



# SRIX4K

## 13.56MHz Short Range Contactless Memory Chip with 4096 bit EEPROM, anti-collision and anti-clone functions

PRELIMINARY DATA

- ISO 14443 - 2 Type B Air Interface Compliant
- ISO 14443 - 3 Type B Frame Format Compliant
- 13.56MHz Carrier Frequency
- 847kHz Sub-carrier Frequency
- 106K bit/s Data Transfer
- France Telecom Proprietary Anti-Clone Function
- 8 bits Chip\_ID based anticollision system
- 2 Count Down Binary Counters
- 64 bits Unique Identifier
- 4096 bit EEPROM with Write Protect Feature
- READ BLOCK & WRITE BLOCK (32 Bits)
- Internal Tuning Capacitor
- 1M ERASE/WRITE Cycles
- 10 Years Data Retention
- Self-Timed Programming Cycle
- 5ms Typical Programming Time

### DESCRIPTION

The SRIX4K is a contactless memory powered by the received carrier electromagnetic wave. It is a 4096 bits user EEPROM fabricated with STMicroelectronics CMOS technology. The memory is organised as 128 blocks of 32 bits. The SRIX4K is accessed via the 13.56MHz carrier. Incoming data are demodulated and decoded from the received Amplitude Shift Keying modulation signal (ASK) and outgoing data are generated by load variation using Bit Phase Shift Keying coding (BPSK) of a 847kHz sub-carrier. The received ASK wave is 10% modulated. The Data transfer rate between the SRIX4K and the reader is 106Kbit/s in both reception and emission modes.

The SRIX4K follows the ISO 14443 part 2 type B recommendation for Radio frequency power and signal interface.

Figure 1. Delivery Forms

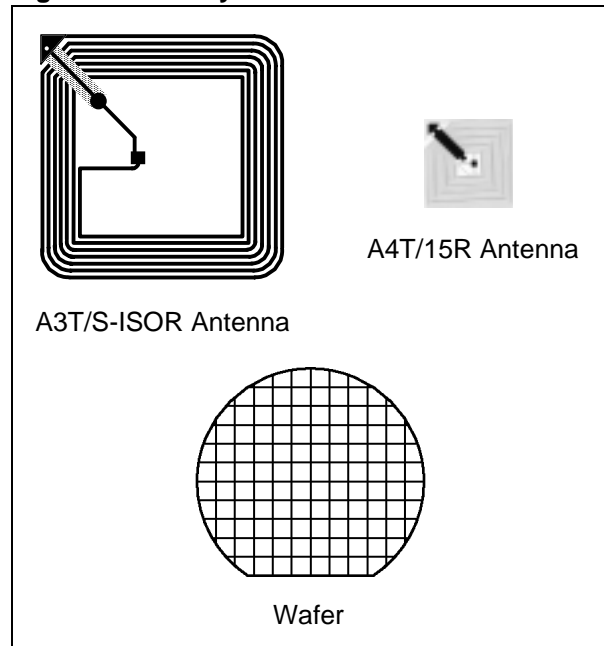


Figure 2. Pad Connections

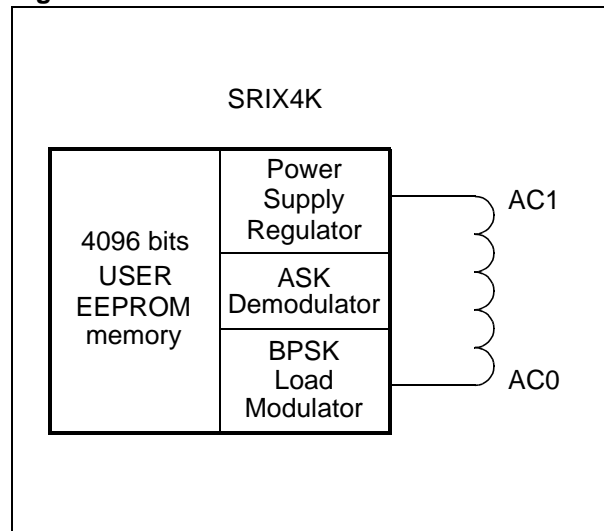


Table 1. Signal Names

AC1	Antenna Coil
AC0	Antenna Coil

**SIGNALS DESCRIPTION**

**AC1, AC0:** AC1 and AC0 pads must directly be bonded to the antenna.

**DESCRIPTION (cont'd)**

The SRIX4K targets short range applications which need secured and re-usable products. The SRIX4K includes an anti-collision mechanism which allowed to detect and select tags present at the same time in the reader range. The anti-collision is based on a probabilist scanning methodology using slot markers. The SRIX4K provides an anti-clone function which allows its authentication. Using the STMicroelectronics single chip coupler CRX14, it will be easy to design a reader with the authentication capability and get a final application with a high level of security.

The SRIX4K contactless EEPROM memory can be randomly read and write in block mode. One block is composed by 32 bits. It offers a set of 10 commands: READ\_BLOCK, WRITE\_BLOCK, INITIATE, PCALL16, SLOT\_MARKER, SELECT, COMPLETION, RESET\_TO\_INVENTORY, AUTHENTICATE, GET\_UID.

