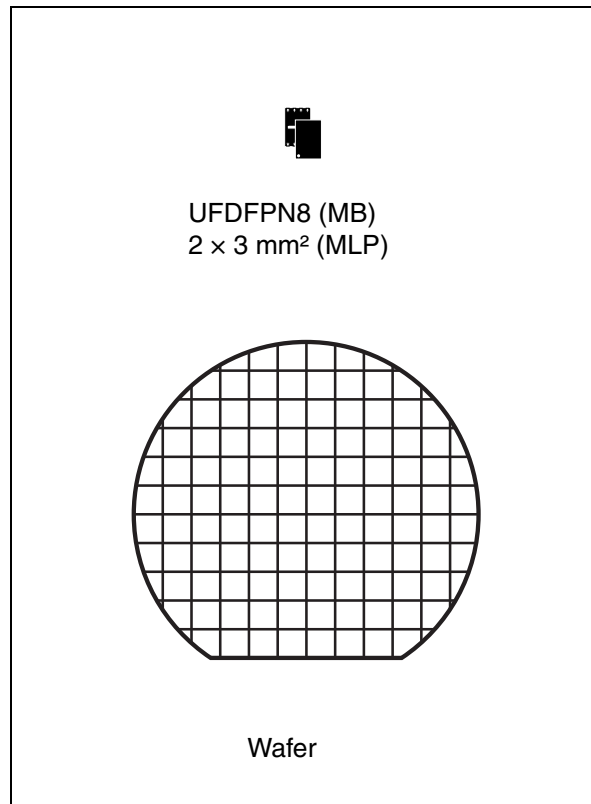


2048-bit EEPROM tag IC at 13.56 MHz, with 64-bit UID and kill code, ISO 15693 and ISO 18000-3 Mode 1 compliant

## Features

- ISO 15693 standard fully compliant
- ISO 18000-3 Mode 1 standard fully compliant
- 13.56 MHz  $\pm 7$  kHz 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 capacitor (21 pF, 23.5 pF, 28.5 pF, 97 pF)
- 1 000 000 Erase/Write cycles (minimum)
- 40 year data retention (minimum)
- 2048 bits EEPROM with Block Lock feature
- 64-bit unique identifier (UID)
- Electrical article surveillance capable (software controlled)
- Kill function
- Read & Write (Block of 32 bits)
- 5 ms programming time
- Packages
  - ECOPACK® (RoHS compliant)



# Contents

|          |                                                      |           |
|----------|------------------------------------------------------|-----------|
| <b>1</b> | <b>Description .....</b>                             | <b>10</b> |
| 1.1      | Memory mapping .....                                 | 11        |
| 1.2      | Commands .....                                       | 12        |
| 1.3      | Initial dialogue for vicinity cards .....            | 13        |
| 1.3.1    | Power transfer .....                                 | 13        |
| 1.3.2    | Frequency .....                                      | 13        |
| 1.3.3    | Operating field .....                                | 13        |
| <b>2</b> | <b>Communication signal from VCD to LRI2K .....</b>  | <b>14</b> |
| <b>3</b> | <b>Data rate and data coding .....</b>               | <b>15</b> |
| 3.1      | Data coding mode: 1 out of 256 .....                 | 15        |
| 3.2      | Data coding mode: 1 out of 4 .....                   | 17        |
| 3.3      | VCD to LRI2K frames .....                            | 18        |
| 3.4      | Start of frame (SOF) .....                           | 18        |
| <b>4</b> | <b>Communications signal from LRI2K to VCD .....</b> | <b>19</b> |
| 4.1      | Load modulation .....                                | 19        |
| 4.2      | Subcarrier .....                                     | 19        |
| 4.3      | Data rates .....                                     | 19        |
| <b>5</b> | <b>Bit representation and coding .....</b>           | <b>20</b> |
| 5.1      | Bit coding using one subcarrier .....                | 20        |
| 5.1.1    | High data rate .....                                 | 20        |
| 5.1.2    | Low data rate .....                                  | 21        |
| 5.2      | Bit coding using two subcarriers .....               | 22        |
| 5.2.1    | High data rate .....                                 | 22        |
| 5.2.2    | Low data rate .....                                  | 22        |
| <b>6</b> | <b>LRI2K to VCD frames .....</b>                     | <b>23</b> |
| 6.1      | SOF when using one subcarrier .....                  | 23        |
| 6.1.1    | High data rate .....                                 | 23        |
| 6.1.2    | Low data rate .....                                  | 24        |

|           |                                                     |           |
|-----------|-----------------------------------------------------|-----------|
| 6.2       | SOF when using two subcarriers .....                | 24        |
| 6.2.1     | High data rate .....                                | 24        |
| 6.2.2     | Low data rate .....                                 | 24        |
| 6.3       | EOF when using one subcarrier .....                 | 25        |
| 6.3.1     | High data rate .....                                | 25        |
| 6.3.2     | Low data rate .....                                 | 25        |
| 6.4       | EOF when using two subcarriers .....                | 26        |
| 6.4.1     | High data rate .....                                | 26        |
| 6.4.2     | Low data rate .....                                 | 26        |
| <b>7</b>  | <b>Unique identifier (UID) .....</b>                | <b>27</b> |
| <b>8</b>  | <b>Application family identifier (AFI) .....</b>    | <b>28</b> |
| <b>9</b>  | <b>Data storage format identifier (DSFID) .....</b> | <b>29</b> |
| 9.1       | CRC .....                                           | 29        |
| <b>10</b> | <b>LRI2K protocol description .....</b>             | <b>30</b> |
| <b>11</b> | <b>LRI2K states .....</b>                           | <b>32</b> |
| 11.1      | Power-off state .....                               | 32        |
| 11.2      | Ready state .....                                   | 32        |
| 11.3      | Quiet state .....                                   | 32        |
| 11.4      | Selected state .....                                | 32        |
| <b>12</b> | <b>Modes .....</b>                                  | <b>34</b> |
| 12.1      | Addressed mode .....                                | 34        |
| 12.2      | Non-Addressed mode (general request) .....          | 34        |
| 12.3      | Select mode .....                                   | 34        |
| <b>13</b> | <b>Request format .....</b>                         | <b>35</b> |
| 13.1      | Request flags .....                                 | 35        |
| <b>14</b> | <b>Response format .....</b>                        | <b>37</b> |
| 14.1      | Response flags .....                                | 37        |
| 14.2      | Response error code .....                           | 38        |

|           |                                                                                               |           |
|-----------|-----------------------------------------------------------------------------------------------|-----------|
| <b>15</b> | <b>Anticollision</b> .....                                                                    | <b>39</b> |
|           | 15.1 Request parameters .....                                                                 | 39        |
| <b>16</b> | <b>Request processing by the LRI2K</b> .....                                                  | <b>41</b> |
| <b>17</b> | <b>Explanation of the possible cases</b> .....                                                | <b>42</b> |
| <b>18</b> | <b>Inventory Initiated command</b> .....                                                      | <b>44</b> |
| <b>19</b> | <b>Timing definition</b> .....                                                                | <b>45</b> |
|           | 19.1 t <sub>1</sub> : LRI2K response delay .....                                              | 45        |
|           | 19.2 t <sub>2</sub> : VCD new request delay .....                                             | 45        |
|           | 19.3 t <sub>3</sub> : VCD new request delay in the absence of a response from the LRI2K ..... | 45        |
| <b>20</b> | <b>Commands codes</b> .....                                                                   | <b>46</b> |
|           | 20.1 Inventory .....                                                                          | 47        |
|           | 20.2 Stay Quiet .....                                                                         | 48        |
|           | 20.3 Read Single Block .....                                                                  | 49        |
|           | 20.4 Write Single Block .....                                                                 | 51        |
|           | 20.5 Lock Block .....                                                                         | 52        |
|           | 20.6 Read Multiple Block .....                                                                | 53        |
|           | 20.7 Select .....                                                                             | 55        |
|           | 20.8 Reset to Ready .....                                                                     | 56        |
|           | 20.9 Write AFI .....                                                                          | 57        |
|           | 20.10 Lock AFI .....                                                                          | 58        |
|           | 20.11 Write DSFID .....                                                                       | 59        |
|           | 20.12 Lock DSFID .....                                                                        | 60        |
|           | 20.13 Get System Info .....                                                                   | 61        |
|           | 20.14 Get Multiple Block Security Status .....                                                | 62        |
|           | 20.15 Kill .....                                                                              | 64        |
|           | 20.16 Write Kill .....                                                                        | 65        |
|           | 20.17 Lock Kill .....                                                                         | 66        |
|           | 20.18 Fast Read Single Block .....                                                            | 68        |
|           | 20.19 Fast Inventory Initiated .....                                                          | 70        |
|           | 20.20 Fast Initiate .....                                                                     | 71        |

|                         |                                                         |           |
|-------------------------|---------------------------------------------------------|-----------|
| 20.21                   | Fast Read Multiple Block .....                          | 72        |
| 20.22                   | Inventory Initiated .....                               | 74        |
| 20.23                   | Initiate .....                                          | 75        |
| <b>21</b>               | <b>Maximum rating .....</b>                             | <b>76</b> |
| <b>22</b>               | <b>DC and AC parameters .....</b>                       | <b>77</b> |
| <b>23</b>               | <b>Package mechanical data .....</b>                    | <b>79</b> |
| <b>24</b>               | <b>Part numbering .....</b>                             | <b>80</b> |
| <b>Appendix A</b>       | <b>Anticollision algorithm (Informative) .....</b>      | <b>81</b> |
| A.1                     | Algorithm for pulsed slots .....                        | 81        |
| <b>Appendix B</b>       | <b>CRC (Informative) .....</b>                          | <b>82</b> |
| B.1                     | CRC error detection method .....                        | 82        |
| B.2                     | CRC calculation example .....                           | 82        |
| B.3                     | Application family identifier (AFI) (informative) ..... | 84        |
| <b>Revision history</b> | <b>.....</b>                                            | <b>85</b> |

## List of tables

|           |                                                                          |    |
|-----------|--------------------------------------------------------------------------|----|
| Table 1.  | Signal names . . . . .                                                   | 10 |
| Table 2.  | LRI2K memory map . . . . .                                               | 11 |
| Table 3.  | 10% modulation parameters . . . . .                                      | 14 |
| Table 4.  | Response data rate . . . . .                                             | 19 |
| Table 5.  | UID format . . . . .                                                     | 27 |
| Table 6.  | CRC transmission rules . . . . .                                         | 29 |
| Table 7.  | VCD request frame format . . . . .                                       | 30 |
| Table 8.  | LRI2K response frame format . . . . .                                    | 30 |
| Table 9.  | LRI2K response depending on request flags . . . . .                      | 33 |
| Table 10. | General request format . . . . .                                         | 35 |
| Table 11. | Definitions of request flags 1 to 4 . . . . .                            | 35 |
| Table 12. | Request flags 5 to 8 when bit 3 = 0 . . . . .                            | 36 |
| Table 13. | Request flags 5 to 8 when bit 3 = 1 . . . . .                            | 36 |
| Table 14. | General response format . . . . .                                        | 37 |
| Table 15. | Definitions of response flags 1 to 8 . . . . .                           | 37 |
| Table 16. | Response error code definition . . . . .                                 | 38 |
| Table 17. | Inventory request format . . . . .                                       | 39 |
| Table 18. | Example of the addition of 0-bits to an 11-bit mask value . . . . .      | 39 |
| Table 19. | Timing values . . . . .                                                  | 45 |
| Table 20. | Command codes . . . . .                                                  | 46 |
| Table 21. | Inventory request format . . . . .                                       | 47 |
| Table 22. | Inventory response format . . . . .                                      | 47 |
| Table 23. | Stay Quiet request format . . . . .                                      | 48 |
| Table 24. | Read Single Block request format . . . . .                               | 49 |
| Table 25. | Read Single Block response format when Error_flag is NOT set . . . . .   | 49 |
| Table 26. | Block Locking status . . . . .                                           | 49 |
| Table 27. | Read Single Block response format when Error_flag is set . . . . .       | 49 |
| Table 28. | Write Single Block request format . . . . .                              | 51 |
| Table 29. | Write Single Block response format when Error_flag is NOT set . . . . .  | 51 |
| Table 30. | Write Single Block response format when Error_flag is set . . . . .      | 51 |
| Table 31. | Lock Single Block request format . . . . .                               | 52 |
| Table 32. | Lock Block response format when Error_flag is NOT set . . . . .          | 52 |
| Table 33. | Lock Block response format when Error_flag is set . . . . .              | 52 |
| Table 34. | Read Multiple Block request format . . . . .                             | 53 |
| Table 35. | Read Multiple Block response format when Error_flag is NOT set . . . . . | 53 |
| Table 36. | Block Locking status . . . . .                                           | 53 |
| Table 37. | Read Multiple Block response format when Error_flag is set . . . . .     | 53 |
| Table 38. | Select request format . . . . .                                          | 55 |
| Table 39. | Select Block response format when Error_flag is NOT set . . . . .        | 55 |
| Table 40. | Select response format when Error_flag is set . . . . .                  | 55 |
| Table 41. | Reset to Ready request format . . . . .                                  | 56 |
| Table 42. | Reset to Ready response format when Error_flag is NOT set . . . . .      | 56 |
| Table 43. | Reset to ready response format when Error_flag is set . . . . .          | 56 |
| Table 44. | Write AFI request format . . . . .                                       | 57 |
| Table 45. | Write AFI response format when Error_flag is NOT set . . . . .           | 57 |
| Table 46. | Write AFI response format when Error_flag is set . . . . .               | 57 |
| Table 47. | Lock AFI request format . . . . .                                        | 58 |
| Table 48. | Lock AFI response format when Error_flag is NOT set . . . . .            | 58 |

|           |                                                                                                     |    |
|-----------|-----------------------------------------------------------------------------------------------------|----|
| Table 49. | Lock AFI response format when Error_flag is set . . . . .                                           | 58 |
| Table 50. | Write DSFID request format . . . . .                                                                | 59 |
| Table 51. | Write DSFID response format when Error_flag is NOT set . . . . .                                    | 59 |
| Table 52. | Write DSFID response format when Error_flag is set . . . . .                                        | 59 |
| Table 53. | Lock DSFID request format . . . . .                                                                 | 60 |
| Table 54. | Lock DSFID response format when Error_flag is NOT set . . . . .                                     | 60 |
| Table 55. | Lock DSFID response format when Error_flag is set . . . . .                                         | 60 |
| Table 56. | Get System Info request format . . . . .                                                            | 61 |
| Table 57. | Get System Info response format when Error_flag is NOT set . . . . .                                | 61 |
| Table 58. | Get System Info response format when Error_flag is set . . . . .                                    | 61 |
| Table 59. | Get Multiple Block Security Status request format . . . . .                                         | 62 |
| Table 60. | Get Multiple Block Security Status response format when Error_flag is NOT set . . . . .             | 62 |
| Table 61. | Block Locking status . . . . .                                                                      | 62 |
| Table 62. | Get Multiple Block Security Status response format when Error_flag is set . . . . .                 | 62 |
| Table 63. | Kill request format . . . . .                                                                       | 64 |
| Table 64. | Kill response format when Error_flag is NOT set . . . . .                                           | 64 |
| Table 65. | Kill response format when Error_flag is set . . . . .                                               | 64 |
| Table 66. | Write Kill request format . . . . .                                                                 | 65 |
| Table 67. | Write Kill response format when Error_flag is NOT set . . . . .                                     | 65 |
| Table 68. | Write Kill response format when Error_flag is set . . . . .                                         | 65 |
| Table 69. | Lock Kill request format . . . . .                                                                  | 66 |
| Table 70. | Lock Kill response format when Error_flag is NOT set . . . . .                                      | 66 |
| Table 71. | Lock Kill response format when Error_flag is set . . . . .                                          | 66 |
| Table 72. | Fast Read Single Block request format . . . . .                                                     | 68 |
| Table 73. | Fast Read Single Block response format when Error_flag is NOT set . . . . .                         | 68 |
| Table 74. | Block Locking status . . . . .                                                                      | 68 |
| Table 75. | Fast Read Single Block response format when Error_flag is set . . . . .                             | 68 |
| Table 76. | Fast Inventory Initiated request format . . . . .                                                   | 70 |
| Table 77. | Fast Inventory Initiated response format . . . . .                                                  | 70 |
| Table 78. | Fast Initiate request format . . . . .                                                              | 71 |
| Table 79. | Fast Initiate response format . . . . .                                                             | 71 |
| Table 80. | Fast Read Multiple Block request format . . . . .                                                   | 72 |
| Table 81. | Fast Read Multiple Block response format when Error_flag is NOT set . . . . .                       | 72 |
| Table 82. | Block Locking status if Option_flag is set . . . . .                                                | 72 |
| Table 83. | Fast Read Multiple Block response format when Error_flag is set . . . . .                           | 72 |
| Table 84. | Inventory Initiated request format . . . . .                                                        | 74 |
| Table 85. | Inventory Initiated response format . . . . .                                                       | 74 |
| Table 86. | Initiate request format . . . . .                                                                   | 75 |
| Table 87. | Initiate Initiated response format . . . . .                                                        | 75 |
| Table 88. | Absolute maximum ratings . . . . .                                                                  | 76 |
| Table 89. | AC characteristics . . . . .                                                                        | 77 |
| Table 90. | DC characteristics . . . . .                                                                        | 78 |
| Table 91. | Operating conditions . . . . .                                                                      | 78 |
| Table 92. | UFDFPN8 - 8-lead ultra thin fine pitch dual flat package no lead (MLP)<br>mechanical data . . . . . | 79 |
| Table 93. | Ordering information scheme . . . . .                                                               | 80 |
| Table 94. | CRC definition . . . . .                                                                            | 82 |
| Table 95. | AFI coding . . . . .                                                                                | 84 |
| Table 96. | Document revision history . . . . .                                                                 | 85 |

