0031 | |

1. Values in inches are converted from mm and rounded to 4 decimal digits.

2. Applied for exposed die paddle and terminals. Exclude embedding part of exposed die paddle from measuring.

Figure 76. TSSOP8 – 8-lead thin shrink small outline, package outline



1. Drawing is not to scale.

Table 110. TSSOP8 – 8-lead thin shrink small outline, package mechanical data

| Symbol | millimeters | | | inches ⁽¹⁾ | | |
|--------|-------------|------|------|-----------------------|--------|--------|
| | Typ | Min | Max | Typ | Min | Max |
| A | | | 1.2 | | | 0.0472 |
| A1 | | 0.05 | 0.15 | | 0.002 | 0.0059 |
| A2 | 1 | 0.8 | 1.05 | 0.0394 | 0.0315 | 0.0413 |
| b | | 0.19 | 0.3 | | 0.0075 | 0.0118 |
| c | | 0.09 | 0.2 | | 0.0035 | 0.0079 |
| CP | | | 0.1 | | | 0.0039 |
| D | 3 | 2.9 | 3.1 | 0.1181 | 0.1142 | 0.122 |
| e | 0.65 | - | - | 0.0256 | - | - |
| E | 6.4 | 6.2 | 6.6 | 0.252 | 0.2441 | 0.2598 |
| E1 | 4.4 | 4.3 | 4.5 | 0.1732 | 0.1693 | 0.1772 |
| L | 0.6 | 0.45 | 0.75 | 0.0236 | 0.0177 | 0.0295 |
| L1 | 1 | | | 0.0394 | | |
| a | | 0° | 8° | | 0° | 8° |
| N | 8 | | | 8 | | |

1. Values in inches are converted from mm and rounded to 4 decimal digits.

31 Part numbering

Table 111. Ordering information scheme

| | | | | | | |
|--------------------------|---|---|----|---|---|----|
| Example: | M24LR64-R | - | MN | 6 | T | /2 |
| Device type | M24LR64 = Dual-access EERPOM | | | | | |
| Operating voltage | R = $V_{CC} = 1.8 \text{ V to } 5.5 \text{ V}$ | | | | | |
| Package | MN = SO8N (150 mils width) MB = UFDFPN8 (MLP8) DW = TSSOP8 | | | | | |
| Device grade | 6 = industrial: device tested with standard test flow over -40 to $85 \text{ }^\circ\text{C}$ | | | | | |
| Option | T = Tape and reel packing | | | | | |
| Capacitance | /2 = 28.5 pF | | | | | |

For a list of available options (speed, package, etc.) or for further information on any aspect of this device, please contact your nearest ST sales office.

Appendix A Anticollision algorithm (informative)

The following pseudocode describes how anticollision could be implemented on the VCD, using recursivity.

A.1 Algorithm for pulsed slots

```

function push (mask, address); pushes on private stack
function pop (mask, address); pops from private stack
function pulse_next_pause; generates a power pulse
function store(M24LR64-R_UID); stores M24LR64-R_UID

function poll_loop (sub_address_size as integer)
  pop (mask, address)
  mask = address & mask; generates new mask
    ; send the request
  mode = anticollision
  send_Request (Request_cmd, mode, mask length, mask value)
  for sub_address = 0 to (2^sub_address_size - 1)
    pulse_next_pause
    if no_collision_is_detected ; M24LR64-R is inventoried
      then
        store (M24LR64-R_UID)
      else ; remember a collision was detected
        push(mask,address)
      endif
    next sub_address

    if stack_not_empty ; if some collisions have been detected and
      then ; not yet processed, the function calls itself
        poll_loop (sub_address_size); recursively to process the
last stored collision
      endif
  end poll_loop

main_cycle:
  mask = null
  address = null
  push (mask, address)
  poll_loop(sub_address_size)
end_main_cycle

```

Appendix B CRC (informative)

B.1 CRC error detection method

The cyclic redundancy check (CRC) is calculated on all data contained in a message, from the start of the flags through to the end of Data. The CRC is used from VCD to M24LR64-R and from M24LR64-R to VCD.