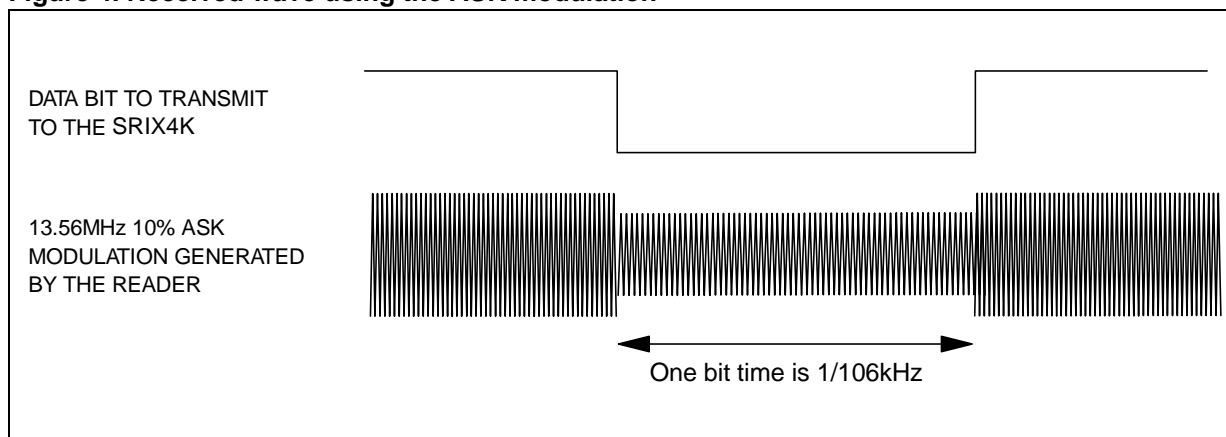
The SRIX4K memory is divided into 3 different areas as described in Figure 8 on page 8. The first area is a resettable EPROM in which bits can only switch from the value '1' to '0'. Using a special command, it is possible to erase all bits of this area to '1'. The second area provides a 24 bits and a 32 bits binary counters which can only be decremented from value FF...Fh to 00...0h. The last area is the EEPROM memory. It is accessible by block of 32 bits and include auto-erase cycle during each WRITE\_BLOCK command.

**DATA TRANSFER**

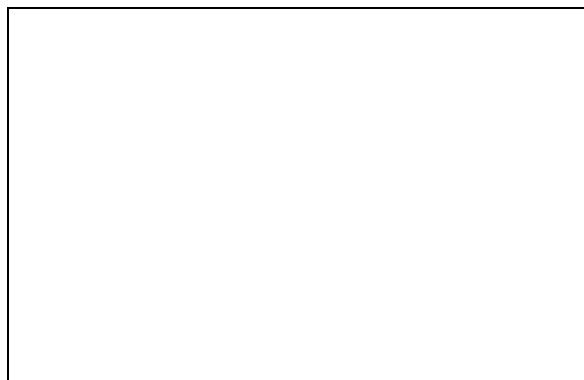
**Input data transfer from the Reader to the SRIX4K (Request Frame)**

The reader used to access the SRIX4K must generate a 13.56MHz sinusoidal carrier frequency on its antenna with enough energy to "tele-power" the memory. The energy received on the SRIX4K antenna will be transformed to a Power Supply Voltage through a regulator and to data bits through the ASK demodulator. In order to decode correctly the information send to the SRIX4K, the reader must use a 10% amplitude modulation of the 13.56 MHz wave as shown in Figure 4. The data transfer rate is 106 Kbit/s.

**Figure 4. Received wave using the ASK modulation**



**Figure 3. SRIX4K Die Floor Plan**



**Character transmission format for Request Frame**

Data bytes are transmitted and received by the SRIX4K by characters of 10 bits as detailed in Table 2 with the LSB ( $b_0$ ) transmitted first. These characters, with the addition of the Start Of Frame (SOF) and the End Of Frame (EOF), are grouped to form a Command Frame as shown in Figure 6. The frame includes SOF, instructions, addresses, data, CRC and EOF as defined in the ISO14443-3 type B. If an error is detected during the data transfer, no error frame is generated by the SRIX4K and the instruction is not executed. Each bit duration is called ETU (Elementary Time Unit). One ETU is equal to  $9.44\mu\text{s}$  ( $1/106\text{Khz}$ ).

**Table 2. SRIX4K Request Frame character format**

	$b_0$	$b_1$	$b_2$	$b_3$	$b_4$	$b_5$	$b_6$	$b_7$	$b_8$	$b_9$	
1 ETU	Start "0"	Information Byte							MSB	Stop "1"	

Bit	Description	Value
$b_0$	start bit used to synchronize the transmission	$b_0 = 0$
$b_1$ to $b_8$	Information byte (instruction, address or data)	Information byte is sent LSB first
$b_9$	Stop bit used to indicate the end of the character	$b_9 = 1$

**Request Start Of Frame**

The SOF described in Table 3 is composed by:

- one falling edge,
- followed by 10 etu with a logical 0,
- followed by one single rising edge
- followed by at least 2 etu (but no more than 3 etu) with a logical 1.