## List of figures

|            |                                                                                 |    |
|------------|---------------------------------------------------------------------------------|----|
| Figure 1.  | Pad connections . . . . .                                                       | 10 |
| Figure 2.  | UFDFPN8 (MLP) connections . . . . .                                             | 10 |
| Figure 3.  | 100% modulation waveform . . . . .                                              | 14 |
| Figure 4.  | 10% modulation waveform . . . . .                                               | 14 |
| Figure 5.  | 1 out of 256 coding mode . . . . .                                              | 15 |
| Figure 6.  | Detail of one time period . . . . .                                             | 16 |
| Figure 7.  | 1 out of 4 coding mode . . . . .                                                | 17 |
| Figure 8.  | 1 out of 4 coding example . . . . .                                             | 17 |
| Figure 9.  | SOF to select 1 out of 256 data coding mode . . . . .                           | 18 |
| Figure 10. | SOF to select 1 out of 4 data coding mode . . . . .                             | 18 |
| Figure 11. | EOF for either data coding mode . . . . .                                       | 18 |
| Figure 12. | Logic 0, high data rate . . . . .                                               | 20 |
| Figure 13. | Logic 0, high data rate x2 . . . . .                                            | 20 |
| Figure 14. | Logic 1, high data rate . . . . .                                               | 20 |
| Figure 15. | Logic 1, high data rate x2 . . . . .                                            | 20 |
| Figure 16. | Logic 0, low data rate . . . . .                                                | 21 |
| Figure 17. | Logic 0, low data rate x2 . . . . .                                             | 21 |
| Figure 18. | Logic 1, low data rate . . . . .                                                | 21 |
| Figure 19. | Logic 1, low data rate x2 . . . . .                                             | 21 |
| Figure 20. | Logic 0, high data rate . . . . .                                               | 22 |
| Figure 21. | Logic 1, high data rate . . . . .                                               | 22 |
| Figure 22. | Logic 0, low data rate . . . . .                                                | 22 |
| Figure 23. | Logic 1, low data rate . . . . .                                                | 22 |
| Figure 24. | Start of frame, high data rate, one subcarrier . . . . .                        | 23 |
| Figure 25. | Start of frame, high data rate, one subcarrier x2 . . . . .                     | 23 |
| Figure 26. | Start of frame, low data rate, one subcarrier . . . . .                         | 24 |
| Figure 27. | Start of frame, low data rate, one subcarrier x2 . . . . .                      | 24 |
| Figure 28. | Start of frame, high data rate, two subcarriers . . . . .                       | 24 |
| Figure 29. | Start of frame, low data rate, two subcarriers . . . . .                        | 24 |
| Figure 30. | End of frame, high data rate, one subcarrier . . . . .                          | 25 |
| Figure 31. | End of frame, high data rate, one subcarrier x2 . . . . .                       | 25 |
| Figure 32. | End of frame, low data rate, one subcarrier . . . . .                           | 25 |
| Figure 33. | End of frame, low data rate, one subcarrier x2 . . . . .                        | 25 |
| Figure 34. | End of frame, high data rate, two subcarriers . . . . .                         | 26 |
| Figure 35. | End of frame, low data rate, two subcarriers . . . . .                          | 26 |
| Figure 36. | LRI2K decision tree for AFI . . . . .                                           | 28 |
| Figure 37. | LRI2K protocol timing . . . . .                                                 | 31 |
| Figure 38. | LRI2K state transition diagram . . . . .                                        | 33 |
| Figure 39. | Principle of comparison between the mask, the slot number and the UID . . . . . | 40 |
| Figure 40. | Description of a possible anticollision sequence . . . . .                      | 43 |
| Figure 41. | Stay Quiet frame exchange between VCD and LRI2K . . . . .                       | 48 |
| Figure 42. | READ Single Block frame exchange between VCD and LRI2K . . . . .                | 50 |
| Figure 43. | Write Single Block frame exchange between VCD and LRI2K . . . . .               | 51 |
| Figure 44. | Lock Block frame exchange between VCD and LRI2K . . . . .                       | 52 |
| Figure 45. | Read Multiple Block frame exchange between VCD and LRI2K . . . . .              | 54 |
| Figure 46. | Select frame exchange between VCD and LRI2K . . . . .                           | 55 |
| Figure 47. | Reset to Ready frame exchange between VCD and LRI2K . . . . .                   | 56 |
| Figure 48. | Write AFI frame exchange between VCD and LRI2K . . . . .                        | 57 |



|            |                                                                                          |    |
|------------|------------------------------------------------------------------------------------------|----|
| Figure 49. | Lock AFI frame exchange between VCD and LRI2K . . . . .                                  | 58 |
| Figure 50. | Write DSFID frame exchange between VCD and LRI2K . . . . .                               | 59 |
| Figure 51. | Lock DSFID frame exchange between VCD and LRI2K . . . . .                                | 60 |
| Figure 52. | Get System Info frame exchange between VCD and LRI2K . . . . .                           | 61 |
| Figure 53. | Get Multiple Block Security Status frame exchange between VCD and LRI2K . . . . .        | 63 |
| Figure 54. | Kill frame exchange between VCD and LRI2K . . . . .                                      | 64 |
| Figure 55. | Write Kill frame exchange between VCD and LRI2K . . . . .                                | 65 |
| Figure 56. | Lock Kill frame exchange between VCD and LRI2K . . . . .                                 | 67 |
| Figure 57. | Fast Read Single Block frame exchange between VCD and LRI2K . . . . .                    | 69 |
| Figure 58. | Fast Initiate frame exchange between VCD and LRI2K . . . . .                             | 71 |
| Figure 59. | Fast Read Multiple Block frame exchange between VCD and LRI2K . . . . .                  | 73 |
| Figure 60. | Initiate frame exchange between VCD and LRI2K . . . . .                                  | 75 |
| Figure 61. | LRI2K synchronous timing, transmit and receive . . . . .                                 | 78 |
| Figure 62. | UFDFPN8 - 8-lead ultra thin fine pitch dual flat package no lead (MLP) outline . . . . . | 79 |

# 1 Description

The LRI2K is a contactless memory powered by the received carrier electromagnetic wave. It is a 2048-bit electrically erasable programmable memory (EEPROM). The memory is organized as 64 blocks of 32 bits. The LRI2K 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 LRI2K load variation using Manchester coding with one or two subcarrier frequencies at 423 kHz and 484 kHz. Data are transferred from the LRI2K at 6.6 Kbit/s in low data rate mode and 26 Kbit/s fast data rate mode. The LRI2K supports 53 Kbit/s in high data rate mode with one subcarrier frequency at 423 kHz.

The LRI2K follows the ISO 15693 recommendation for radio-frequency power and signal interface.

Figure 1. Pad connections

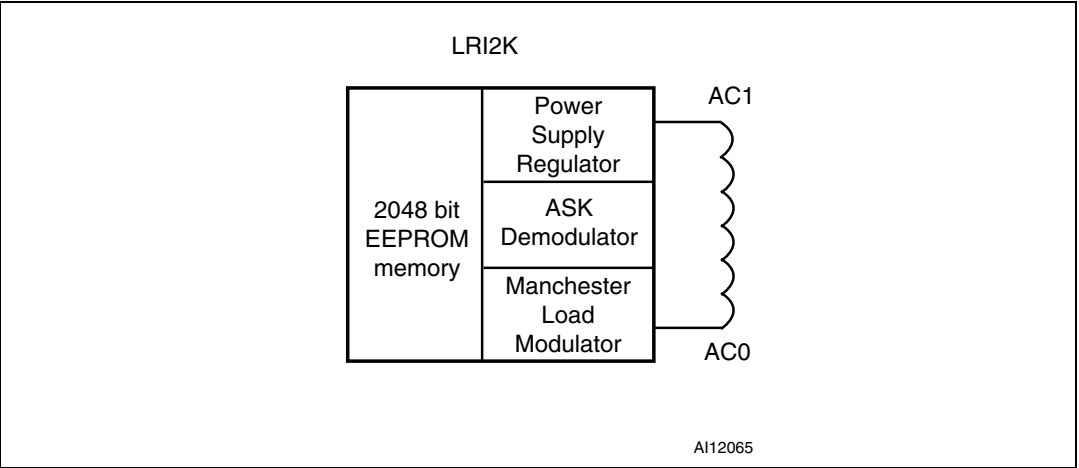
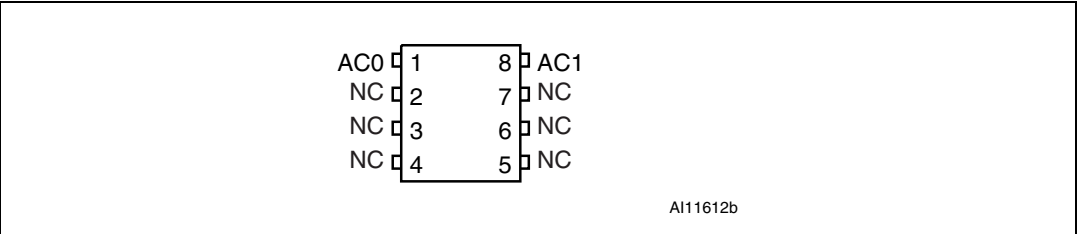


Table 1. Signal names

| Signal name | Function     |
|-------------|--------------|
| AC1         | Antenna coil |
| AC0         | Antenna coil |

Figure 2. UFDFPN8 (MLP) connections



1. NC means not connected internally.

## 1.1 Memory mapping

The LRI2K is divided into 64 blocks of 32 bits. Each block can be individually write-protected using the Lock command.

**Table 2. LRI2K memory map**

| Add | 0         | 7 | 8     | 15 | 16    | 23 | 24    | 31 |
|-----|-----------|---|-------|----|-------|----|-------|----|
| 0   | User area |   |       |    |       |    |       |    |
| 1   | User area |   |       |    |       |    |       |    |
| 2   | User area |   |       |    |       |    |       |    |
| 3   | User area |   |       |    |       |    |       |    |
| 4   | User area |   |       |    |       |    |       |    |
| 5   | User area |   |       |    |       |    |       |    |
| 6   | User area |   |       |    |       |    |       |    |
| 7   | User area |   |       |    |       |    |       |    |
| 8   | User area |   |       |    |       |    |       |    |
|     | User area |   |       |    |       |    |       |    |
|     | User area |   |       |    |       |    |       |    |
|     | User area |   |       |    |       |    |       |    |
| 60  | User area |   |       |    |       |    |       |    |
| 61  | User area |   |       |    |       |    |       |    |
| 62  | User area |   |       |    |       |    |       |    |
| 63  | User area |   |       |    |       |    |       |    |
|     |           |   |       |    |       |    |       |    |
|     | UID 0     |   | UID 1 |    | UID 2 |    | UID 3 |    |
|     | UID 4     |   | UID 5 |    | UID 6 |    | UID 7 |    |
|     | AFI       |   | DSFID |    |       |    |       |    |
|     | Kill code |   |       |    |       |    |       |    |

The User area consists of blocks that are always accessible in read mode. Write operations are possible if the addressed block is not protected. During a write operation, the 32 bits of the block are replaced by the new 32-bit value.

The LRI2K also has a 64-bit block that is used to store the 64-bit unique identifier (UID). The UID is compliant to 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 LRI2K 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 LRI2K has an additional 32-bit block in which the kill code is stored.

## 1.2 Commands

The LRI2K supports the following commands:

- **Inventory**, used to perform the anticollision sequence.
- **Stay Quiet**, used to put the LRI2K in quiet mode, where it does not respond to any inventory command.
- **Select**, used to select the LRI2K. After this command, the LRI2K processes all Read/Write commands with Select\_flag set.
- **Reset To Ready**, used to put the LRI2K 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.
- **Lock Block**, used to lock the selected block. After this command, the block cannot be modified.
- **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.
- **Kill**, used to definitively deactivate the tag.
- **Write Kill**, used to write the 32-bit Kill code value
- **Lock Kill**, used to lock the Kill Code register.
- **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 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.

## 1.3 Initial dialogue for vicinity cards

The dialog between the vicinity coupling device (VCD) and the vicinity integrated circuit card or VICC (LRI2K) takes place as follows:

- activation of the LRI2K by the RF operating field of the VCD
- transmission of a command by the VCD
- transmission of a response by the LRI2K

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).

### 1.3.1 Power transfer

Power is transferred to the LRI2K by radio frequency at 13.56 MHz via coupling antennas in the LRI2K and the VCD. The RF operating field of the VCD is transformed on the LRI2K antenna as an AC voltage which is rectified, filtered and internally regulated. The amplitude modulation (ASK) on this received signal is demodulated by the ASK demodulator.

### 1.3.2 Frequency

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

### 1.3.3 Operating field

The LRI2K operates continuously between  $H_{min}$  and  $H_{max}$ .

- The minimum operating field is  $H_{min}$  and has a value of 150 mA/m rms.
- The maximum operating field is  $H_{max}$  and has a value of 5 A/m rms.

A VCD must generate a field of at least  $H_{min}$  and not exceeding  $H_{max}$  in the operating volume.

## 2 Communication signal from VCD to LRI2K

Communications between the VCD and the LRI2K take place using the modulation principle of ASK (amplitude shift keying). Two modulation indexes are used, 10% and 100%. The LRI2K 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 3](#) and [Figure 4](#).

The LRI2K is operational for any degree of modulation index between 10% and 30%.

Figure 3. 100% modulation waveform

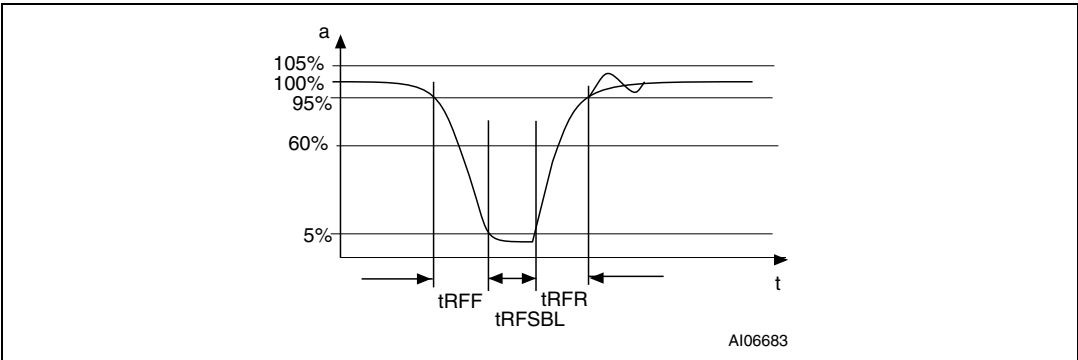
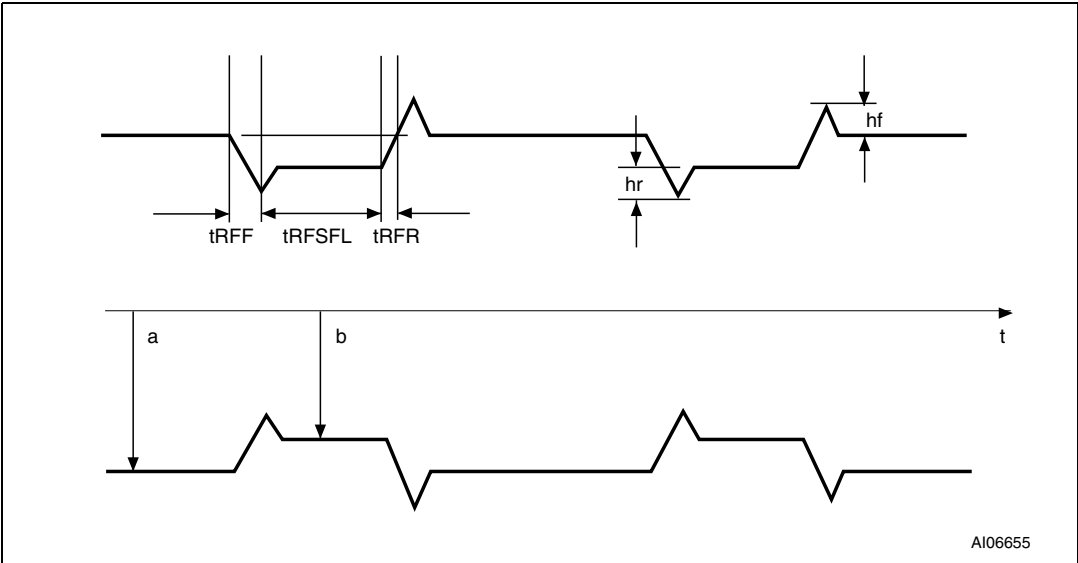


Table 3. 10% modulation parameters

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

Figure 4. 10% modulation waveform



### 3 Data rate and data coding

The data coding implemented in the LRI2K uses pulse position modulation. Both data coding modes that are described in the ISO 15693 are supported by the LRI2K. The selection is made by the VCD and indicated to the LRI2K within the start of frame (SOF).

#### 3.1 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 5* illustrates this pulse position modulation technique. In this Figure, data E1h (225 decimal) is sent by the VCD to the LRI2K.

The pause occurs during the second half of the position of the time period that determines the value, as shown in *Figure 6*.

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

**Figure 5. 1 out of 256 coding mode**

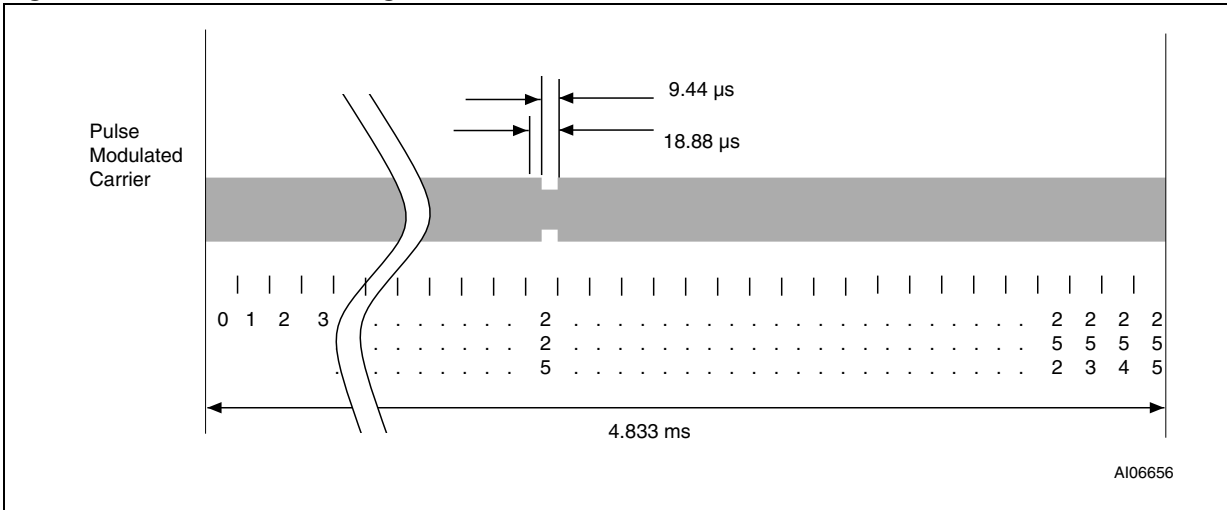
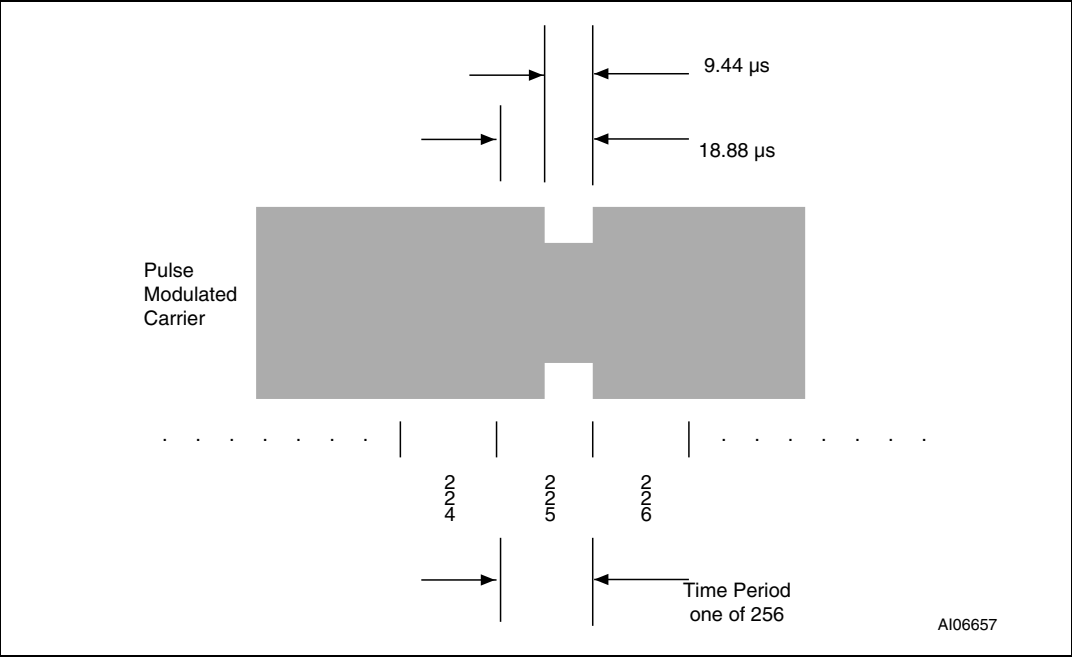