Table 112. CRC definition

| CRC definition | | | | | |
|----------------|---------|-------------------------------------|-----------|--------|---------|
| CRC type | Length | Polynomial | Direction | Preset | Residue |
| ISO/IEC 13239 | 16 bits | $X^{16} + X^{12} + X^5 + 1 = 8408h$ | Backward | FFFFh | F0B8h |

To add extra protection against shifting errors, a further transformation on the calculated CRC is made. The One's Complement of the calculated CRC is the value attached to the message for transmission.

To check received messages the 2 CRC bytes are often also included in the re-calculation, for ease of use. In this case, the expected value for the generated CRC is the residue F0B8h.

B.2 CRC calculation example

This example in C language illustrates one method of calculating the CRC on a given set of bytes comprising a message.

C-example to calculate or check the CRC16 according to ISO/IEC 13239

```
#define POLYNOMIAL0x8408// x^16 + x^12 + x^5 + 1
#define PRESET_VALUE0xFFFF
#define CHECK_VALUE0xF0B8

#define NUMBER_OF_BYTES4// Example: 4 data bytes
#define CALC_CRC1
#define CHECK_CRC0

void main()
{
    unsigned int current_crc_value;
    unsigned char array_of_databytes[NUMBER_OF_BYTES + 2] = {1, 2, 3,
4, 0x91, 0x39};
    int number_of_databytes = NUMBER_OF_BYTES;
    int calculate_or_check_crc;
    int i, j;
    calculate_or_check_crc = CALC_CRC;
// calculate_or_check_crc = CHECK_CRC;// This could be an other
example
    if (calculate_or_check_crc == CALC_CRC)
    {
```

```

        number_of_databytes = NUMBER_OF_BYTES;
    }
    else    // check CRC
    {
        number_of_databytes = NUMBER_OF_BYTES + 2;
    }

    current_crc_value = PRESET_VALUE;

    for (i = 0; i < number_of_databytes; i++)
    {
        current_crc_value = current_crc_value ^ ((unsigned
int)array_of_databytes[i]);

        for (j = 0; j < 8; j++)
        {
            if (current_crc_value & 0x0001)
            {
                current_crc_value = (current_crc_value >> 1) ^
POLYNOMIAL;
            }
            else
            {
                current_crc_value = (current_crc_value >> 1);
            }
        }
    }

    if (calculate_or_check_crc == CALC_CRC)
    {
        current_crc_value = ~current_crc_value;

        printf ("Generated CRC is 0x%04X\n", current_crc_value);

        // current_crc_value is now ready to be appended to the data
stream
        // (first LSByte, then MSByte)
    }
    else    // check CRC
    {
        if (current_crc_value == CHECK_VALUE)
        {
            printf ("Checked CRC is ok (0x%04X)\n",
current_crc_value);
        }
        else
        {
            printf ("Checked CRC is NOT ok (0x%04X)\n",
current_crc_value);
        }
    }
}
}

```

Appendix C Application family identifier (AFI) (informative)

The AFI (application family identifier) represents the type of application targeted by the VCD and is used to extract from all the M24LR64-R present only the M24LR64-R meeting the required application criteria.

It is programmed by the M24LR64-R issuer (the purchaser of the M24LR64-R). Once locked, it cannot be modified.

The most significant nibble of the AFI is used to code one specific or all application families, as defined in [Table 113](#).

The least significant nibble of the AFI is used to code one specific or all application subfamilies. Subfamily codes different from 0 are proprietary.