**Table 3. Request Start Of Frame**

	$b_0$	$b_1$	$b_2$	$b_3$	$b_4$	$b_5$	$b_6$	$b_7$	$b_8$	$b_9$	$b_{10}$	$b_{11}$
ETU	0	0	0	0	0	0	0	0	0	0	1	1

**Request End Of Frame**

The EOF shown in Table 4 is composed by:

- one falling edge,
- followed by 10 etu with a logical 0,
- followed by one single rising edge

**Table 4. Request End Of Frame**

	$b_0$	$b_1$	$b_2$	$b_3$	$b_4$	$b_5$	$b_6$	$b_7$	$b_8$	$b_9$	
ETU	0	0	0	0	0	0	0	0	0	0	0

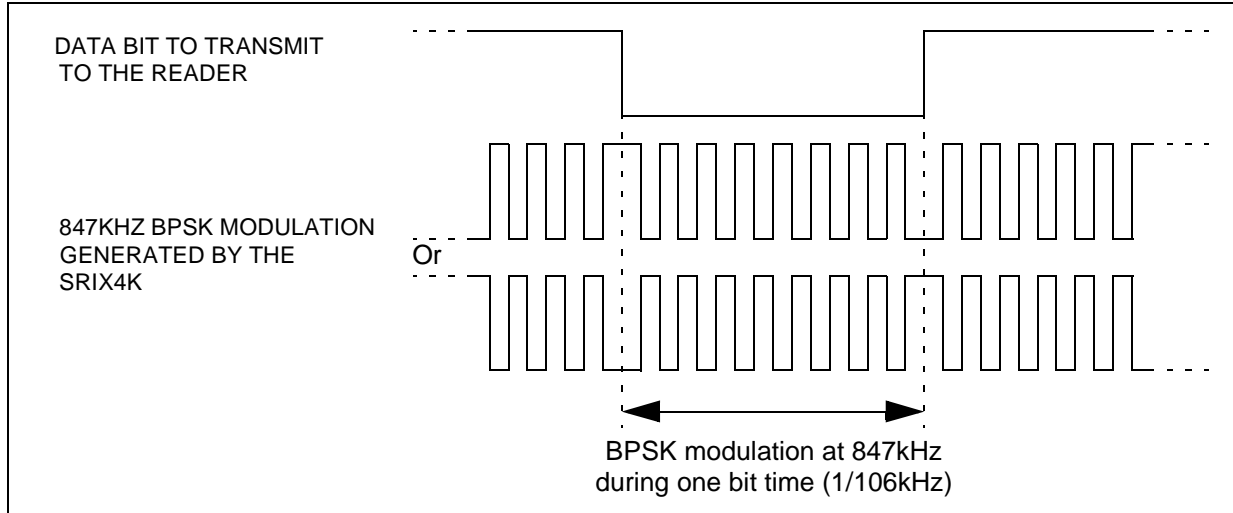
**Output data transfer from the SRIX4K to the reader (Answer Frame)**

Data bits forwarded by the SRIX4K use the retro-modulation mechanism. This modulation is obtained by modifying the SRIX4K current consumption on the antenna (load modulation). The load modulation will induce a variation on the reader antenna using the coupling phenomenon. With an appropriate detector



circuitry, the reader will be able to receive information from the SRIX4K. To improve the load modulation detection, the data is transmitted using the BPSK coding format of a 847kHz sub-carrier frequency  $f_s$  as shown in Figure 5 and as specify in ISO14443-2 type B.

**Figure 5. Emitted wave using the BPSK sub-carrier modulation**



**Character transmission format for Answer Frame**

The character format is the same as for the input data transfer (Table 2). The transmitted frames include SOF, data, CRC and EOF (Figure 6). Like the input data transfer, in case of error, the reader does not emit any error code to the SRIX4K, but must be able to detect and manage this situation. The data transfer rate is 106 kbit/s.

**Answer Start Of Frame**

The SOF described in Table 5 is composed by:

- 10 etu with a logical 0,
- 2 etu with a logical 1.

**Table 5. Answer Start Of Frame**

	$b_0$	$b_1$	$b_2$	$b_3$	$b_4$	$b_5$	$b_6$	$b_7$	$b_8$	$b_9$	$b_{10}$	$b_{11}$
ETU	0	0	0	0	0	0	0	0	0	0	1	1

**Answer End Of Frame**

The EOF shown in Table 6 is composed by:

- 10 etu with a logical 0,
- 2 etu with a logical 1

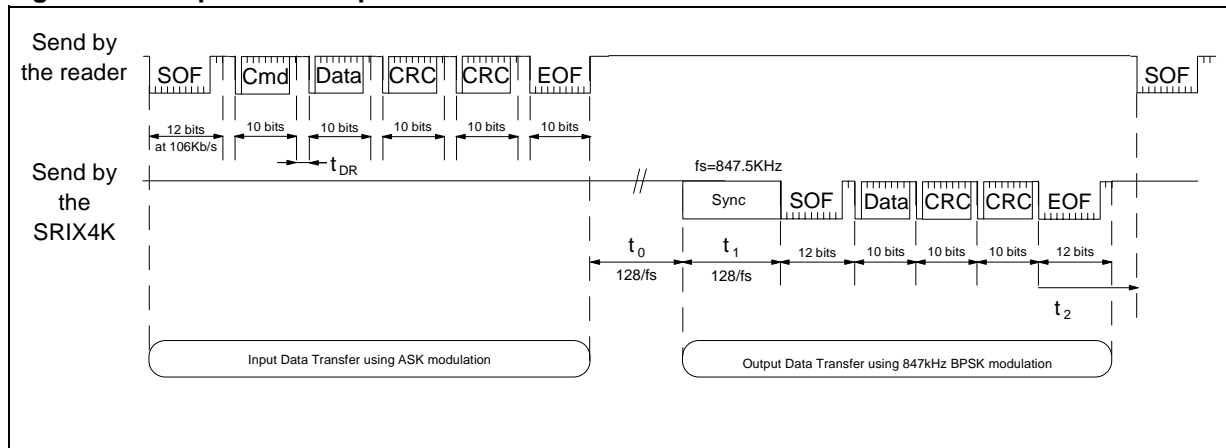
**Table 6. Answer End Of Frame**

	$b_0$	$b_1$	$b_2$	$b_3$	$b_4$	$b_5$	$b_6$	$b_7$	$b_8$	$b_9$	$b_{10}$	$b_{11}$
ETU	0	0	0	0	0	0	0	0	0	0	1	1