Figure 6. Detail of one time period



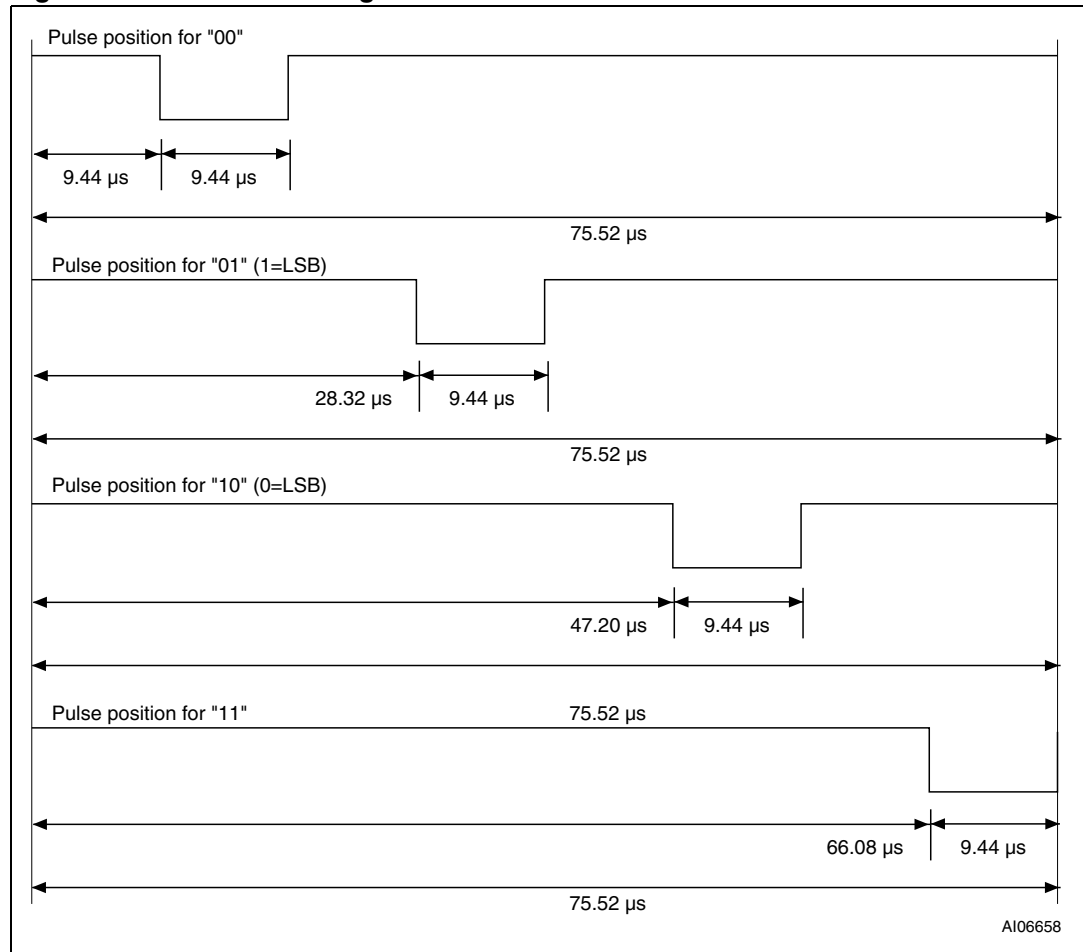


### 3.2 Data coding mode: 1 out of 4

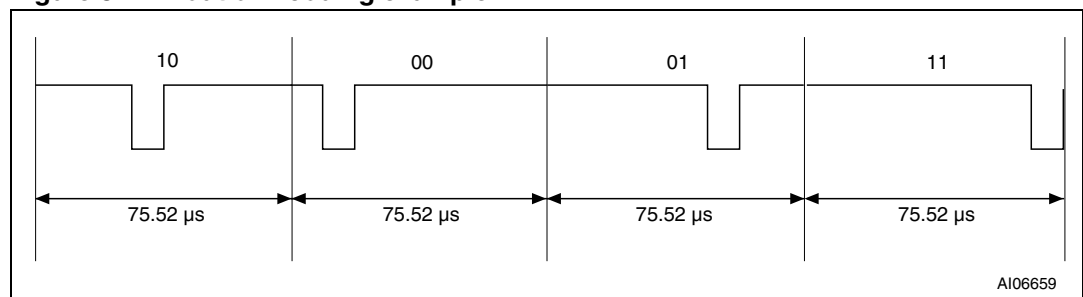
The value of 2 bits is represented by the position of one pause. The position of the pause on 1 of 4 successive time periods of  $18.88\ \mu\text{s}$  ( $256/f_C$ ) determines the value of the 2 bits. Four successive pairs of bits form a byte, where the least significant pair of bits is transmitted first.

In this case the transmission of one byte takes  $302.08\ \mu\text{s}$  and the resulting data rate is  $26.48\ \text{Kbit/s}$  ( $f_C/512$ ). [Figure 7](#) illustrates the 1 out of 4 pulse position technique and coding. [Figure 8](#) shows the transmission of E1h ( $225_{10}$  -  $1110\ 0001_{2}$ ) by the VCD.

**Figure 7. 1 out of 4 coding mode**



**Figure 8. 1 out of 4 coding example**



### 3.3 VCD to LRI2K 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 LRI2K is ready to receive a new command frame from the VCD 311.5  $\mu\text{s}$  ( $t_2$ ) after sending a response frame to the VCD.

The LRI2K takes a Power-On time of 0.1 ms after being activated by the powering field. After this delay, the LRI2K is ready to receive a command frame from the VCD.

### 3.4 Start of frame (SOF)

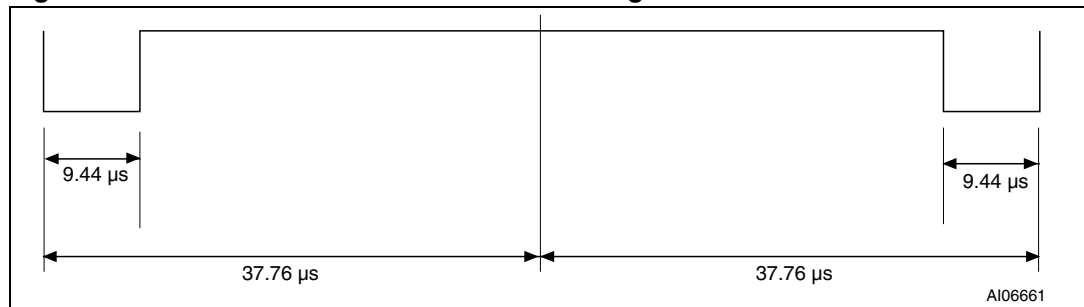
The SOF defines the data coding mode the VCD is to use for the following command frame.

The SOF sequence described in [Figure 9](#) selects the 1 out of 256 data coding mode.

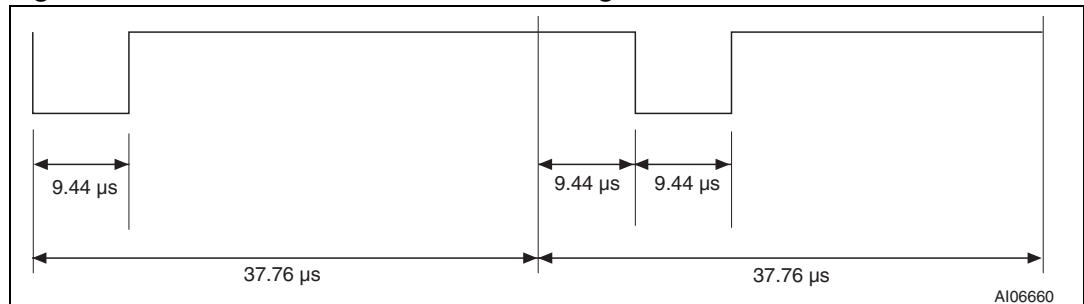
The SOF sequence described in [Figure 10](#) selects the 1 out of 4 data coding mode.

The EOF sequence for either coding mode is described in [Figure 11](#).

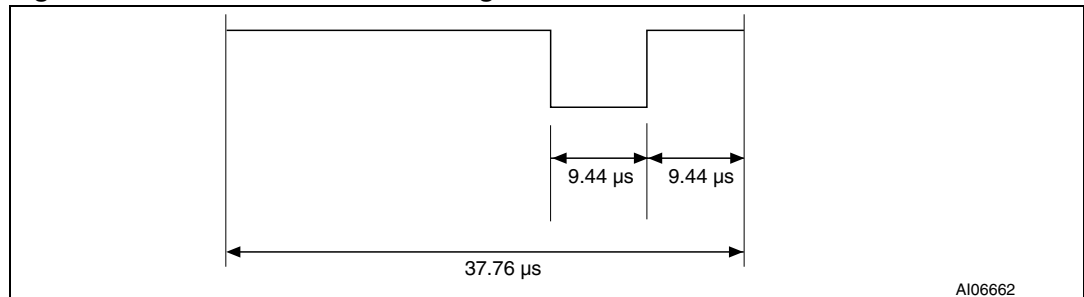
**Figure 9. SOF to select 1 out of 256 data coding mode**



**Figure 10. SOF to select 1 out of 4 data coding mode**



**Figure 11. EOF for either data coding mode**



## 4 Communications signal from LRI2K to VCD

The LRI2K has several modes defined for some parameters, owing to which it can operate in different noise environments and meet different application requirements.

### 4.1 Load modulation

The LRI2K is capable of communication with 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 LRI2K.

The load-modulated amplitude received on the VCD antenna shall be at least 10 mV when measured as described in the test methods defined in International Standard ISO 10373-7.

### 4.2 Subcarrier

The LRI2K 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, 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 LRI2K generates a continuous phase relationship between  $f_{S1}$  and  $f_{S2}$ .

### 4.3 Data rates

The LRI2K 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 4](#) shows the different data rates produced by the LRI2K using the different response format combinations.

**Table 4. Response data rate**

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

## 5 Bit representation and coding

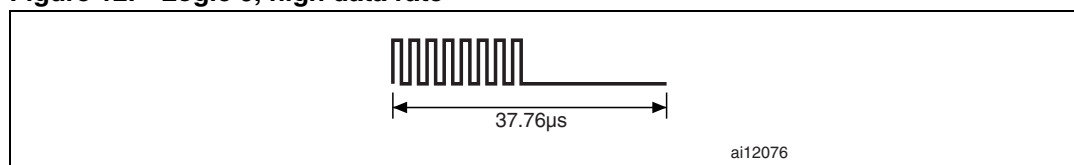
Data bits are encoded using Manchester coding, according to the following schemes. For the low data rate, the same subcarrier frequency or frequencies is/are used, in this case the number of pulses is multiplied by 4 and all times are increased by this factor. For the Fast commands using one subcarrier,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

### 5.1 Bit coding using one subcarrier

#### 5.1.1 High data rate

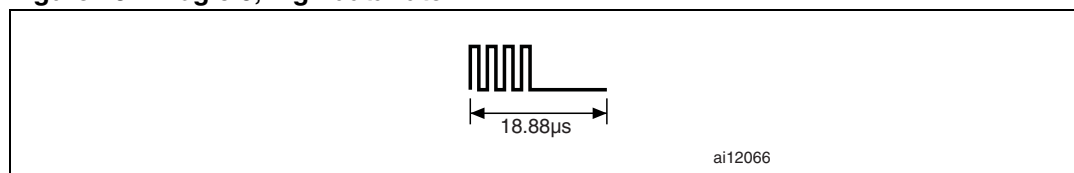
A logic 0 starts with 8 pulses at 423.75 kHz ( $f_C/32$ ) followed by an unmodulated time of 18.88  $\mu\text{s}$  as shown in [Figure 12](#).

**Figure 12. Logic 0, high data rate**



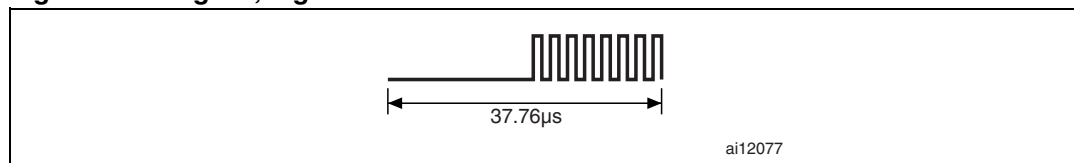
For the Fast commands, a logic 0 starts with 4 pulses at 423.75 kHz ( $f_C/32$ ) followed by an unmodulated time of 9.44  $\mu\text{s}$  as shown in [Figure 13](#).

**Figure 13. Logic 0, high data rate x2**



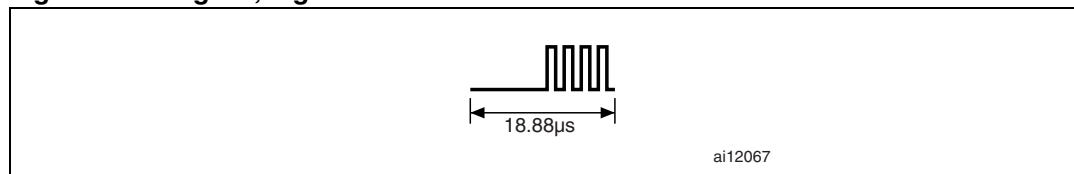
A logic 1 starts with an unmodulated time of 18.88  $\mu\text{s}$  followed by 8 pulses at 423.75 kHz ( $f_C/32$ ) as shown in [Figure 14](#).

**Figure 14. Logic 1, high data rate**



For the Fast commands, a logic 1 starts with an unmodulated time of 9.44  $\mu\text{s}$  followed by 4 pulses at 423.75 kHz ( $f_C/32$ ) as shown in [Figure 15](#).

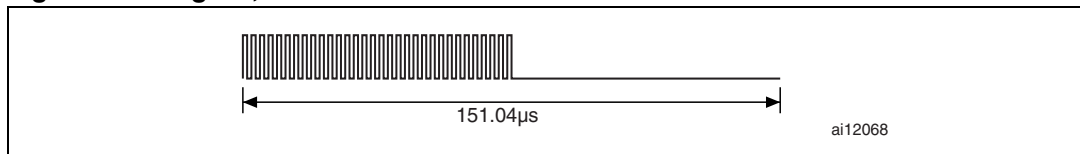
**Figure 15. Logic 1, high data rate x2**



### 5.1.2 Low data rate

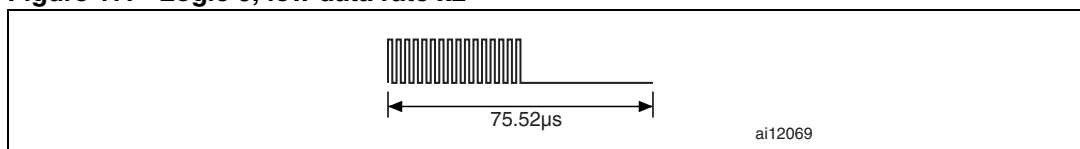
A logic 0 starts with 32 pulses at 423.75 kHz ( $f_C/32$ ) followed by an unmodulated time of 75.52  $\mu\text{s}$  as shown in [Figure 16](#).

**Figure 16. Logic 0, low data rate**



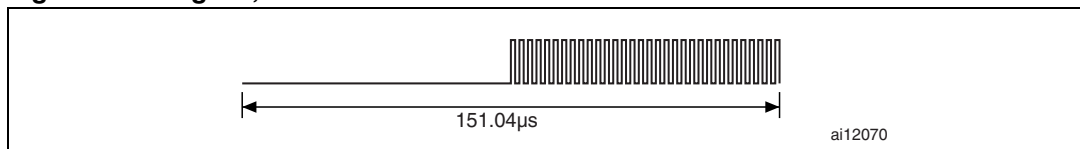
For the fast commands, a logic 0 starts with 16 pulses of 423,75 kHz ( $f_C/32$ ) followed by an unmodulated time of 37,76  $\mu\text{s}$  as shown in [Figure 17](#).

**Figure 17. Logic 0, low data rate x2**



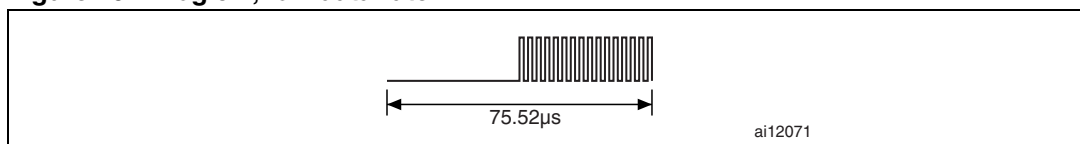
A logic 1 starts with an unmodulated time of 75,52  $\mu\text{s}$  followed by 32 pulses of 423,75 kHz ( $f_C/32$ ) as shown in [Figure 18](#).

**Figure 18. Logic 1, low data rate**



For the Fast commands, a logic 1 starts with an unmodulated time of 37.76  $\mu\text{s}$  followed by 16 pulses at 423.75 kHz ( $f_C/32$ ) as shown in [Figure 19](#).

**Figure 19. Logic 1, low data rate x2**

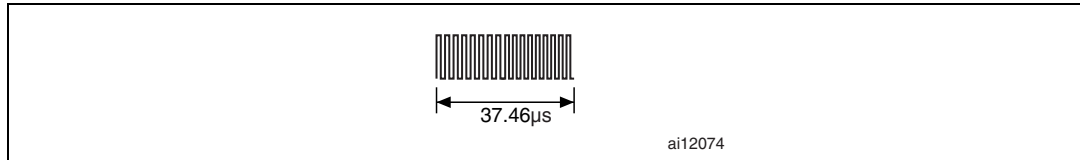


## 5.2 Bit coding using two subcarriers

### 5.2.1 High data rate

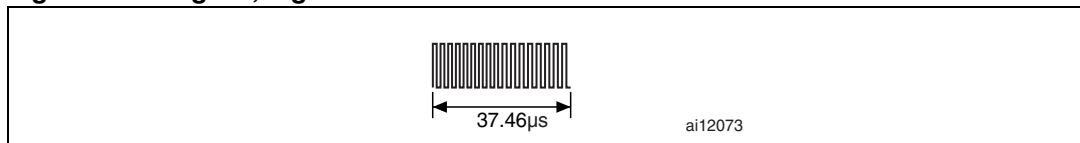
A logic 0 starts with 8 pulses at 423.75 kHz ( $f_C/32$ ) followed by 9 pulses at 484.28 kHz ( $f_C/28$ ) as shown in [Figure 20](#). For the Fast commands, the x2 mode is not available.

**Figure 20. 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 21](#). For the Fast commands, the x2 mode is not available.

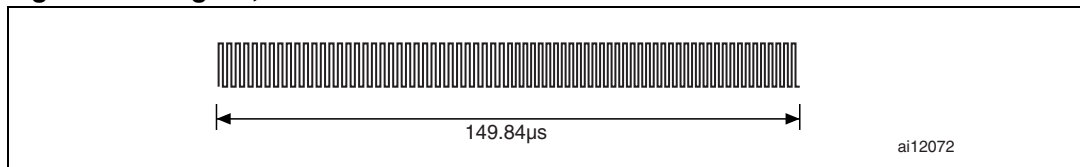
**Figure 21. Logic 1, high data rate**



### 5.2.2 Low data rate

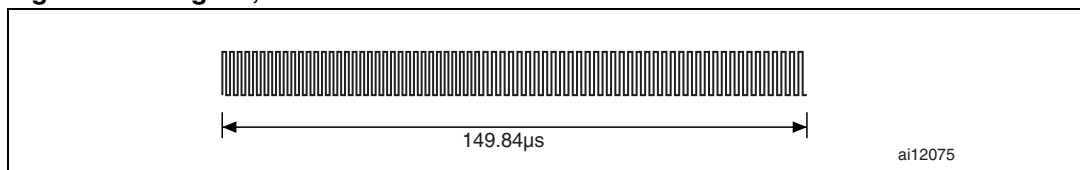
A logic 0 starts with 32 pulses at 423.75 kHz ( $f_C/32$ ) followed by 36 pulses at 484.28 kHz ( $f_C/28$ ) as shown in [Figure 22](#). For the Fast commands, the x2 mode is not available.