Table 113. AFI coding⁽¹⁾

| AFI Most significant nibble | AFI Least significant nibble | Meaning VICCs respond from | Examples / Note |
|-----------------------------|------------------------------|------------------------------------|--------------------------------|
| '0' | '0' | All families and subfamilies | No applicative preselection |
| 'X' | '0' | 'All subfamilies of family X | Wide applicative preselection |
| 'X' | 'Y' | Only the Yth subfamily of family X | |
| '0' | 'Y' | Proprietary subfamily Y only | |
| '1' | "0', 'Y' | Transport | Mass transit, Bus, Airline,... |
| '2' | "0', 'Y' | Financial | IEP, Banking, Retail,... |
| '3' | "0', 'Y' | Identification | Access Control,... |
| '4' | "0', 'Y' | Telecommunication | Public Telephony, GSM,... |
| '5' | '0', 'Y' | Medical | |
| '6' | "0', 'Y' | Multimedia | Internet services.... |
| '7' | "0', 'Y' | Gaming | |
| 8 | "0', 'Y' | Data Storage | Portable Files,... |
| '9' | "0', 'Y' | Item Management | |
| 'A' | "0', 'Y' | Express Parcels | |
| 'B' | "0', 'Y' | Postal Services | |
| 'C' | "0', 'Y' | Airline Bags | |
| 'D' | "0', 'Y' | RFU | |
| 'E' | "0', 'Y' | RFU | |
| 'F' | '0', 'Y' | RFU | |

1. X = '1' to 'F', Y = '1' to 'F'

Revision history

Table 114. Document revision history

| Date | Revision | Changes |
|-------------|----------|--|
| 26-Feb-2010 | 8 | Previous revisions: design and engineering phase. Initial public release. |

Please Read Carefully:

Information in this document is provided solely in connection with ST products. STMicroelectronics NV and its subsidiaries ("ST") reserve the right to make changes, corrections, modifications or improvements, to this document, and the products and services described herein at any time, without notice.

All ST products are sold pursuant to ST's terms and conditions of sale.

Purchasers are solely responsible for the choice, selection and use of the ST products and services described herein, and ST assumes no liability whatsoever relating to the choice, selection or use of the ST products and services described herein.

No license, express or implied, by estoppel or otherwise, to any intellectual property rights is granted under this document. If any part of this document refers to any third party products or services it shall not be deemed a license grant by ST for the use of such third party products or services, or any intellectual property contained therein or considered as a warranty covering the use in any manner whatsoever of such third party products or services or any intellectual property contained therein.

UNLESS OTHERWISE SET FORTH IN ST'S TERMS AND CONDITIONS OF SALE ST DISCLAIMS ANY EXPRESS OR IMPLIED WARRANTY WITH RESPECT TO THE USE AND/OR SALE OF ST PRODUCTS INCLUDING WITHOUT LIMITATION IMPLIED WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE (AND THEIR EQUIVALENTS UNDER THE LAWS OF ANY JURISDICTION), OR INFRINGEMENT OF ANY PATENT, COPYRIGHT OR OTHER INTELLECTUAL PROPERTY RIGHT.

UNLESS EXPRESSLY APPROVED IN WRITING BY AN AUTHORIZED ST REPRESENTATIVE, ST PRODUCTS ARE NOT RECOMMENDED, AUTHORIZED OR WARRANTED FOR USE IN MILITARY, AIR CRAFT, SPACE, LIFE SAVING, OR LIFE SUSTAINING APPLICATIONS, NOR IN PRODUCTS OR SYSTEMS WHERE FAILURE OR MALFUNCTION MAY RESULT IN PERSONAL INJURY, DEATH, OR SEVERE PROPERTY OR ENVIRONMENTAL DAMAGE. ST PRODUCTS WHICH ARE NOT SPECIFIED AS "AUTOMOTIVE GRADE" MAY ONLY BE USED IN AUTOMOTIVE APPLICATIONS AT USER'S OWN RISK.

Resale of ST products with provisions different from the statements and/or technical features set forth in this document shall immediately void any warranty granted by ST for the ST product or service described herein and shall not create or extend in any manner whatsoever, any liability of ST.

ST and the ST logo are trademarks or registered trademarks of ST in various countries.

Information in this document supersedes and replaces all information previously supplied.

The ST logo is a registered trademark of STMicroelectronics. All other names are the property of their respective owners.

© 2010 STMicroelectronics - All rights reserved

STMicroelectronics group of companies

Australia - Belgium - Brazil - Canada - China - Czech Republic - Finland - France - Germany - Hong Kong - India - Israel - Italy - Japan - Malaysia - Malta - Morocco - Philippines - Singapore - Spain - Sweden - Switzerland - United Kingdom - United States of America

www.st.com