**Transmission frame**

Between the Request and the Answer data transfer, there is a guard time without ASK and BPSK modulation for a minimum delay of  $t_0=128/f_s$ . This delay allows the reader to switch from transmission to reception mode and is applied after each frame. After  $t_0$ , the 13.56MHz carrier frequency is modulated by the SRIX4K at 847kHz for a delay of  $t_1=128/f_s$  to allow the reader synchronization. After  $t_1$ , the first phase transition generated by the SRIX4K will represent the start bit ('0') of the Answer SOF. After the falling edge of the Answer EOF, the reader has to wait to minimum delay of  $t_2$  before sending a new Request Frame to the SRIX4K.

**Figure 6. Example of a Complete Transmission frame**



**CRC**

The 16 bits CRC used by the SRIX4K follows the ISO14443 type B recommendation. For further information, please see the annex on page 32. The initial register content is all ones : "FFFF".

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

Upon reception of a Request from a reader, the SRIX4K verify that the CRC value is valid. If it is invalid, it discard the frame and does not answer to the reader.

Upon reception of an Answer from the SRIX4K, it is recommended that the reader verify that the CRC value is valid. If it is invalid, actions to be performed are left to the responsibility of the reader designer.

The CRC is transmitted Least Significant Byte first. Each byte is transmitted Least Significant Bit first .

**Table 7. CRC transmission rules**

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



**Table 8. AC Characteristics (TA = -20°C to 85°C)**

Symbol	Parameter	Condition	Min	Max	Unit
f <sub>CC</sub>	External RF Signal Frequency		13.553	13.567	MHz
MI <sub>CARRIER</sub>	Carrier Modulation Index	MI=(A-B)/(A+B)	8	14	%
t <sub>RFR</sub> , t <sub>RFF</sub>	10% Rise and Fall time			1.5	µs
t <sub>RFSBL</sub>	Minimum Pulse Width for Start bit	etu = 128/f <sub>CC</sub>	9.44		µs
t <sub>JIT</sub>	ASK modulation Data Jitter	Coupler to SRIX4K	- 2	+ 2	µs
t <sub>MAX</sub>	Maximum Carrier Rise Time	Zero to Maximum field strength induced Voltage on AC0/AC1		3	V/µs
t <sub>MIN CD</sub>	Minimum Time from Carrier Generation to First Data		5		ms
f <sub>S</sub>	Sub-carrier Frequency	f <sub>CC</sub> /16	847.5		KHz
t <sub>0</sub>	Antenna Reversal delay	128/f <sub>S</sub>	151		µs
t <sub>1</sub>	Synchronization delay	128/f <sub>S</sub>	151		µs
t <sub>2</sub>	Answer to New Request delay	14 etu	132		µs
t <sub>DR</sub>	Time Between Request characters	Coupler to SRIX4K	0	57	µs
t <sub>DA</sub>	Time Between Answer characters	SRIX4K to Coupler	0		µs
r <sub>L</sub>	Resistive Load (for Modulation)		1800	2900	Ohm
P <sub>A</sub>	H-field Energy on SRIX4K Antenna <sup>1</sup>	A3T/S-ISOR Antenna A4T/15R Antenna	1.0 tbc	7.5 tbc	A/m A/m
t <sub>W</sub>	Programming Time for WRITE	No Auto-Erase Cycle		2.5	ms
t <sub>W</sub>	Programming Time for WRITE	With Auto-Erase Cycle		5	ms
t <sub>W</sub>	Programming Time for WRITE	Binary Counter Decrement		7	ms
t <sub>AUT</sub>	Authenticate Time		tbc		µs

1) P<sub>A</sub> Min is the minimum H-field required to communicate with the SRIX4K  
P<sub>A</sub> Max is the maximum H-field that the device can support before clamping the incoming signal