**Figure 22. Logic 0, low data rate**



A logic 1 starts with 36 pulses at 484.28kHz ( $f_C/28$ ) followed by 32 pulses at 423.75kHz ( $f_C/32$ ) as shown in [Figure 23](#). For the fast commands, the x2 mode is not available.

**Figure 23. Logic 1, low data rate**



## 6 LRI2K to VCD frames

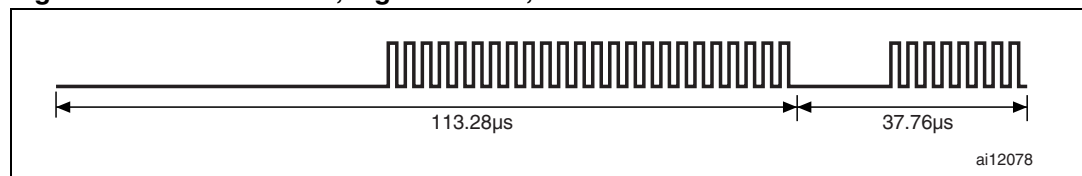
Frames are delimited by an SOF and an EOF. They are implemented using code violation. Unused options are reserved for future use. For the low data rate, the same subcarrier frequency or frequencies is/are used. In this case the number of pulses is multiplied by 4. For the Fast commands using one subcarrier,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

### 6.1 SOF when using one subcarrier

#### 6.1.1 High data rate

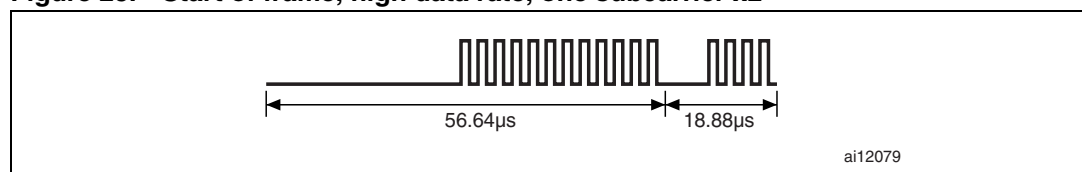
The SOF includes an unmodulated time of 56.64  $\mu\text{s}$  followed by 24 pulses at 423.75 kHz ( $f_C/32$ ), and a logic 1 that consists of an unmodulated time of 18.88  $\mu\text{s}$  followed by 8 pulses at 423.75 kHz. The SOF is shown in [Figure 24](#).

**Figure 24. Start of frame, high data rate, one subcarrier**



For the Fast commands, the SOF comprises an unmodulated time of 28.32  $\mu\text{s}$ , followed by 12 pulses at 423.75 kHz ( $f_C/32$ ), and a logic 1 that consists of an unmodulated time of 9.44  $\mu\text{s}$  followed by 4 pulses at 423.75 kHz as shown in [Figure 25](#).

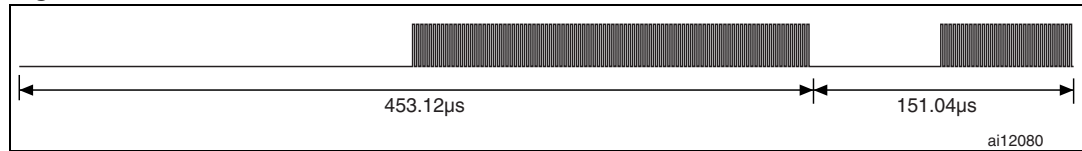
**Figure 25. Start of frame, high data rate, one subcarrier x2**



### 6.1.2 Low data rate

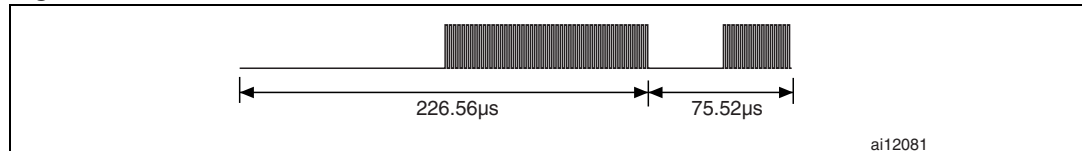
SOF comprises an unmodulated time of 226.56  $\mu\text{s}$ , followed by 96 pulses at 423.75 kHz ( $f_C/32$ ), and a logic 1 that consists of an unmodulated time of 75.52  $\mu\text{s}$  followed by 32 pulses at 423.75 kHz as shown in [Figure 26](#).

**Figure 26. Start of frame, low data rate, one subcarrier**



For the Fast commands, the SOF comprises an unmodulated time of 113.28  $\mu\text{s}$  followed by 48 pulses at 423.75 kHz ( $f_C/32$ ), and a logic 1 that includes an unmodulated time of 37.76  $\mu\text{s}$  followed by 16 pulses at 423.75 kHz as shown in [Figure 27](#).

**Figure 27. Start of frame, low data rate, one subcarrier x2**



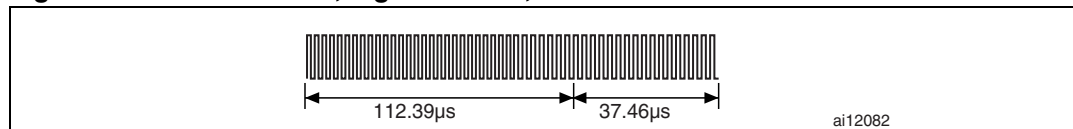
## 6.2 SOF when using two subcarriers

### 6.2.1 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 28](#).

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

**Figure 28. Start of frame, high data rate, two subcarriers**

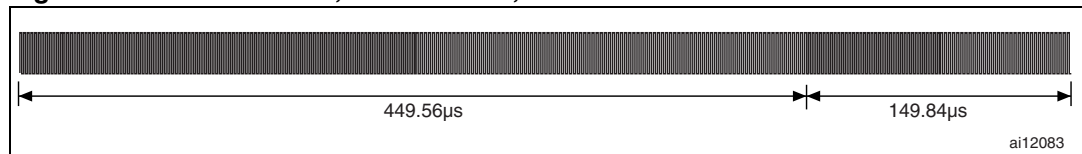


### 6.2.2 Low data rate

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 29](#).

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

**Figure 29. Start of frame, low data rate, two subcarriers**



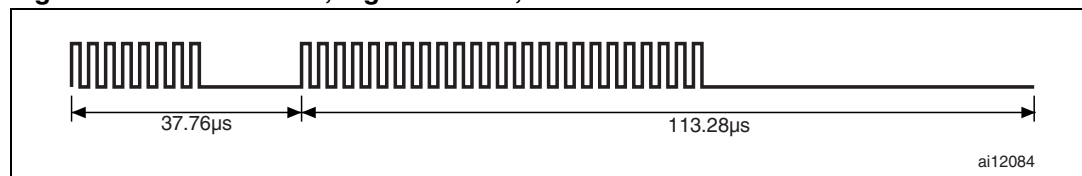


## 6.3 EOF when using one subcarrier

### 6.3.1 High data rate

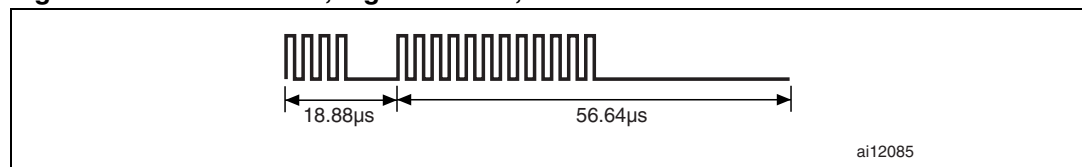
The EOF comprises a logic 0 that includes 8 pulses at 423.75 kHz and an unmodulated time of 18.88  $\mu\text{s}$ , followed by 24 pulses at 423.75 kHz ( $f_C/32$ ) and by an unmodulated time of 56.64  $\mu\text{s}$  as shown in [Figure 30](#).

**Figure 30. End of frame, high data rate, one subcarrier**



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  $\mu\text{s}$ , followed by 12 pulses at 423.75 kHz ( $f_C/32$ ) and an unmodulated time of 28.32  $\mu\text{s}$  as shown in [Figure 31](#).

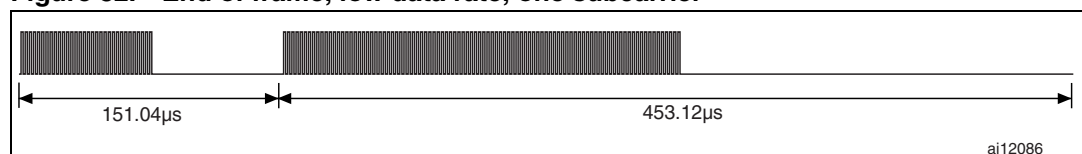
**Figure 31. End of frame, high data rate, one subcarrier x2**



### 6.3.2 Low data rate

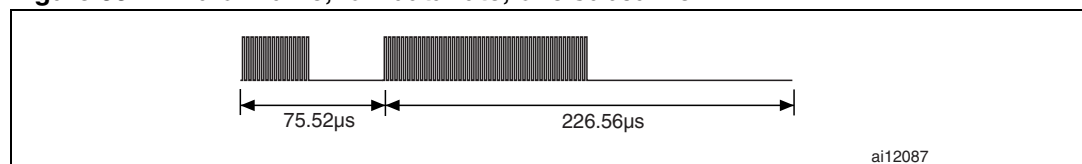
The EOF comprises a logic 0 that includes 32 pulses at 423.75 kHz and an unmodulated time of 75.52  $\mu\text{s}$ , followed by 96 pulses at 423.75 kHz ( $f_C/32$ ) and an unmodulated time of 226.56  $\mu\text{s}$  as shown in [Figure 32](#).

**Figure 32. End of frame, low data rate, one subcarrier**



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  $\mu\text{s}$ , followed by 48 pulses at 423.75 kHz ( $f_C/32$ ) and an unmodulated time of 113.28  $\mu\text{s}$  as shown in [Figure 33](#).

**Figure 33. End of frame, low data rate, one subcarrier x2**



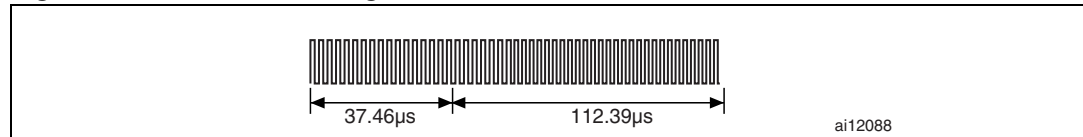
## 6.4 EOF when using two subcarriers

### 6.4.1 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 ( $f_C/32$ ) and 27 pulses at 484.28 kHz ( $f_C/28$ ) as shown in [Figure 34](#).

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

**Figure 34. End of frame, high data rate, two subcarriers**

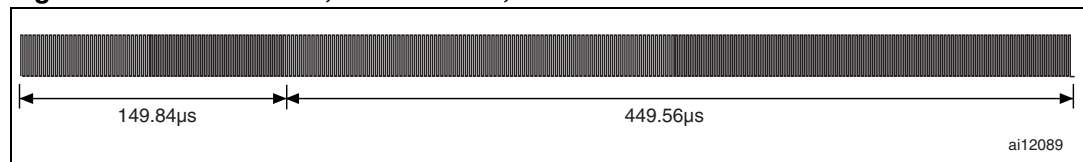


### 6.4.2 Low data rate

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 ( $f_C/32$ ) and 108 pulses at 484.28 kHz ( $f_C/28$ ) as shown in [Figure 35](#).

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

**Figure 35. End of frame, low data rate, two subcarriers**



## 7 Unique identifier (UID)

The LRI2Ks are 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:

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

**Table 5. UID format**

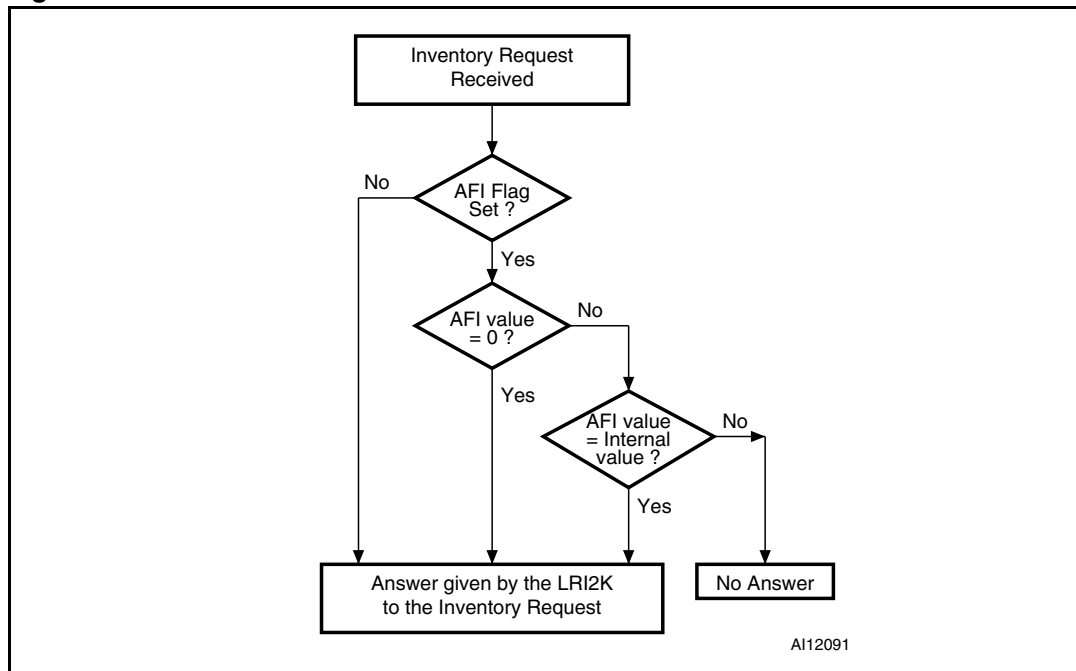
| MSB |    |     |    | LSB                  |   |
|-----|----|-----|----|----------------------|---|
| 63  | 56 | 55  | 48 | 47                   | 0 |
| E0h |    | 02h |    | Unique serial number |   |

With the UID each LRI2K can be addressed uniquely and individually during the anticollision loop and for one-to-one exchanges between a VCD and an LRI2K.

## 8 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 LRI2Ks present, only the LRI2Ks that meet the required application criteria.

**Figure 36. LRI2K decision tree for AFI**



The AFI is programmed by the LRI2K 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)

## 9 Data storage format identifier (DSFID)

The data storage format identifier indicates how the data is structured in the LRI2K 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, respectively. It is coded on one byte.

### 9.1 CRC

The CRC used in the LRI2K is calculated as per the definition in ISO/IEC 13239.

The initial register contents are all ones: "FFFF".

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

Upon reception of a request from the VCD, the LRI2K verifies that the CRC value is valid. If it is invalid, the LRI2K discards the frame and does not answer to the VCD.

Upon reception of a response from the LRI2K, 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 designers.

The CRC is transmitted least significant byte first.

Each byte is transmitted least significant bit first.

**Table 6. CRC transmission rules**

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

## 10 LRI2K protocol description

The transmission protocol (or simply protocol) defines the mechanism used to exchange instructions and data between the VCD and the LRI2K, in both directions. It is based on the concept of "VCD talks first".

This means that an LRI2K 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 LRI2K
- a response from the LRI2K to the VCD

Each request and each response are contained in a frame. The frame delimiters (SOF, EOF) are described in [Section 6: LRI2K to VCD frames](#).

Each request consists of:

- a request SOF (see [Figure 9](#) and [Figure 10](#))
- flags
- a command code
- parameters, depending on the command
- application data
- a 2-byte CRC
- a request EOF (see [Figure 11](#))

Each response consists of:

- an answer SOF (see [Figure 24](#) to [Figure 29](#))
- flags
- parameters, depending on the command
- application data
- a 2-byte CRC
- an Answer EOF (see [Figure 30](#) to [Figure 35](#))

The protocol is bit-oriented. The number of bits transmitted in a frame is a multiple of eight (8), i.e. 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, with each byte 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.

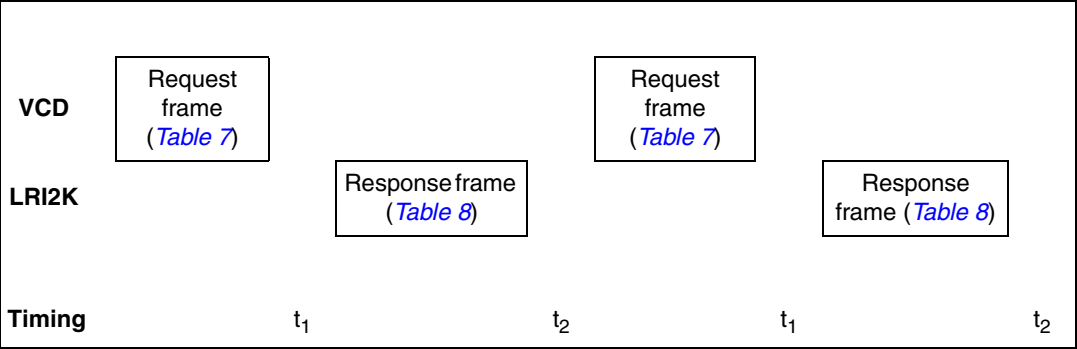
**Table 7. VCD request frame format**

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

**Table 8. LRI2K response frame format**

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

Figure 37. LRI2K protocol timing



## 11 LRI2K states

An LRI2K can be in one of 4 states:

- Power-off
- Ready
- Quiet
- Selected

Transitions between these states are specified in [Figure 38: LRI2K state transition diagram](#) and [Table 9: LRI2K response depending on request flags](#).

### 11.1 Power-off state

The LRI2K is in the Power-off state when it does not receive enough energy from the VCD.

### 11.2 Ready state

The LRI2K is in the Ready state when it receives enough energy from the VCD. When in the Ready state, the LRI2K answers any request where the Select\_flag is not set.

### 11.3 Quiet state

When in the Quiet state, the LRI2K answers any request except for Inventory requests with the Address\_flag set.