**Table 9. DC Characteristics (TA = -20°C to 85°C)**

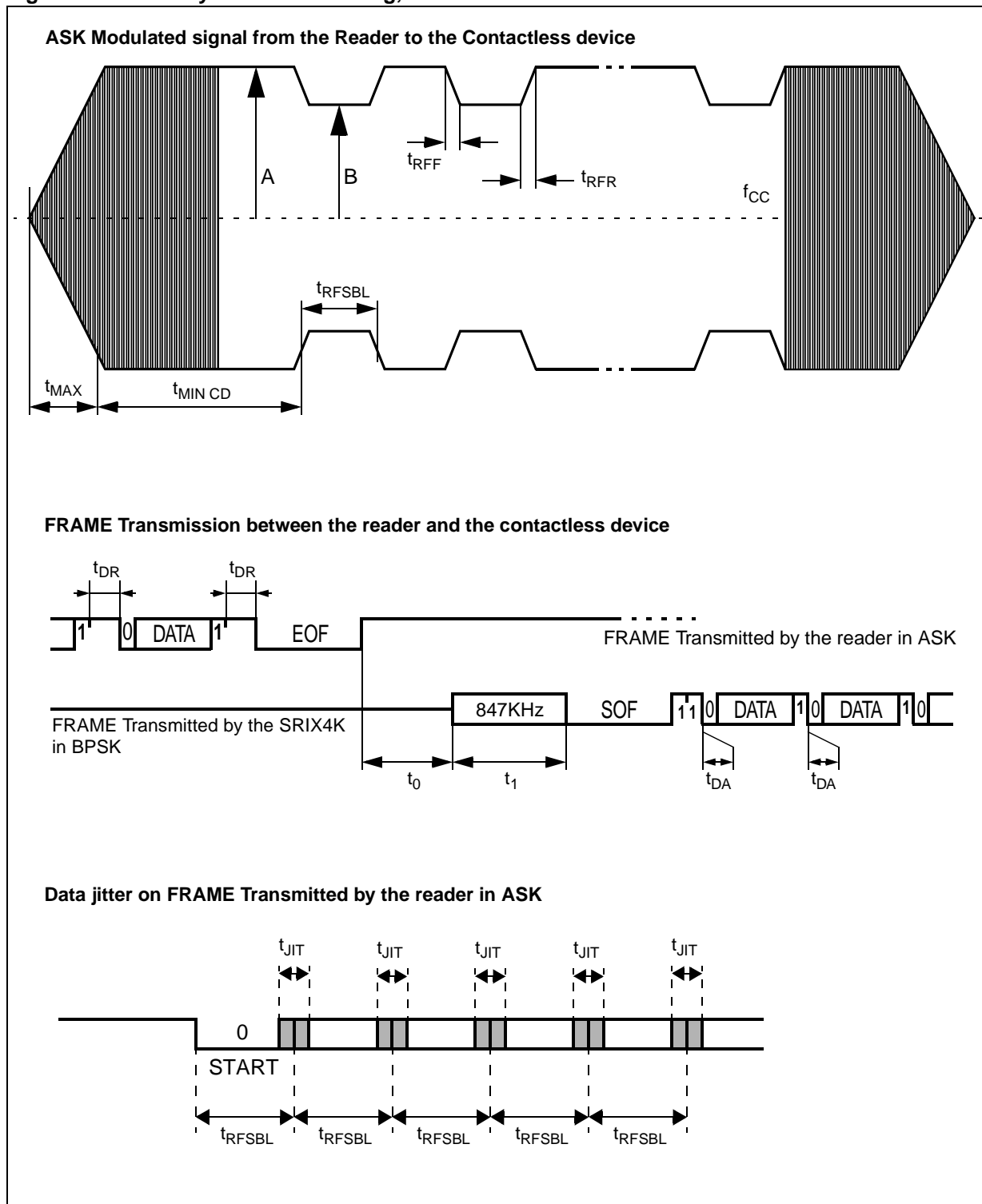
Symbol	Parameter	Condition	Min	Max	Unit
V <sub>CC</sub>	Regulated Voltage		2.5	3.5	V
I <sub>CC</sub>	Supply Current (Active in Read)	V <sub>CC</sub> = 3.0V		100	µA
I <sub>CC</sub>	Supply Current (Active in Write)	V <sub>CC</sub> = 3.0V		250	µA

**Table 10. Absolute Maximum Ratings<sup>1</sup>**

Symbol	Parameter	Value	Unit
T <sub>A</sub>	Ambient Operating Temperature	-20 to + 85	°C
T <sub>STG</sub>	Storage Temperature	-30 to + 90	°C
I <sub>CC</sub>	Supply Current on AC0 / AC1	20	mA
V <sub>ESD</sub>	Electrostatic Discharge Voltage (Human Body model)	4000	V

1) Except for the rating "Operating Temperature Range", stresses above those listed in the Table "Absolute Maximum Ratings" may cause permanent damage to the device. These are stress ratings only, and operation of the device at these or any other conditions above those indicated in the Operating sections of this specification is not implied. Exposure to Absolute Maximum Rating conditions for extended periods may affect device reliability. Refer also to the ST SURE Program and other relevant quality documents.

Figure 7. SRIX4K Synchronous Timing, Transmit and Receive



## SRIX4K

### MEMORY MAPPING

The SRIX4K is organised as 129 blocks of 32 bits as shown in Figure 8. All blocks are accessible by the READ\_BLOCK command. Depending on the write access, they can be updated by the WRITE\_BLOCK command. A WRITE\_BLOCK updates all the 32 bits of the block.

**Figure 8. SRIX4K Memory Mapping.**

Block Addr	Msb b <sub>31</sub>	b <sub>24</sub> b <sub>23</sub>	32 bits Block b <sub>16</sub> b <sub>15</sub>	b <sub>8</sub> b <sub>7</sub>	Lsb b <sub>0</sub>	Description
0			32 bits Boolean Area			Resettable EPROM
1			32 bits Boolean Area			
2			32 bits Boolean Area			
3			32 bits Boolean Area			
4			32 bits Boolean Area			
5			32 bits binary counter			Count down Counter
6			32 bits binary counter			
7			User Area			Lockable EEPROM
8			User Area			
9			User Area			
10			User Area			
11			User Area			
12			User Area			
13			User Area			
14			User Area			
15			User Area			
16			User Area			
...			User Area			EEPROM
127			User Area			
255	OTP_Lock_Reg		ST Reserved		Fixed Chip_ID (Option)	System EPROM
UID0	64 bits UID Area					ROM
UID1						



**Resettable EPROM (addresses 0 to 4)**

Block Addr	Msb b <sub>31</sub>	b <sub>24</sub> b <sub>23</sub>	32 bits Block b <sub>16</sub> b <sub>15</sub>	b <sub>8</sub> b <sub>7</sub>	Lsb b <sub>0</sub>	Description
0	32 bits Boolean Area					resettable EPROM
1	32 bits Boolean Area					
2	32 bits Boolean Area					
3	32 bits Boolean Area					
4	32 bits Boolean Area					

Each location gives access to an individual 32 bits boolean word. In this area, a WRITE\_BLOCK command does not erase the previous content before the write cycle. This feature allow to switch selected bits from binary value '1' to '0'. All bits previously at '0' stay at the same value. When all 32 bits of the block are set to '0', the block is empty and cannot be updated any more.

**WRITE\_BLOCK update in standard mode (Binary format)**

	b <sub>31</sub>													b <sub>0</sub>	
Previous data stored in block	1	...	1	1	0	1	0	1	1	1	1	1	0	1	1
Data to be written	1	...	1	0	0	1	0	1	1	0	0	1	1	1	1
New data stored in block	1	...	1	0	0	1	0	1	1	0	0	1	0	1	1

The SRIX4K allow to erase all five 32 bits blocks by adding an auto-erase cycle on the WRITE\_BLOCK command. This condition is triggered by the SRIX4K on the detection of a reload command. The reload command is performed through a specific update on the 32 bits binary counter at location 6 (see counter description for details).

**WRITE\_BLOCK update in reload mode (Binary format)**

	b <sub>31</sub>													b <sub>0</sub>	
Previous data stored in block	1	...	1	1	0	1	0	1	1	1	1	1	0	1	1
Data to be written	1	...	1	1	1	1	0	1	1	0	0	1	1	1	1
New data stored in block	1	...	1	1	1	1	0	1	1	0	0	1	1	1	1



## SRIX4K

### Binary Counter (addresses 5 to 6)

Block Addr	Msb b <sub>31</sub>	b <sub>24</sub> b <sub>23</sub>	32 bits Block b <sub>16</sub> b <sub>15</sub>	b <sub>8</sub> b <sub>7</sub>	Lsb b <sub>0</sub>	Description
5	32 bits binary counter					Count down Counter
6	32 bits binary counter					

#### – 32 bits binary counter

The 32 bits binary counter in block 5 and in block 6 allow each to count down units from  $2^{32}$  (4096 Million) to 0 by adding bits at the binary value '0'. The SRIX4K provides a dedicated logic which autorises the update of the counter if and only if the new value is lower than the previous one. This feature allows the application to count down 1 by 1 or by steps. The initial value of this counter is FFFFFFFFh. When it reaches the value 00000000h, the counter is empty and cannot be reloaded. Counter update is directly performed by the WRITE\_BLOCK command at the location 5 or 6 which will write the new 32 bits counter value.

#### Count down exemple (Binary format)

	b <sub>31</sub>	...	b <sub>24</sub>	b <sub>23</sub>	b <sub>16</sub>	b <sub>15</sub>	b <sub>8</sub>	b <sub>7</sub>	b <sub>0</sub>
Initial data	1	...	1	1	1	1	1	1	1
1 unit decrement	1	...	1	1	1	1	1	1	0
1 unit decrement	1	...	1	1	1	1	1	1	0
1 unit decrement	1	...	1	1	1	1	1	1	0
8 units decrement	1	...	1	1	1	1	1	1	0
Increment: Not allowed	1	...	1	1	1	1	1	1	0

The counter in block 6 control the reload mechanism which trigger the reset of the resettable EPROM area (address 0 to 4). Bits b<sub>31</sub> to b<sub>21</sub> are also used as a 11 bits reload counter. When the SRIX4K detect an update in one of these 11 bits, it set the erase cycle in the WRITE\_BLOCK for the locations 0 to 4. The erase cycle will be active up to a POWER-OFF or a SELECT command. The SRIX4K resettable EPROM area can be reloaded up to 2047 times ( $2^{11}-1$ ).

## EEPROM (addresses 7 to 127)

Block Addr	Msb b <sub>31</sub>	b <sub>24</sub> b <sub>23</sub>	32 bits Block b <sub>16</sub> b <sub>15</sub>	b <sub>8</sub> b <sub>7</sub>	Lsb b <sub>0</sub>	Description
7			User Area			EEPROM
8			User Area			EEPROM Lockable
9			User Area			
10			User Area			
11			User Area			
12			User Area			
13			User Area			
14			User Area			
15			User Area			
16			User Area			EEPROM
...			User Area			
127			User Area			

Blocks from locations 7 to 127 offer 121 EEPROM blocks of 32 bits each (484 bytes). They can be accessed by READ\_BLOCK and WRITE\_BLOCK which provide an auto-erase cycle prior to the write. Blocks 7 to 15 offer in addition the possibility to be protected against write. Write access is controlled by the 8 bits of the OTP\_Lock\_Reg area at location 255. When protected, it is not possible to unprotect them.

## SRIX4K

---

### System Area

Block Addr	Msb b <sub>31</sub>	b <sub>24</sub> b <sub>23</sub>	32 bits Block b <sub>16</sub> b <sub>15</sub>	b <sub>8</sub> b <sub>7</sub>	LSb b <sub>0</sub>	Description
255	OTP_Lock_Reg		ST Reserved		Fixed Chip_ID (Option)	EPROM

The system area allow to modify the setting of the SRIX4K. It provides 3 registers described below. In this area, a WRITE\_BLOCK command does not erase the previous content before the write cycle. This feature allow to switch selected bits from binary value '1' to '0'. All bits previously at '0' stay at the same value. When all 32 bits of the block are set to '0', the block is empty and cannot be updated any more.