### 11.4 Selected state

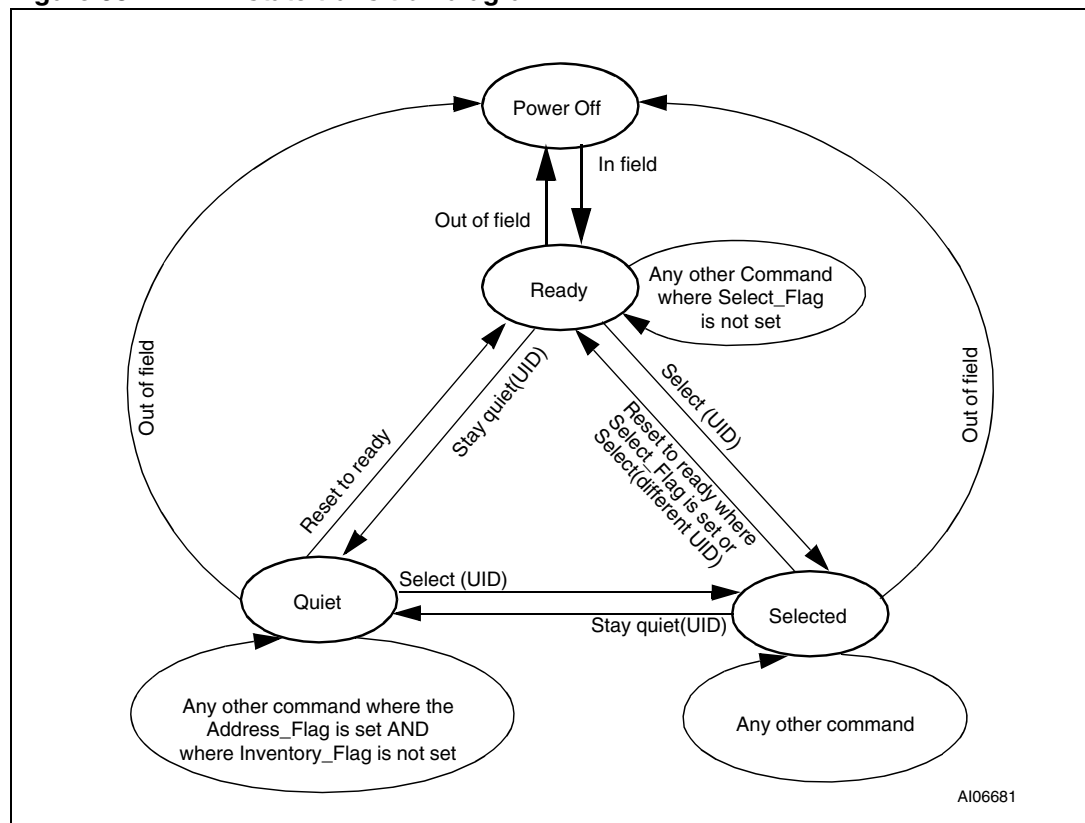
In the Selected state, the LRI2K answers any request in all modes (see [Section 12: Modes](#)):

- 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 9. LRI2K response depending on request flags**

| Flags                                                                       | Address_flag   |                    | Select_flag   |                   |
|-----------------------------------------------------------------------------|----------------|--------------------|---------------|-------------------|
|                                                                             | 1<br>Addressed | 0<br>Non addressed | 1<br>Selected | 0<br>Non selected |
| LRI2K in Ready or Selected state<br>(Devices in Quiet state don't answer)   |                | X                  |               | X                 |
| LRI2K in Selected state                                                     |                | X                  | X             |                   |
| LRI2K in Ready, Quiet or Selected state<br>(the device which match the UID) | X              |                    |               | X                 |
| Error (03h)                                                                 | X              |                    | X             |                   |

**Figure 38. LRI2K state transition diagram**

1. The intention of the state transition method is that only one LRI2K should be in the selected state at a time.

## 12 Modes

The term “mode” refers to the mechanism used in a request to specify the set of LRI2Ks that will answer the request.

### 12.1 Addressed mode

When the Address\_flag is set to 1 (Addressed mode), the request contains the Unique ID (UID) of the addressed LRI2K.

Any LRI2K that receives a request with the Address\_flag set to 1 compares the received Unique ID to its own. If it matches, then the LRI2K executes the request (if possible) and returns a response to the VCD as specified in the command description.

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

### 12.2 Non-Addressed mode (general request)

When the Address\_flag is set to 0 (Non-Addressed mode), the request does not contain a Unique ID. Any LRI2K receiving a request with the Address\_flag set to 0 executes it and returns a response to the VCD as specified in the command description.

### 12.3 Select mode

When the Select\_flag is set to 1 (Select mode), the request does not contain an LRI2K Unique ID. The LRI2K 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 LRI2Ks in the Selected state answer to a request where the Select Flag is set to 1.

The system design ensures in theory that only one LRI2K can be in the Select state at a time.

## 13 Request format

The request consists of:

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

**Table 10. General request format**

|             |               |              |            |      |     |             |
|-------------|---------------|--------------|------------|------|-----|-------------|
| S<br>O<br>F | Request flags | Command code | Parameters | Data | CRC | E<br>O<br>F |
|-------------|---------------|--------------|------------|------|-----|-------------|

### 13.1 Request flags

In a request, the "flags" field specifies the actions to be performed by the LRI2K 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 LRI2K selection criteria.

When bit 3 is set (1), bits 5 to 8 define the LRI2K Inventory parameters.

**Table 11. Definitions of request flags 1 to 4**

| Bit No | Flag                           | Level | Description                                                          |
|--------|--------------------------------|-------|----------------------------------------------------------------------|
| Bit 1  | Subcarrier_flag <sup>(1)</sup> | 0     | A single subcarrier frequency is used by the LRI2K                   |
|        |                                | 1     | Two subcarriers are used by the LRI2K                                |
| Bit 2  | Data_rate_flag <sup>(2)</sup>  | 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 <a href="#">Table 12</a> |
|        |                                | 1     | The meaning of Flags 5 to 8 is described in <a href="#">Table 13</a> |
| Bit 4  | Protocol Extension flag        | 0     | No Protocol format extension                                         |

1. Subcarrier\_flag refers to the LRI2K-to-VCD communication.

2. Data\_rate\_flag refers to the LRI2K-to-VCD communication

**Table 12. Request flags 5 to 8 when bit 3 = 0**

| Bit No | Flag                        | Level | Description                                                                                                                               |
|--------|-----------------------------|-------|-------------------------------------------------------------------------------------------------------------------------------------------|
| Bit 5  | Select_flag <sup>(1)</sup>  | 0     | Request is executed by any LRI2K according to the setting of Address_flag                                                                 |
|        |                             | 1     | Request is executed only by the LRI2K in Selected state                                                                                   |
| Bit 6  | Address_flag <sup>(1)</sup> | 0     | Request is not addressed. UID field is not present. The request is executed by all LRI2Ks.                                                |
|        |                             | 1     | Request is addressed. UID field is present. The request is executed only by the LRI2K whose UID matches the UID specified in the request. |
| Bit 7  | Option flag                 | 0     |                                                                                                                                           |
| 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 13. 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     |                          |

## 14 Response format

The response consists of:

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

**Table 14. General response format**

|             |                |            |      |     |             |
|-------------|----------------|------------|------|-----|-------------|
| S<br>O<br>F | Response flags | Parameters | Data | CRC | E<br>O<br>F |
|-------------|----------------|------------|------|-----|-------------|

### 14.1 Response flags

In a response, the flags indicate how actions have been performed by the LRI2K and whether corresponding fields are present or not. The response flags consist of eight bits.

**Table 15. 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     |                                                     |

## 14.2 Response error code

If the Error\_flag is set by the LRI2K in the response, the Error code field is present and provides information about the error that occurred.

Error codes not specified in [Table 16](#) are reserved for future use.

**Table 16. Response error code definition**

| Error code | Meaning                                                                    |
|------------|----------------------------------------------------------------------------|
| 03h        | The command option is not supported                                        |
| 0F         | Error with no information given or a specific error code is not supported. |
| 10h        | The specified block is not available (does not exist).                     |
| 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.                           |

## 15 Anticollision

The purpose of the anticollision sequence is to inventory the LRI2Ks present in the VCD field using their unique ID (UID).

The VCD is the master of communications with one or several LRI2Ks. It initiates LRI2K communication by issuing the Inventory request.

The LRI2K sends its response in the determined slot or does not respond.

### 15.1 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. The 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 on the next byte boundary.

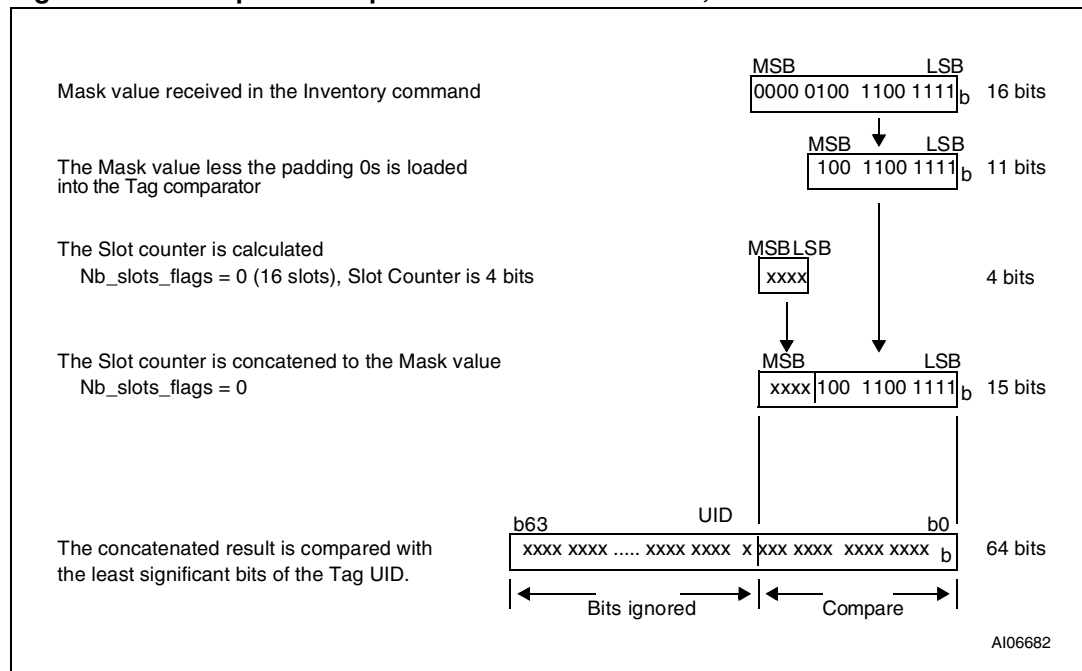
**Table 17. Inventory request format**

| 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 [Table 18](#) and [Figure 39](#), 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 18. Example of the addition of 0-bits to an 11-bit mask value**

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

**Figure 39. 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 LRI2K answer is detected, the VCD may switch to the next slot by sending an EOF,
- if one or more LRI2K answers are detected, the VCD waits until the complete frame has been received before sending an EOF for switching to the next slot.