#### – OTP\_Lock\_Reg

The 8 bits (b<sub>31</sub>-b<sub>24</sub>) of the block 255 are used as OTP\_Lock\_Reg bits in the SRIX4K. They control write access to the 9 EEPROM blocks from location 7 to 15 as followed:

- When b<sub>24</sub> is set to '0', the block 7 and 8 are write protected.
- When b<sub>25</sub> is set to '0', the block 9 is write protected.
- When b<sub>26</sub> is set to '0', the block 10 is write protected.
- When b<sub>27</sub> is set to '0', the block 11 is write protected.
- When b<sub>28</sub> is set to '0', the block 12 is write protected.
- When b<sub>29</sub> is set to '0', the block 13 is write protected.
- When b<sub>30</sub> is set to '0', the block 14 is write protected.
- When b<sub>31</sub> is set to '0', the block 15 is write protected.

The OTP\_Lock\_Reg bits cannot be erased. When write protected, the EEPROM blocks behaves as ROM block and it will not possible to unprotect them.

#### – Fixed Chip\_ID (Option)

The SRIX4K provides an complete anti-collision mechanism based on a random 8 bits Chip\_ID. In order to select a SRIX4K, it is necessary to run an anti-collision sequence to find its Chip\_ID. This feature is very flexible but need time requested by the searching loop.

In certain kind of application, it could be very effective to know in advance the value of the SRIX4K Chip\_ID in order to select it directly without doing any anti-collision sequence. This features save time and allows also to identify and select the SRIX4K by its already known Chip\_ID. For this reason, the SRIX4K offer an optional mask setting which allow to program a fixed 8 bits Chip\_ID in the system area using bits b<sub>7</sub> to b<sub>0</sub>. When the fixed Chip\_ID option is used, the random Chip\_ID function is disabled.

## DEVICE OPERATIONS (INSTRUCTIONS)

All instructions, data and CRC are transmitted to the SRIX4K in 10 bits character format using the ASK modulation. The start bit  $b_0$  of the 10 bits is sent first. The command frame received by the SRIX4K on the antenna is demodulated by the 10% ASK demodulator and is decoded by the internal logic. Prior to any operation, the SRIX4K must have been previously selected by a SELECT command. Each frame transmitted to the SRIX4K must start with a Start Of Frame, followed by one or more data and is ended by two CRC bytes and the End Of Frame. When an invalid frame is decoded by the SRIX4K (wrong instruction or CRC error), the memory does not send any error code.

When a valid frame is received, the SRIX4K may have to send back data to the reader. For this, it sends 10 bits characters back using the BPSK coding with SOF and EOF. The transfer is ended by the SRIX4K sending the 2 CRC bytes and the EOF.

## DEVICE STATE DESCRIPTION

The SRIX4K can be putted in several states allowing the logic to answer only on specific commands. These states are mainly used during anti-collision sequence and allowed to identify and access the SRIX4K in a very short time. The SRIX4K provides 6 different states which are described in the following lines and in Figure 9.

### POWER-OFF State

The SRIX4K is in POWER-OFF state when the electromagnetic field around the tag is not strong enough. In this state, the SRIX4K does not answer to any command.

### READY State

When the electromagnetic field is strong enough, the SRIX4K goes in READY state. After the power-off/Power-on transition, the Chip\_ID is initialised with a random value, all the logic is reset and waiting for a INITIATE() command. In READY state, all other commands different from INITIATE() are ignored by the SRIX4K.

### INVENTORY State

The SRIX4K switch from the READY to the INVENTORY state after a INITIATE() command. In INVENTORY state, it answers to all anti-collision commands INITIATE(), PCALL16() and SLOT\_MARKER() and stays in this state. The SRIX4K switches to the SELECTED state after a SELECT(Chip\_ID) command if it is matching its own Chip\_ID. If the Chip\_ID does not match, it stays in INVENTORY state.

### SELECTED State

In the SELECTED state, the SRIX4K is active and answers to all READ\_BLOCK(), WRITE\_BLOCK(), AUTHENTICATE() and GET\_UID() commands. When a SRIX4K switches in SELECTED state, it will not answer to anti-collision commands anymore. In order to access to an other tag, it is possible to put the SRIX4K in the DESELECTED state by sending a SELECT( $\neq$  Chip\_ID), on which the Chip\_ID is not matching, or in DESACTIVATED state by sending a COMPLETION() command. Only one SRIX4K can be in the SELECTED state at a time.