## 16 Request processing by the LRI2K

Upon reception of a valid request, the LRI2K 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
```

## 17 Explanation of the possible cases

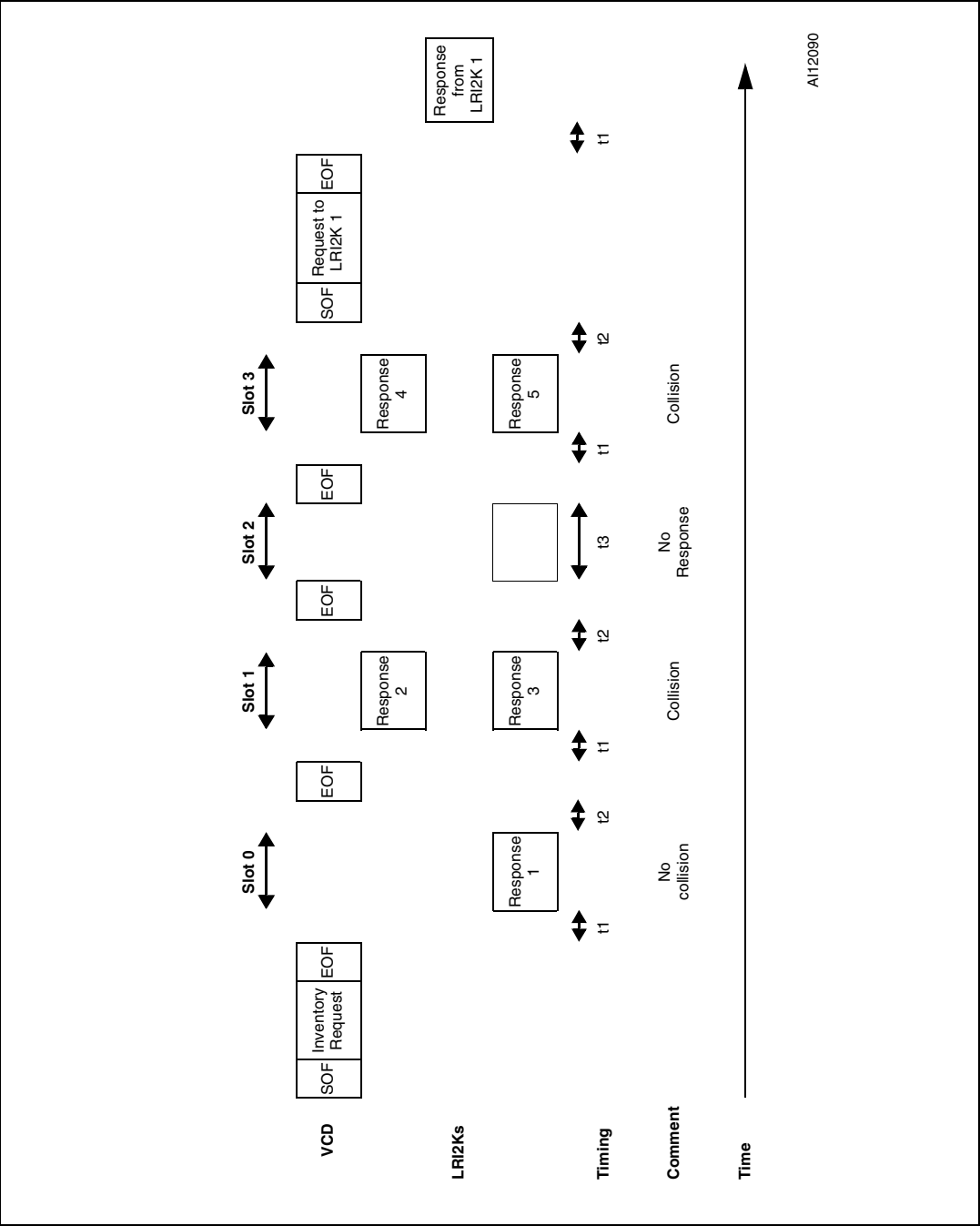
*Figure 40* summarizes the main possible cases that can occur during an anticollision sequence when the slot number is 16.

The different steps are:

- The VCD sends an Inventory request, in a frame terminated by an EOF. The number of slots is 16.
- LRI2K 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 LRI2Ks, LRI2K 2 and LRI2K 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 LRI2K transmits a response. Therefore the VCD does not detect any LRI2K SOF and decides to switch to the next slot by sending an EOF.
- In slot 3, there is another collision caused by responses from LRI2K 4 and LRI2K 5
- The VCD then decides to send a request (for instance a Read Block) to LRI2K 1 whose UID has already been correctly received.
- All LRI2Ks detect an SOF and exit the anticollision sequence. They process this request and since the request is addressed to LRI2K 1, only LRI2K 1 transmits a response.
- All LRI2Ks 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 LRI2K 1.

Figure 40. Description of a possible anticollision sequence



AI12090

## 18 Inventory Initiated command

The LRI2K 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 LRI2Ks to increase their response data rate by a factor of 2, up to 53kbit/s.

## 19 Timing definition

### 19.1 $t_1$ : LRI2K response delay

Upon detection of the rising edge of the EOF received from the VCD, the LRI2K 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 19](#). The EOF is defined in [Figure 11 on page 18](#).

### 19.2 $t_2$ : 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 LRI2K responses have been received during an Inventory command. It starts from the reception of the EOF from the LRI2Ks.

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 LRI2K.

$t_2$  is also the time after which the VCD may send a new request to the LRI2K as described in [Table 37: LRI2K protocol timing](#).

Values of  $t_2$  are given in [Table 19](#).

### 19.3 $t_3$ : VCD new request delay in the absence of a response from the LRI2K

$t_3$  is the time after which the VCD may send an EOF to switch to the next slot when no LRI2K 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 LRI2K.

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 LRI2K nominal response time (which depends on the LRI2K data rate and subcarrier modulation mode) before sending a new EOF.

**Table 19. Timing values<sup>(1)</sup>**

|       | Minimum (min) values             | Nominal (nom) values | Maximum (max) values |
|-------|----------------------------------|----------------------|----------------------|
| $t_1$ | 318.6 $\mu$ s                    | 320.9 $\mu$ s        | 323.3 $\mu$ s        |
| $t_2$ | 309.2 $\mu$ 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 LRI2K to transmit an SOF to the VCD.  $t_{SOF}$  depends on the current data rate: High data rate or Low data rate.

## 20 Commands codes

The LRI2K supports the commands described in this section. Their codes are given in [Table 20](#).

**Table 20. Command codes**

| Command code<br>standard | Function                              |
|--------------------------|---------------------------------------|
| 01h                      | Inventory                             |
| 02h                      | Stay Quiet                            |
| 20h                      | Read Single Block                     |
| 21h                      | Write Single Block                    |
| 22h                      | Lock 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                       |
| 2Ch                      | Get Multiple Block<br>Security Status |

| Command code<br>custom | Function                 |
|------------------------|--------------------------|
| A6h                    | Kill                     |
| B1h                    | Write Kill               |
| B2h                    | Lock Kill                |
| C0h                    | Fast Read Single Block   |
| C1h                    | Fast Inventory Initiated |
| C2h                    | Fast Initiate            |
| C3h                    | Fast Read Multiple Block |
| D1h                    | Inventory Initiated      |
| D2h                    | Initiate                 |

## 20.1 Inventory

When receiving the Inventory request, the LRI2K runs the anticollision sequence. The Inventory\_flag is set to 1. The meaning of flags 5 to 8 is shown in [Table 13: Request flags 5 to 8 when bit 3 = 1](#).

The request contains:

- the flags,
- the Inventory command code (see [Table 20: Command codes](#))
- the AFI if the AFI flag is set
- the mask length
- the mask value
- the CRC

The LRI2K does not generate any answer in case of error.

**Table 21. 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 22. 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 LRI2K 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 LRI2K to transmit an SOF to the VCD
- $t_{NRT}$  is the nominal response time of the LRI2K

$t_{NRT}$  and  $t_{SOF}$  are dependent on the LRI2K-to-VCD data rate and subcarrier modulation mode.

20.2 Stay Quiet

On receiving the Stay Quiet command, the LRI2K enters the Quiet state 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 LRI2K does not process any request if the Inventory\_flag is set,
- the LRI2K processes any Addressed request

The LRI2K 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 23. 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 the Addressed mode (Select\_flag is reset to 0 and Address\_flag is set to 1).

Figure 41. Stay Quiet frame exchange between VCD and LRI2K





## 20.3 Read Single Block

On receiving the Read Single Block command, the LRI2K reads the requested block and sends back its 32 bits value in the response. The Option\_flag is supported.

**Table 24. Read Single Block request format**

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

Request parameters:

- Option\_flag
- UID (Optional)
- Block number

**Table 25. Read Single Block response format when Error\_flag is NOT set**

| Response SOF | Response_flags | Block locking status | Data    | CRC16   | Response EOF |
|--------------|----------------|----------------------|---------|---------|--------------|
|              | 8 bits         | 8 bits               | 32 bits | 16 bits |              |

Response parameter:

- Block Locking Status if Option\_flag is set (see [Table 26: Block Locking status](#))
- 4 bytes of block data

**Table 26. Block Locking status**

| b <sub>7</sub> | b <sub>6</sub> | b <sub>5</sub> | b <sub>4</sub> | b <sub>3</sub> | b <sub>2</sub> | b <sub>1</sub> | b <sub>0</sub>                                         |
|----------------|----------------|----------------|----------------|----------------|----------------|----------------|--------------------------------------------------------|
| all 0          |                |                |                |                |                |                | 0: Current Block not locked<br>1: Current Block locked |

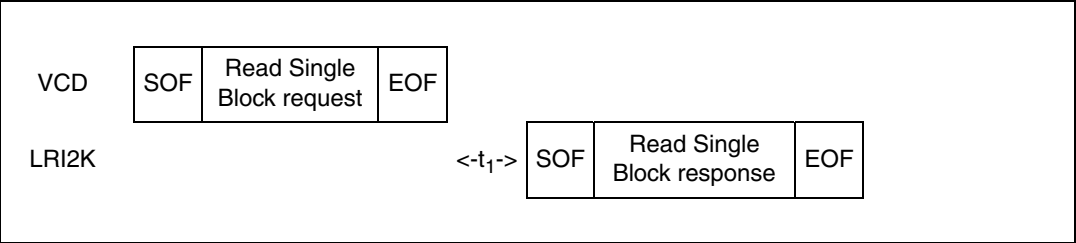
**Table 27. Read Single Block response format when Error\_flag is set**

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

Response parameter:

- Error code as Error\_flag is set:
  - 0Fh: other error
  - 10h: block address not available

Figure 42. READ Single Block frame exchange between VCD and LRI2K



## 20.4 Write Single Block

On receiving the Write Single Block Command, the LRI2K writes the data contained in the request to the requested block and reports whether the write operation was successful in the response. The Option\_flag is supported.

During the write cycle  $t_W$ , there should be no modulation (neither 100% nor 10%). Otherwise, the LRI2K may not program correctly the data into the memory. The  $t_W$  time is equal to  $t_{1nom} + 18 \times 302\mu s$ .

**Table 28. Write Single Block request format**

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

Request parameters:

- UID (Optional)
- Block number
- Data

**Table 29. Write Single Block response format when Error\_flag is NOT set**

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

Response parameter:

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

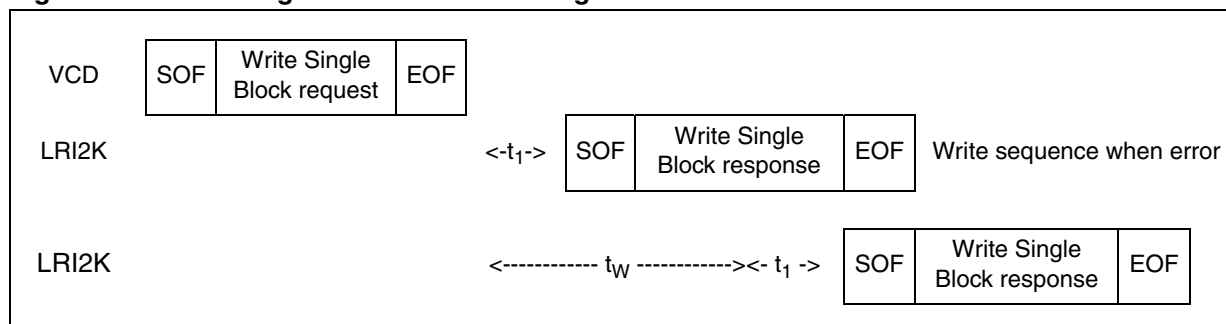
**Table 30. Write Single Block response format when Error\_flag is set**

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

Response parameter:

- Error code as Error\_flag is set:
  - 10h: block address not available
  - 12h: block is locked
  - 13h: block not programmed

**Figure 43. Write Single Block frame exchange between VCD and LRI2K**



# 20.5 Lock Block

On receiving the Lock Block command, the LRI2K permanently locks the selected block. The Option\_flag is supported.

During the write cycle  $t_W$ , there should be no modulation (neither 100% nor 10%). Otherwise, the LRI2K may not lock correctly the memory block. The  $t_W$  time is equal to  $t_{1nom} + 18 \times 302\mu s$ .

**Table 31. Lock Single Block request format**

| Request SOF | Request_flags | Lock Block | UID     | Block number | CRC16   | Request EOF |
|-------------|---------------|------------|---------|--------------|---------|-------------|
|             | 8 bits        | 22h        | 64 bits | 8 bits       | 16 bits |             |

- Request parameters:
- (Optional) UID
  - Block number

**Table 32. Lock Block response format when Error\_flag is NOT set**

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

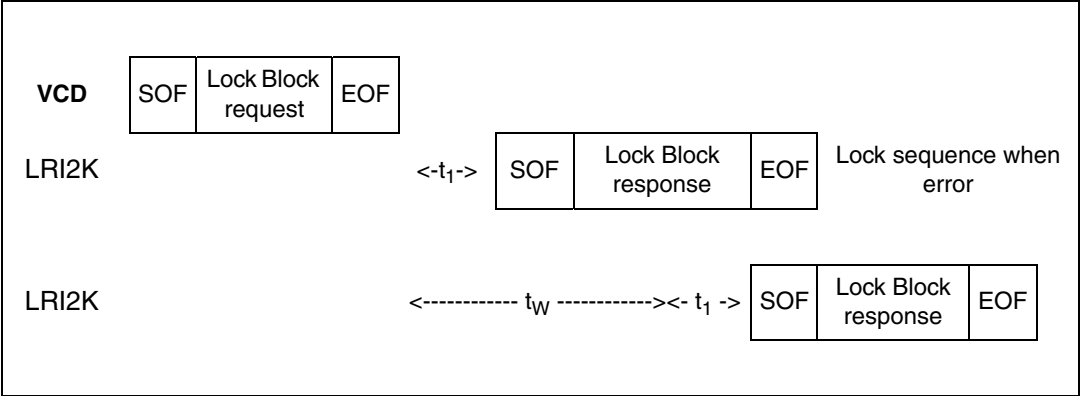
- Response parameter:
- No parameter.

**Table 33. Lock Block response format when Error\_flag is set**

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

- Response parameter:
- Error code as Error\_flag is set:
    - 10h: block address not available
    - 11h: block is locked
    - 14h: block not locked

**Figure 44. Lock Block frame exchange between VCD and LRI2K**



## 20.6 Read Multiple Block

When receiving the Read Multiple Block command, the LRI2K reads the selected blocks and sends back their value in multiples of 32 bits in the response. The blocks are numbered from '00 to '3F' 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 will be read. The maximum number of blocks is fixed at 64. During Sequential Block Read, when the block address reaches 64, it rolls over to 0. The Option\_flag is supported.

**Table 34. Read Multiple Block request format**

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

Request parameters:

- Option\_flag
- UID (Optional)
- First block number
- Number of blocks

**Table 35. Read Multiple Block response format when Error\_flag is NOT set**

| Response SOF | Response flags | Block Locking Status  | Data                   | CRC16   | Response EOF |
|--------------|----------------|-----------------------|------------------------|---------|--------------|
|              | 8 bits         | 8 bits <sup>(1)</sup> | 32 bits <sup>(1)</sup> | 16 bits |              |

1. Repeated as needed.

Response parameter:

- Block Locking Status if Option\_flag is set (see [Table 36: Block Locking status](#))
- N blocks of data

**Table 36. Block Locking status**

| b <sub>7</sub> | b <sub>6</sub> | b <sub>5</sub> | b <sub>4</sub> | b <sub>3</sub> | b <sub>2</sub> | b <sub>1</sub> | b <sub>0</sub>                                         |
|----------------|----------------|----------------|----------------|----------------|----------------|----------------|--------------------------------------------------------|
| All 0          |                |                |                |                |                |                | 0: Current Block not locked<br>1: Current Block locked |

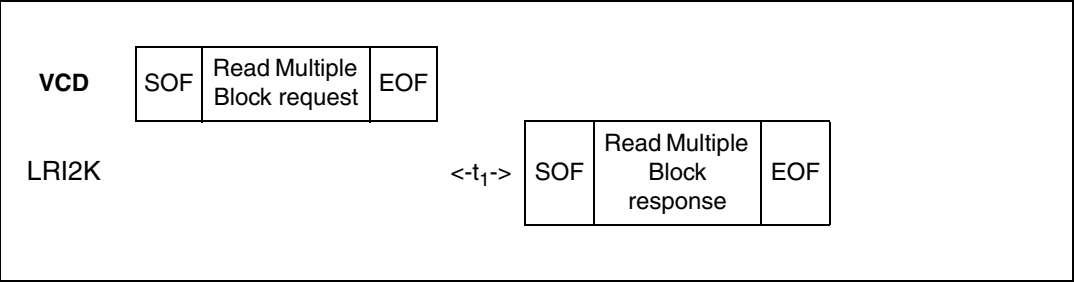
**Table 37. Read Multiple Block response format when Error\_flag is set**

| 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 45. Read Multiple Block frame exchange between VCD and LRI2K



## 20.7 Select

When receiving the Select command:

- if the UID is equal to its own UID, the LRI2K enters or stays in the Selected state and sends a response.
- if the UID does not match its own, the selected LRI2K returns to the Ready state and does not send a response.

The LRI2K answers an error code only if the UID is equal to its own UID. If not, no response is generated.

**Table 38. Select request format**

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

Request parameter:

- UID

**Table 39. 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.

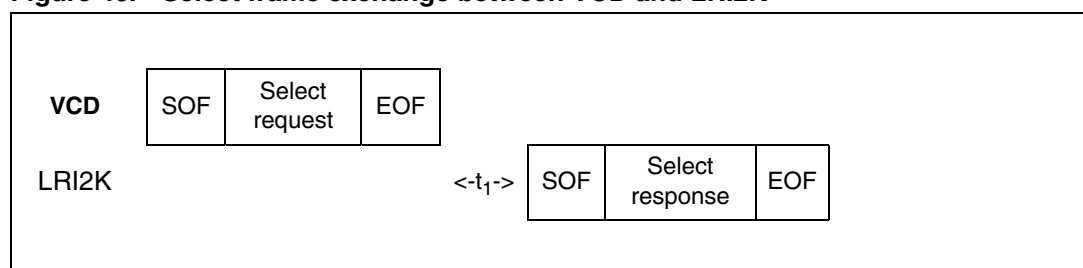
**Table 40. Select response format when Error\_flag is set**

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

Response parameter:

- Error code as Error\_flag is set:
  - 0Fh: other error

**Figure 46. Select frame exchange between VCD and LRI2K**



# 20.8 Reset to Ready

On receiving a Reset to Ready command, the LRI2K returns to the Ready state. In the Addressed mode, the LRI2K answers an error code only if the UID is equal to its own UID. If not, no response is generated.

**Table 41. Reset to Ready request format**

| Request SOF | Request flags | Reset to Ready | UID     | CRC16   | Request EOF |
|-------------|---------------|----------------|---------|---------|-------------|
|             | 8 bits        | 26h            | 64 bits | 16 bits |             |

Request parameter:

- UID (Optional)

**Table 42. 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.

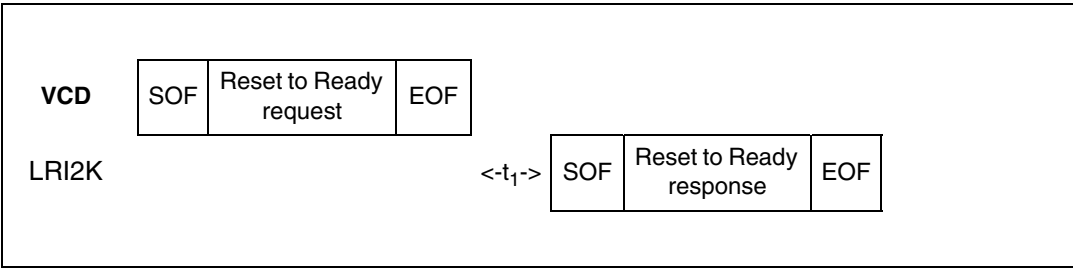
**Table 43. Reset to ready response format when Error\_flag is set**

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

Response parameter:

- Error code as Error\_flag is set:
  - 0Fh: other error

**Figure 47. Reset to Ready frame exchange between VCD and LRI2K**





## 20.9 Write AFI

On receiving the Write AFI request, the LRI2K writes the AFI byte value into its memory. The Option\_flag is supported.

During the write cycle  $t_W$ , there should be no modulation (neither 100% nor 10%). Otherwise, the LRI2K may not write correctly the AFI value into the memory. The  $t_W$  time is equal to  $t_{1nom} + 18 \times 302\mu s$ .

**Table 44. Write AFI request format**

| Request SOF | Request_flags | Write AFI | UID     | AFI    | CRC16   | Request EOF |
|-------------|---------------|-----------|---------|--------|---------|-------------|
|             | 8 bits        | 27h       | 64 bits | 8 bits | 16 bits |             |

Request parameters:

- UID (Optional)
- AFI

**Table 45. 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.

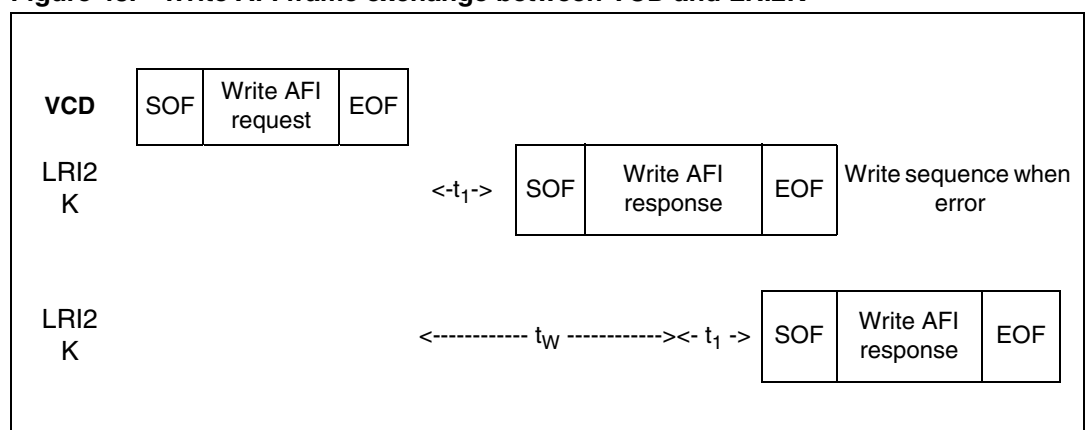
**Table 46. Write AFI response format when Error\_flag is set**

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

Response parameter:

- Error code as Error\_flag is set:
  - 12h: block is locked
  - 13h: block not programmed

**Figure 48. Write AFI frame exchange between VCD and LRI2K**



## 20.10 Lock AFI

On receiving the Lock AFI request, the LRI2K locks the AFI value permanently. The Option\_flag is supported.

During the write cycle  $t_W$ , there should be no modulation (neither 100% nor 10%). Otherwise, the LRI2K may not Lock correctly the AFI value in memory. The  $t_W$  time is equal to  $t_{1nom} + 18 \times 302 \mu s$ .

**Table 47. Lock AFI request format**

| Request SOF | Request_flags | Lock AFI | UID     | CRC16   | Request EOF |
|-------------|---------------|----------|---------|---------|-------------|
|             | 8 bits        | 28h      | 64 bits | 16 bits |             |

Request parameter:

- UID (Optional)

**Table 48. 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.

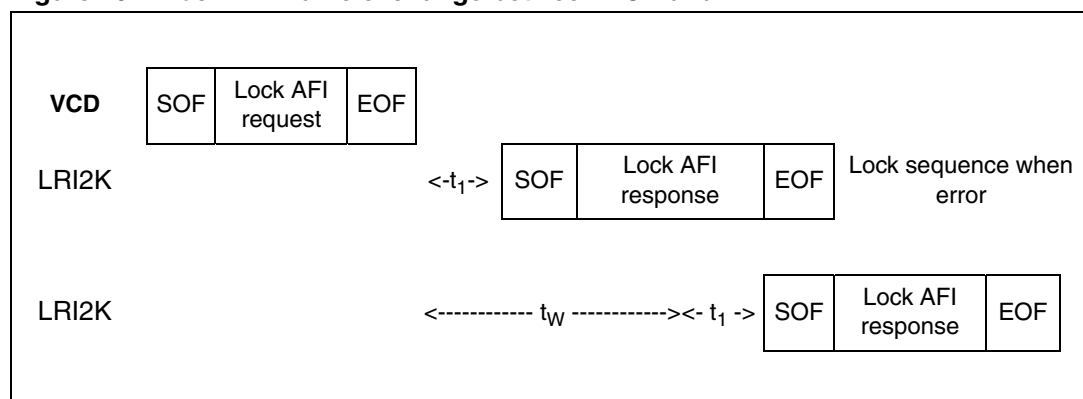
**Table 49. Lock AFI response format when Error\_flag is set**

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

Response parameter:

- Error code as Error\_flag is set:
  - 11h: block is locked
  - 14h: block not locked

**Figure 49. Lock AFI frame exchange between VCD and LRI2K**



## 20.11 Write DSFID

On receiving the Write DSFID request, the LRI2K writes the DSFID byte value into its memory. The Option\_flag is supported.

During the write cycle  $t_W$ , there should be no modulation (neither 100% nor 10%). Otherwise, the LRI2K may not write correctly the DSFID value in memory. The  $t_W$  time is equal to  $t_{1nom} + 18 \times 302\mu s$ .

**Table 50. Write DSFID request format**

| Request SOF | Request_flags | Write DSFID | UID     | DSFID  | CRC16   | Request EOF |
|-------------|---------------|-------------|---------|--------|---------|-------------|
|             | 8 bits        | 29h         | 64 bits | 8 bits | 16 bits |             |

Request parameters:

- UID (Optional)
- DSFID

**Table 51. 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.

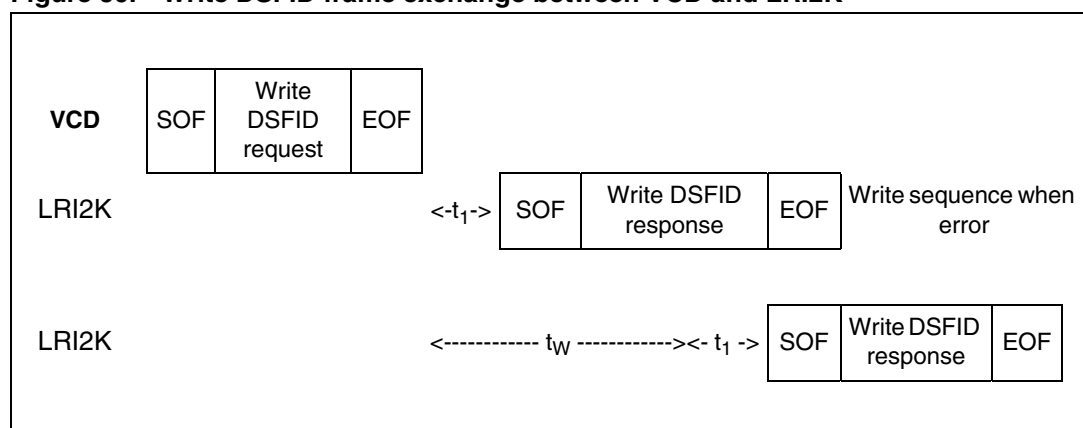
**Table 52. Write DSFID response format when Error\_flag is set**

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

Response parameter:

- Error code as Error\_flag is set:
  - 12h: block is locked
  - 13h: block not programmed

**Figure 50. Write DSFID frame exchange between VCD and LRI2K**



## 20.12 Lock DSFID

On receiving the Lock DSFID request, the LRI2K locks the DSFID value permanently. The Option\_flag is supported.

During the write cycle  $t_W$ , there should be no modulation (neither 100% nor 10%). Otherwise, the LRI2K may not lock correctly the DSFID value in memory. The  $t_W$  time is equal to  $t_{1nom} + 18 \times 302\mu s$ .

**Table 53. Lock DSFID request format**

| Request SOF | Request_flags | Lock DSFID | UID     | CRC16   | Request EOF |
|-------------|---------------|------------|---------|---------|-------------|
|             | 8 bits        | 2Ah        | 64 bits | 16 bits |             |

Request parameter:

- UID (Optional)

**Table 54. 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.

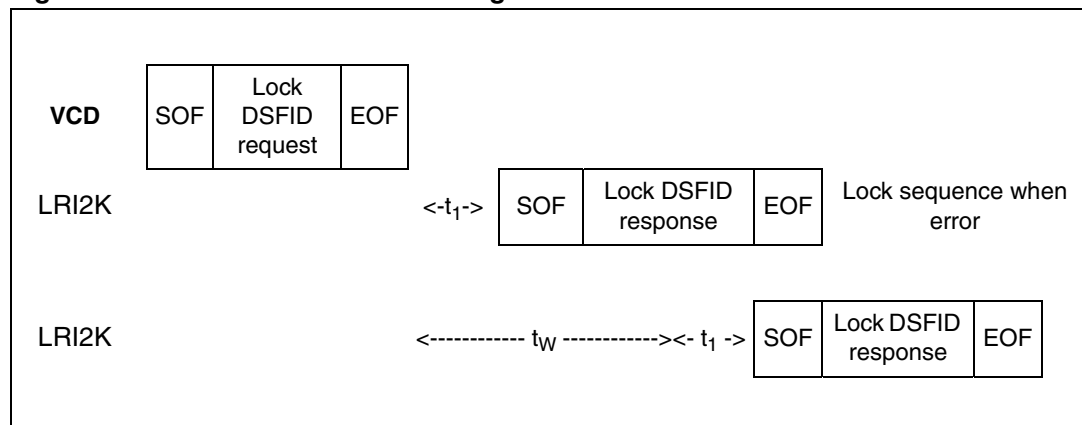
**Table 55. Lock DSFID response format when Error\_flag is set**

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

Response parameter:

- Error code as Error\_flag is set:
  - 11h: block is locked
  - 14h: block not locked

**Figure 51. Lock DSFID frame exchange between VCD and LRI2K**



## 20.13 Get System Info

When receiving the Get System Info command, the LRI2K 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.

**Table 56. Get System Info request format**

| Request SOF | Request flags | Get System Info | UID     | CRC16   | Request EOF |
|-------------|---------------|-----------------|---------|---------|-------------|
|             | 8 bits        | 2Bh             | 64 bits | 16 bits |             |

Request parameter:

- UID (Optional)

**Table 57. 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 | 033Fh       | 001000xx <sub>b</sub> | 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 LRI2K provides 64 blocks (3Fh) of 4 bytes (03h).
- IC Reference. Only the 6 MSBs are significant. The product code of the LRI2K is 00 1000<sub>b</sub>=8<sub>d</sub>

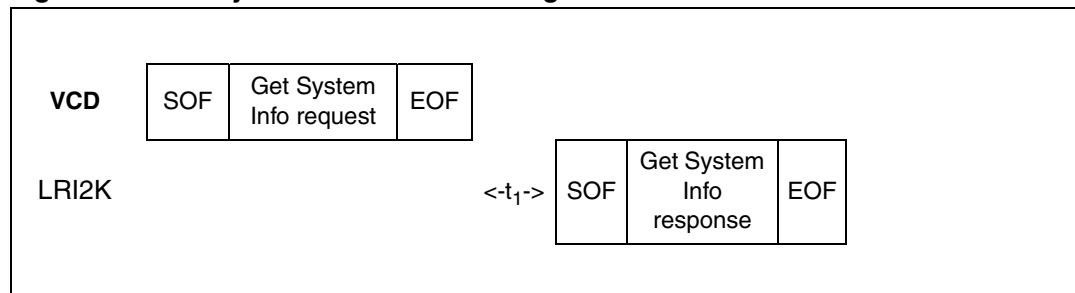
**Table 58. Get System Info response format when Error\_flag is set**

| 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 52. Get System Info frame exchange between VCD and LRI2K**



## 20.14 Get Multiple Block Security Status

When receiving the Get Multiple Block Security Status command, the LRI2K sends back the block security status. The blocks are numbered from '00' to '3F' 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.

**Table 59. Get Multiple Block Security Status request format**

| Request SOF | Request_flags | Get Multiple Block Security Status | UID     | First block number | Number of blocks | CRC16   | Request EOF |
|-------------|---------------|------------------------------------|---------|--------------------|------------------|---------|-------------|
|             | 8 bits        | 2Ch                                | 64 bits | 8 bits             | 8 bits           | 16 bits |             |

Request parameters:

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

**Table 60. Get Multiple Block Security Status response format when Error\_flag is NOT set**

| Response SOF | Response_flags | Block Locking Status  | CRC16   | Response EOF |
|--------------|----------------|-----------------------|---------|--------------|
|              | 8 bits         | 8 bits <sup>(1)</sup> | 16 bits |              |

1. Repeated as needed.

Response parameters:

- Block Locking Status (see [Table 61: Block Locking status](#))
- N block of data

**Table 61. Block Locking status**

| b <sub>7</sub> | b <sub>6</sub> | b <sub>5</sub> | b <sub>4</sub> | b <sub>3</sub> | b <sub>2</sub> | b <sub>1</sub> | b <sub>0</sub>                                         |
|----------------|----------------|----------------|----------------|----------------|----------------|----------------|--------------------------------------------------------|
| All 0          |                |                |                |                |                |                | 0: Current block not locked<br>1: Current block locked |

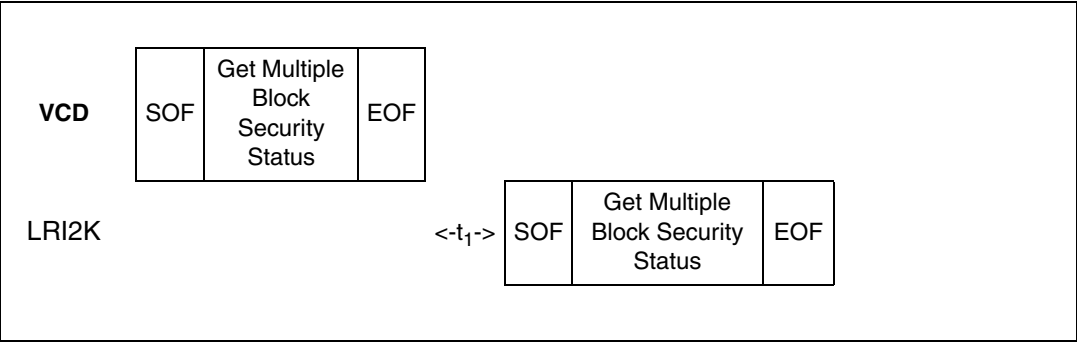
**Table 62. Get Multiple Block Security Status response format when Error\_flag is set**

| 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: Option not supported
  - 0Fh: other error

Figure 53. Get Multiple Block Security Status frame exchange between VCD and LRI2K



## 20.15 Kill

On receiving the Kill command, in the Addressed mode only, the LRI2K compares the kill code with the data contained in the request and reports whether the operation was successful in the response. The Option\_flag is supported. If the command is received in the Non Addressed or the Selected mode, the LRI2K returns an error response.

During the comparison cycle equal to  $t_W$ , there should be no modulation (neither 100% nor 10%). Otherwise, the LRI2K may not match the kill code correctly. The  $t_W$  time is equal to  $t_{1nom} + 18 \times 302\mu s$ . After a successful Kill command, the LRI2K is deactivated and does not interpret any other command.

**Table 63. Kill request format**

| Request SOF | Request_flags | Kill | IC Mfg code | UID     | Kill access | Kill code | CRC16   | Request EOF |
|-------------|---------------|------|-------------|---------|-------------|-----------|---------|-------------|
|             | 8 bits        | A6h  | 02h         | 64 bits | 00h         | 32 bits   | 16 bits |             |

Request parameters:

- UID (Optional)
- Kill Code

**Table 64. Kill 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

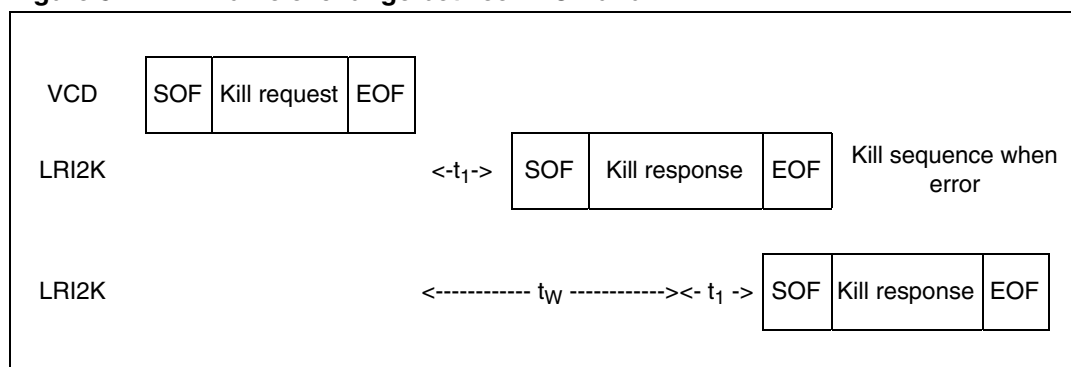
**Table 65. Kill response format when Error\_flag is set**

| 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
  - 14h: block not locked

**Figure 54. Kill frame exchange between VCD and LRI2K**





## 20.16 Write Kill

On receiving the Write Kill command, the LRI2K writes the kill code with the data contained in the request and reports whether the operation was successful in the response. The Option\_flag is supported. After a successful write, the kill code must be locked by a Lock Kill command to activate the protection.

During the write cycle  $t_W$ , there should be no modulation (neither 100% nor 10%). Otherwise, the LRI2K may not correctly program the data to the memory. The  $t_W$  time is equal to  $t_{1nom} + 18 \times 302 \mu s$ .

**Table 66. Write Kill request format**

| Request SOF | Request flags | Write Kill | IC Mfg code | UID     | Kill access | Kill code | CRC16   | Request EOF |
|-------------|---------------|------------|-------------|---------|-------------|-----------|---------|-------------|
|             | 8 bits        | B1h        | 02h         | 64 bits | 00h         | 32 bits   | 16 bits |             |

Request parameters:

- UID (Optional)
- Kill Address (00h = Kill, other = Error)
- Data

**Table 67. Write Kill response format when Error\_flag is NOT set**

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

No parameter. The response is send back after the write cycle.

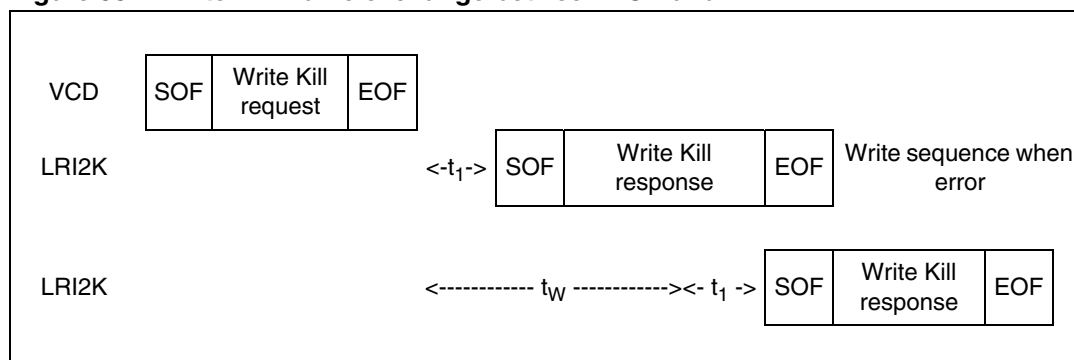
**Table 68. Write Kill response format when Error\_flag is set**

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

Response parameter:

- Error code as Error\_flag is set:
  - 10h: block address not available
  - 12h: block is locked
  - 13h: block not programmed

**Figure 55. Write Kill frame exchange between VCD and LRI2K**



## 20.17 Lock Kill

On receiving the Lock Kill command, the LRI2K locks the Kill code permanently. The Option\_flag is supported. RFU bit 8 of the request flag must be set to '1'.

During the write cycle  $t_W$ , there should be no modulation (neither 100% nor 10%). Otherwise, the LRI2K may not lock the memory block correctly. The  $t_W$  time is equal to  $t_{1nom} + 18 \times 302 \mu s$ .

**Table 69. Lock Kill request format**

| Request SOF | Request_flags | Lock Kill | IC Mfg code | UID     | Kill access | Protect Status | CRC16   | Request EOF |
|-------------|---------------|-----------|-------------|---------|-------------|----------------|---------|-------------|
|             | 8 bits        | B2h       | 02h         | 64 bits | 00f         | 8 bits         | 16 bits |             |

Request parameters:

- (Optional) UID
- Kill Address (bit 8 = '1': 00h = KILL, other = Error)
- Protect status (see table below)

|                |                |                |                |                |                |                |                |
|----------------|----------------|----------------|----------------|----------------|----------------|----------------|----------------|
| b <sub>7</sub> | b <sub>6</sub> | b <sub>5</sub> | b <sub>4</sub> | b <sub>3</sub> | b <sub>2</sub> | b <sub>1</sub> | b <sub>0</sub> |
| 0              | 0              | 0              | 0              | 0              | 0              | 0              | 1              |

**Table 70. Lock Kill response format when Error\_flag is NOT set**

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

Response parameter:

- No parameter.

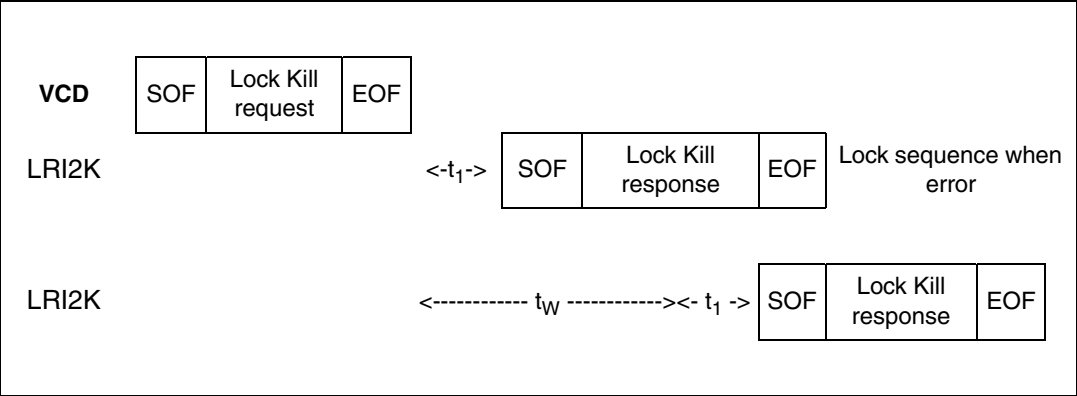
**Table 71. Lock Kill response format when Error\_flag is set**

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

Response parameter:

- Error code as Error\_flag is set:
  - 10h: block address not available
  - 11h: block is locked
  - 14h: block not locked

Figure 56. Lock Kill frame exchange between VCD and LRI2K



## 20.18 Fast Read Single Block

On receiving the Fast Read Single Block command, the LRI2K 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.

**Table 72. Fast Read Single Block request format**

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

Request parameters:

- Option\_flag
- UID (Optional)
- Block number

**Table 73. Fast Read Single Block response format when Error\_flag is NOT set**

| Response SOF | Response flags | Block locking status | Data    | CRC16   | Response EOF |
|--------------|----------------|----------------------|---------|---------|--------------|
|              | 8 bits         | 8 bits               | 32 bits | 16 bits |              |

Response parameter:

- Block Locking Status if Option\_flag is set
- 4 bytes of Block Data

**Table 74. Block Locking status**

| b <sub>7</sub> | b <sub>6</sub> | b <sub>5</sub> | b <sub>4</sub> | b <sub>3</sub> | b <sub>2</sub> | b <sub>1</sub> | b <sub>0</sub>                                         |
|----------------|----------------|----------------|----------------|----------------|----------------|----------------|--------------------------------------------------------|
| All 0          |                |                |                |                |                |                | 0: Current Block not locked<br>1: Current Block locked |

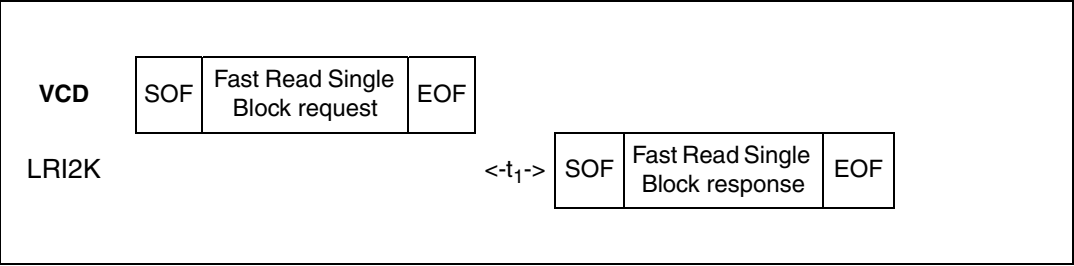
**Table 75. Fast Read Single Block response format when Error\_flag is set**

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

Response parameter:

- Error code as Error\_flag is set:
  - 0Fh: other error
  - 10h: block address not available

Figure 57. Fast Read Single Block frame exchange between VCD and LRI2K



## 20.19 Fast Inventory Initiated

Before receiving the Fast Inventory Initiated command, the LRI2K must have received an Initiate or a Fast Initiate command in order to set the Initiate\_flag. If not, the LRI2K does not answer to the Fast Inventory Initiated command.

On receiving the Fast Inventory Initiated request, the LRI2K runs the anticollision sequence. The Inventory\_flag must be set to 1. The Meaning of Flags 5 to 8 is shown in [Table 13: 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 LRI2K does not generate any answer if an error occurs.

**Table 76. 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 77. Fast Inventory Initiated response format**

| Response SOF | Response Flags | DSFID | UID     | CRC16   | Response EOF |
|--------------|----------------|-------|---------|---------|--------------|
|              | 8 bits         | 00h   | 64 bits | 16 bits |              |

During an Inventory process, if the VCD does not receive an RF LRI2K 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 LRI2K to transmit an SOF to the VCD
- $t_{NRT}$  is the nominal response time of the LRI2K

$t_{NRT}$  and  $t_{SOF}$  are dependent on the LRI2K-to-VCD data rate and subcarrier modulation mode.

20.20 Fast Initiate

On receiving the Fast Initiate command, the LRI2K sets the internal Initiate\_flag and sends back a response. 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 LRI2K does not generate any answer. The Initiate\_flag is reset after a power off of the LRI2K. The data rate of the response is multiplied by 2.

The request contains:

- No data

Table 78. Fast Initiate request format

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

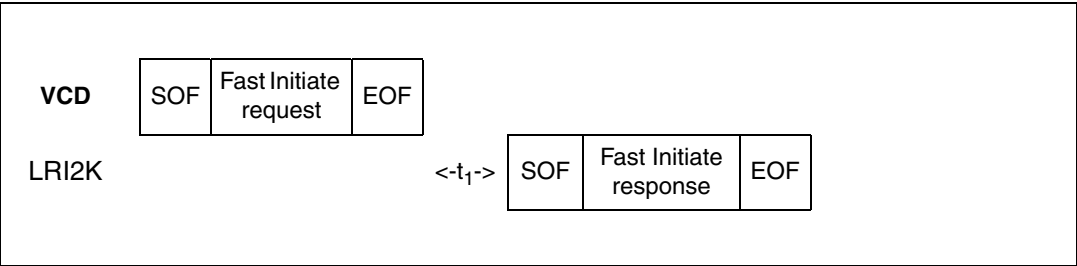
The response contains:

- the flags
- the Unique ID

Table 79. Fast Initiate response format

| Response SOF | Response flags | DSFID | UID     | CRC16   | Response EOF |
|--------------|----------------|-------|---------|---------|--------------|
|              | 8 bits         | 00h   | 64 bits | 16 bits |              |

Figure 58. Fast Initiate frame exchange between VCD and LRI2K



## 20.21 Fast Read Multiple Block

On receiving the Fast Read Multiple Block command, the LRI2K reads the requested blocks and sends back their value in multiples of 32 bits in the response. The blocks are numbered from '00' to '3F' in the request and the value is minus one (–1) in the field. For example, a value 06h in the “number of blocks” field causes the LRI2K to read 7 blocks. The maximum number of blocks is fixed at 64. During Sequential Block Read, when the block address reaches 64, it rolls over to 0. The Option\_flag is supported. The data rate of the response is multiplied by 2.

**Table 80. Fast Read Multiple Block request format**

| Request SOF | Request flags | Fast Read Multiple Block | ICMfg code | UID     | First block number | Number of blocks | CRC16   | Request EOF |
|-------------|---------------|--------------------------|------------|---------|--------------------|------------------|---------|-------------|
|             | 8 bits        | C3h                      | 02h        | 64 bits | 8 bits             | 8 bits           | 16 bits |             |

Request parameters:

- Option\_flag
- UID (Optional)
- First block number
- Number of blocks

**Table 81. Fast Read Multiple Block response format when Error\_flag is NOT set**

| Response SOF | Response flags | Block Locking Status  | Data                   | CRC16   | Response EOF |
|--------------|----------------|-----------------------|------------------------|---------|--------------|
|              | 8 bits         | 8 bits <sup>(1)</sup> | 32 bits <sup>(1)</sup> | 16 bits |              |

1. Repeated as needed.

Response parameters:

- Block Locking Status if Option\_flag is set
- N block of data

**Table 82. Block Locking status if Option\_flag is set**

| b <sub>7</sub> | b <sub>6</sub> | b <sub>5</sub> | b <sub>4</sub> | b <sub>3</sub> | b <sub>2</sub> | b <sub>1</sub> | b <sub>0</sub>                                         |
|----------------|----------------|----------------|----------------|----------------|----------------|----------------|--------------------------------------------------------|
| All 0          |                |                |                |                |                |                | 0: Current block not locked<br>1: Current block locked |

**Table 83. Fast Read Multiple Block response format when Error\_flag is set**

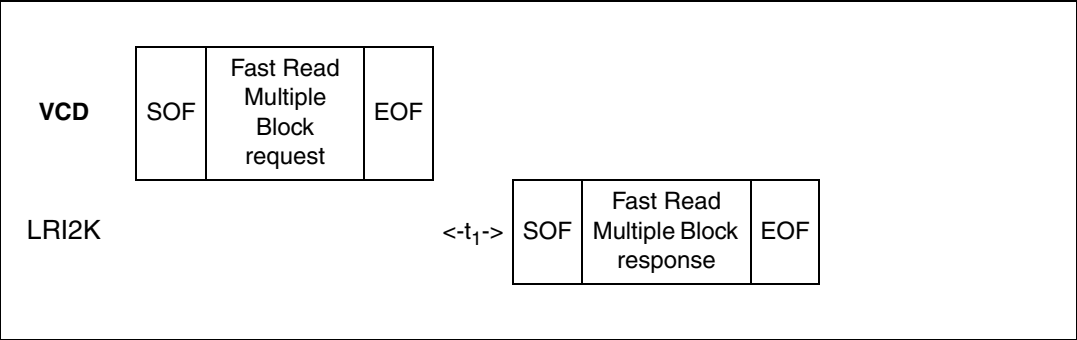
| Response SOF | Response flags | Error code | CRC16   | Response EOF |
|--------------|----------------|------------|---------|--------------|
|              | 8 bits         | 8 bits     | 16 bits |              |

Response parameter:

- Error code as Error\_flag is set:
  - 0Fh: other error
  - 10h: block address not available



Figure 59. Fast Read Multiple Block frame exchange between VCD and LRI2K



## 20.22 Inventory Initiated

Before receiving the Inventory Initiated command, the LRI2K must have received an Initiate or a Fast Initiate command in order to set the Initiate\_ flag. If not, the LRI2K does not answer to the Inventory Initiated command.

On receiving the Inventory Initiated request, the LRI2K runs the anticollision sequence. The Inventory\_flag must be set to 1. The Meaning of Flags 5 to 8 is given in [Table 13: 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 LRI2K does not generate any answer if an error occurs.

**Table 84. Inventory Initiated request format**

| Request SOF | Request Flags | Inventory Initiated | ICMfg code | Optiona I 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 85. Inventory Initiated response format**

| Response SOF | Response Flags | DSFID | UID     | CRC16   | Response EOF |
|--------------|----------------|-------|---------|---------|--------------|
|              | 8 bits         | 00h   | 64 bits | 16 bits |              |

During an Inventory process, if the VCD does not receive an RF LRI2K 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 LRI2K to transmit an SOF to the VCD
- $t_{NRT}$  is the nominal response time of the LRI2K

$t_{NRT}$  and  $t_{SOF}$  are dependent on the LRI2K-to-VCD data rate and subcarrier modulation mode.

20.23 Initiate

On receiving the Initiate command, the LRI2K sets the internal Initiate\_flag and sends back a response. 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 LRI2K does not generate any answer. The Initiate\_flag is reset after a power off of the LRI2K.

The request contains:

- No data

Table 86. Initiate request format

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

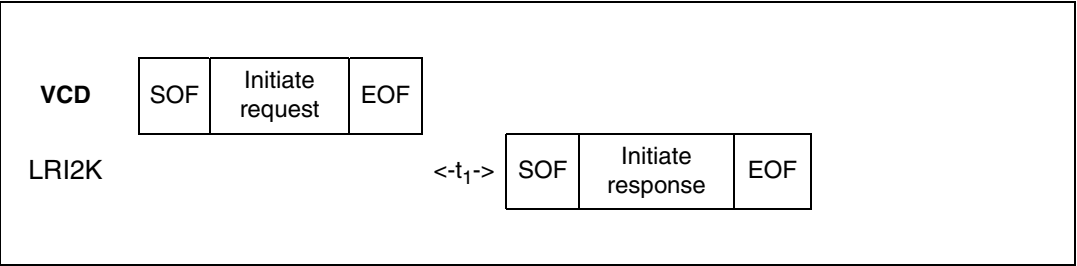
The response contain:

- the flags
- the Unique ID

Table 87. Initiate Initiated response format

| Response SOF | Response Flags | DSFID | UID     | CRC16   | Response EOF |
|--------------|----------------|-------|---------|---------|--------------|
|              | 8 bits         | 00h   | 64 bits | 16 bits |              |

Figure 60. Initiate frame exchange between VCD and LRI2K



## 21 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 88. Absolute maximum ratings**

| Symbol    | Parameter                                      |                                       | Min.  | Max. | Unit   |
|-----------|------------------------------------------------|---------------------------------------|-------|------|--------|
| $T_{STG}$ | Storage temperature                            | UFDFPN8                               | –65   | 150  | °C     |
|           |                                                | Wafer<br>(kept in its antistatic bag) | 15    | 25   |        |
| $t_{STG}$ | Storage time                                   | Wafer<br>(kept in its antistatic bag) |       | 23   | months |
| $I_{CC}$  | Supply current on AC0 / AC1                    |                                       | –20   | 20   | mA     |
| $V_{MAX}$ | Input voltage on AC0 / AC1                     |                                       | –7    | 7    | V      |
| $V_{ESD}$ | Electrostatic discharge voltage <sup>(1)</sup> | UFDFPN8 (HBM <sup>(2)</sup> )         | –1000 | 1000 | V      |
|           |                                                | UFDFPN8 (MM <sup>(3)</sup> )          | –100  | 100  | V      |

1. Mil. Std. 883 - Method 3015.

2. Human body model.

3. Machine model.

## 22 DC and AC parameters

This section summarizes the operating and measurement conditions, and the DC and AC characteristics of the device. 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 89. AC characteristics<sup>(1) (2)</sup>**

| Symbol             | Parameter                                          | Condition        | Min    | Typ    | Max    | Unit    |
|--------------------|----------------------------------------------------|------------------|--------|--------|--------|---------|
| $f_{CC}$           | External RF signal frequency                       |                  | 13.553 | 13.56  | 13.567 | MHz     |
| $MI_{CARRIER}$     | 10% carrier modulation index                       | $MI=(A-B)/(A+B)$ | 10     |        | 30     | %       |
| $t_{RFR}, t_{RFF}$ | 10% rise and fall time                             |                  | 0.5    |        | 3.0    | $\mu s$ |
| $t_{RFSBL}$        | 10% minimum pulse width for bit                    |                  | 7.1    |        | 9.44   | $\mu s$ |
| $MI_{CARRIER}$     | 100% carrier modulation index                      | $MI=(A-B)/(A+B)$ | 95     |        | 100    | %       |
| $t_{RFR}, t_{RFF}$ | 100% rise and fall time                            |                  | 0.5    |        | 3.5    | $\mu s$ |
| $t_{RFSBL}$        | 100% minimum pulse width for bit                   |                  | 7.1    |        | 9.44   | $\mu s$ |
| $t_{JIT}$          | Bit pulse jitter                                   |                  | -2     |        | +2     | $\mu s$ |
| $t_{MIN CD}$       | 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 LRI2K response                            | $4224/F_S$       | 318.6  | 320.9  | 323.3  | $\mu s$ |
| $t_2$              | Time between command                               | $4224/F_S$       | 309    | 311.5  | 314    | $\mu s$ |
| $t_W$              | Programming time                                   |                  |        |        | 5.8    | ms      |

1.  $T_A = -20$  to  $85^\circ C$ .

2. All timing measurements were performed 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 (LRI2K-W4)  
 Value of the coil: 4.3  $\mu H$   
 Tuning frequency: 13.8 MHz.

**Table 90. DC characteristics<sup>(1)</sup>**

| Symbol    | Parameter                      |       | Test conditions                 | Min. | Typ. | Max. | Unit          |
|-----------|--------------------------------|-------|---------------------------------|------|------|------|---------------|
| $V_{CC}$  | Regulated voltage              |       |                                 | 1.5  |      | 3.0  | V             |
| $V_{RET}$ | Retromodulated induced voltage |       | ISO10373-7                      | 10   |      |      | mV            |
| $I_{CC}$  | Supply current                 | Read  | $V_{CC} = 3.0\text{ V}$         |      |      | 50   | $\mu\text{A}$ |
|           |                                | Write | $V_{CC} = 3.0\text{ V}$         |      |      | 150  | $\mu\text{A}$ |
| $C_{TUN}$ | Internal tuning capacitor      |       | $f = 13.56\text{ MHz for W4/1}$ |      | 21   |      | pF            |
|           |                                |       | $f = 13.56\text{ MHz for W4/2}$ |      | 28.5 |      | pF            |
|           |                                |       | $f = 13.56\text{ MHz for W4/3}$ |      | 97   |      | pF            |
|           |                                |       | $f = 13.56\text{ MHz for W4/4}$ |      | 23.5 |      | pF            |

1.  $T_A = -20$  to  $85^\circ\text{C}$ .

**Table 91. Operating conditions**

| Symbol | Parameter                     | Min. | Max. | Unit             |
|--------|-------------------------------|------|------|------------------|
| $T_A$  | Ambient operating temperature | -20  | 85   | $^\circ\text{C}$ |

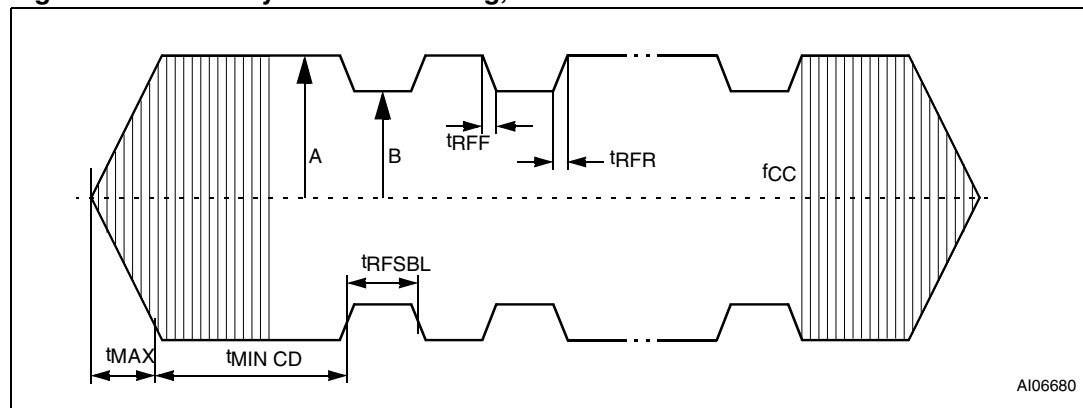
**Figure 61. LRI2K synchronous timing, transmit and receive**

Figure 61 shows an ASK modulated signal, from the VCD to the LRI2K. The test condition for the AC/DC parameters are:

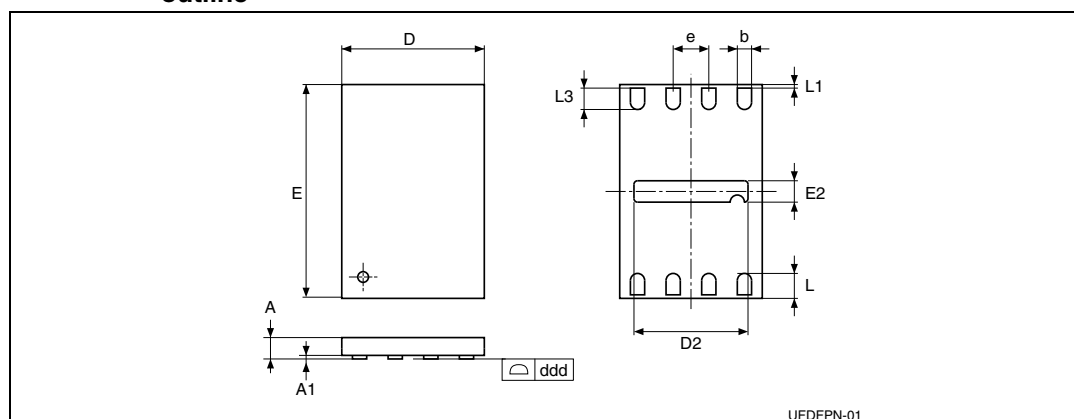
- Close coupling condition with tester antenna (1mm)
- LRI2K performance measured at the tag antenna

## 23 Package mechanical data

In order to meet environmental requirements, ST offers these devices in ECOPACK<sup>®</sup> packages. These packages have a Lead-free second-level interconnect. The category of second-level interconnect is marked on the package and on the inner box label, in compliance with JEDEC Standard JESD97.

The maximum ratings related to soldering conditions are also marked on the inner box label. ECOPACK is an ST trademark. ECOPACK specifications are available at: [www.st.com](http://www.st.com).

**Figure 62. UFDFPN8 - 8-lead ultra thin fine pitch dual flat package no lead (MLP) outline**



1. Drawing is not to scale.

**Table 92. UFDFPN8 - 8-lead ultra thin fine pitch dual flat package no lead (MLP) mechanical data**

| Symbol             | Millimeters |      |      | Inches <sup>(1)</sup> |        |        |
|--------------------|-------------|------|------|-----------------------|--------|--------|
|                    | 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 <sup>(2)</sup> | 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.

24 Part numbering

Table 93. Ordering information scheme

|                           |                                                                                                                                                                                                                                        |   |        |    |
|---------------------------|----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|---|--------|----|
| Example:                  | LRI2K                                                                                                                                                                                                                                  | - | W4 / 2 | GE |
| Device type               | LRI2K                                                                                                                                                                                                                                  |   |        |    |
| Package                   | W4 =180 µm ± 15 µm unsawn wafer<br>SBN18 = 180 µm ± 15 µm bumped and sawn wafer on 8-inch frame<br>MBTG = UFDFPN8 (MLP8), tape & reel packing, ECOPACK®, lead-free, RoHS compliant, Sb <sub>2</sub> O <sub>3</sub> -free and TBBA-free |   |        |    |
| Tuning capacitance        | 1 = 21 pF<br>2 = 28.5 pF<br>3 = 97 pF<br>4 = 23 pF                                                                                                                                                                                     |   |        |    |
| Customer code given by ST | GE = generic product<br>xx = customer code after personalization                                                                                                                                                                       |   |        |    |

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(LRI2K_UID); stores LRI2K_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 ; LRI2K is inventoried
            then
                store (LRI2K_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 LRI2K and from LRI2K to VCD.

**Table 94. 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 POLYNOMIAL8408h// x^16 + x^12 + x^5 + 1
#define PRESET_VALUEFFFFh
#define CHECK_VALUEF0B8h

#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, 91h, 39h};
    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 & 0001h)
            {
                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);
        }
    }
}

```

### B.3 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 LRI2K present only the LRI2K meeting the required application criteria.

It is programmed by the LRI2K issuer (the purchaser of the LRI2K). 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 95](#).

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 95. AFI coding<sup>(1)</sup>**

| AFI most significant nibble | AFI least significant nibble | Meaning<br>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 Y <sup>th</sup> subfamily of family X |                                 |
| '0'                         | 'Y'                          | Proprietary subfamily Y only                   |                                 |
| '1'                         | "0", 'Y'                     | Transport                                      | Mass transit, Bus, Airline etc. |
| '2'                         | "0", 'Y'                     | Financial                                      | IEP, Banking, Retail etc.       |
| '3'                         | "0", 'Y'                     | Identification                                 | Access Control etc.             |
| '4'                         | "0", 'Y'                     | Telecommunication                              | Public Telephony, GSM etc.      |
| '5'                         | '0', 'Y'                     | Medical                                        |                                 |
| '6'                         | "0", 'Y'                     | Multimedia                                     | Internet services etc.          |
| '7'                         | "0", 'Y'                     | Gaming                                         |                                 |
| '8'                         | "0", 'Y'                     | Data storage                                   | Portable Files etc.             |
| '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 96. Document revision history**

| Date        | Revision | Changes                                                                                                                                                                                                                                                                                                                                                  |
|-------------|----------|----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| 17-Feb-2006 | 1        | Initial release.                                                                                                                                                                                                                                                                                                                                         |
| 08-Feb-2007 | 2        | <a href="#">Figure 2: UFDFPN8 (MLP) connections</a> added.<br>Only bits set to '1' are programmed to the AFI and DSFID Registers (see <a href="#">Section 20.9: Write AFI</a> and <a href="#">Section 20.11: Write DSFID</a> .<br>$C_{TUN}$ typical value for W4/3 modified in <a href="#">Table 90: DC characteristics</a> .<br>Small text changes.     |
| 15-Jun-2007 | 3        | <a href="#">Section 20.9: Write AFI</a> and <a href="#">Section 20.11: Write DSFID</a> modified.                                                                                                                                                                                                                                                         |
| 20-Jul-2007 | 4        | Document status promoted from Preliminary Data to full Datasheet.<br>Small text changes.                                                                                                                                                                                                                                                                 |
| 31-Aug-2007 | 5        | 23.5 pF internal tuning capacitor ( $C_{TUN}$ ) value added (see <a href="#">Features on page 1</a> and <a href="#">Table 90: DC characteristics</a> .<br>$V_{ESD}$ max modified for MLP in <a href="#">Table 88: Absolute maximum ratings</a> .                                                                                                         |
| 07-Sep-2007 | 6        | $V_{ESD}$ min modified for MLP in <a href="#">Table 88: Absolute maximum ratings</a> .                                                                                                                                                                                                                                                                   |
| 08-Apr-2008 | 7        | Response parameters modified in <a href="#">Section 20.14: Get Multiple Block Security Status on page 62</a> .<br>UFDFPN8 package mechanical data updated and dimensions in inches rounded to four decimal digits instead of three in <a href="#">Table 92: UFDFPN8 - 8-lead ultra thin fine pitch dual flat package no lead (MLP) mechanical data</a> . |
| 16-Sep-2008 | 8        | LRI2K products are no longer offered in A1 inlays and A6 and A7 antennas.<br>$T_{STG}$ added for UFDFPN8 package in <a href="#">Table 88: Absolute maximum ratings</a> . <a href="#">Table 93: Ordering information scheme</a> clarified.                                                                                                                |

**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.

© 2008 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 - Singapore - Spain - Sweden - Switzerland - United Kingdom - United States of America

[www.st.com](http://www.st.com)

# Mouser Electronics

Authorized Distributor

Click to View Pricing, Inventory, Delivery & Lifecycle Information:

[STMicroelectronics:](#)

[LRI2K-A1S/1GE](#) [LRI2K-W4/3GE](#)