### DESELECTED State

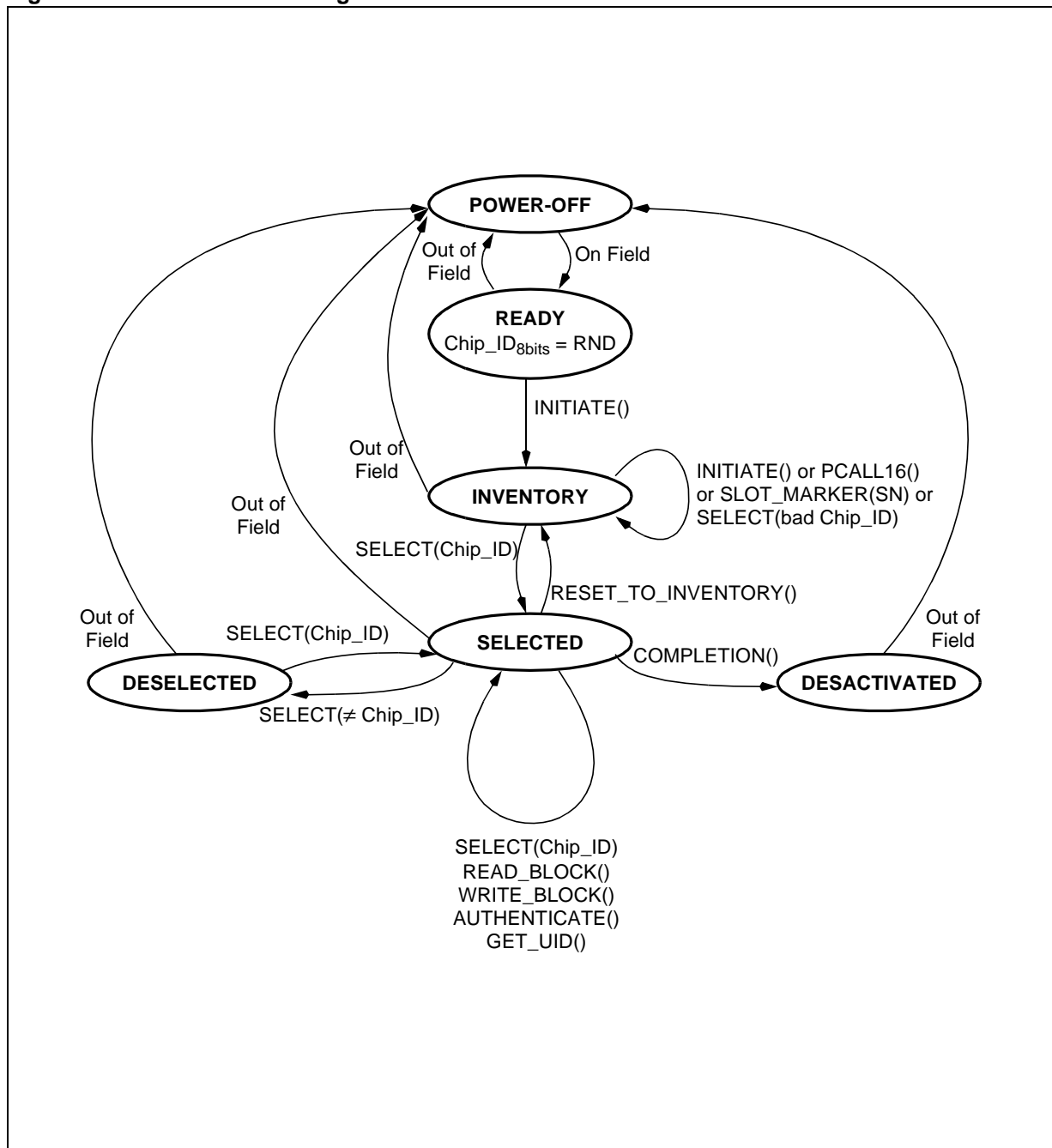
In this state, the SRIX4K is waiting for a SELECT(Chip\_ID), on which the Chip\_ID is matching, to switch back to the SELECTED state. All other commands are ignored.

### DESACTIVATED State

In this state, the SRIX4K is waiting for a power-off. All commands are ignored.



Figure 9. States Transition Diagram



**ANTI-COLLISION**

The SRIX4K provides an anti-collision method to search for the Chip\_ID of each device available in the reader field range. When known, the Chip\_ID is used to select individually a SRIX4K and access to its memory. The anti-collision sequence is managed by the reader through a set of commands described in the following pages: INITIATE(), PCALL16() and SLOT\_MARKER().

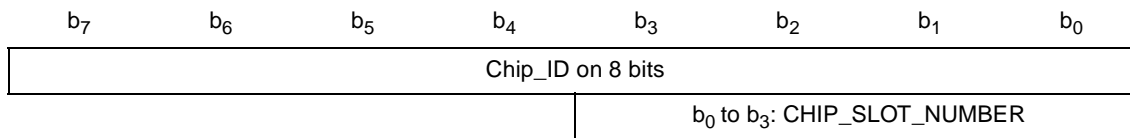
The reader is the master of the communication with one or multiple SRIX4K. It initiates tag communication activity by issuing a INITIATE(), PCALL16() or SLOT\_MARKER() commands to prompt for SRIX4K to answer. During the anti-collision sequence it may happen that two or more SRIX4K respond simultaneously: this is a collision. The command set allows the reader to handle sequences to separate SRIX4K transmissions in time. Having completed the anti-collision sequence, SRIX4K communication will fully be under control of the reader, allowing only one SRIX4K to talk at a time.

The Anti-collision scheme is based on definition of time slots in which SRIX4K are invited to answer with minimum identification data: the Chip\_ID. The number of slots is fixed to 16 for the PCALL16(). For INITIATE(), there is no slot and the SRIX4K answer after the command. SRIX4K are allowed to answer only once in the anticollision sequence. Consequently, even in case of multiple SRIX4K present in the reader field, there will probably be a slot in which only one SRIX4K answers and where the reader is able to capture the Chip\_ID. Based on the Chip\_ID, the reader is able to establish a communication channel with the identified SRIX4K. An anticollision sequence allows selection of one SRIX4K at a time.

The SRIX4K uses a 8 bits Chip\_ID value which allows up to 256 tags to be selected individually within a reader field range. In READY state, the Chip\_ID is initialised with a random value.

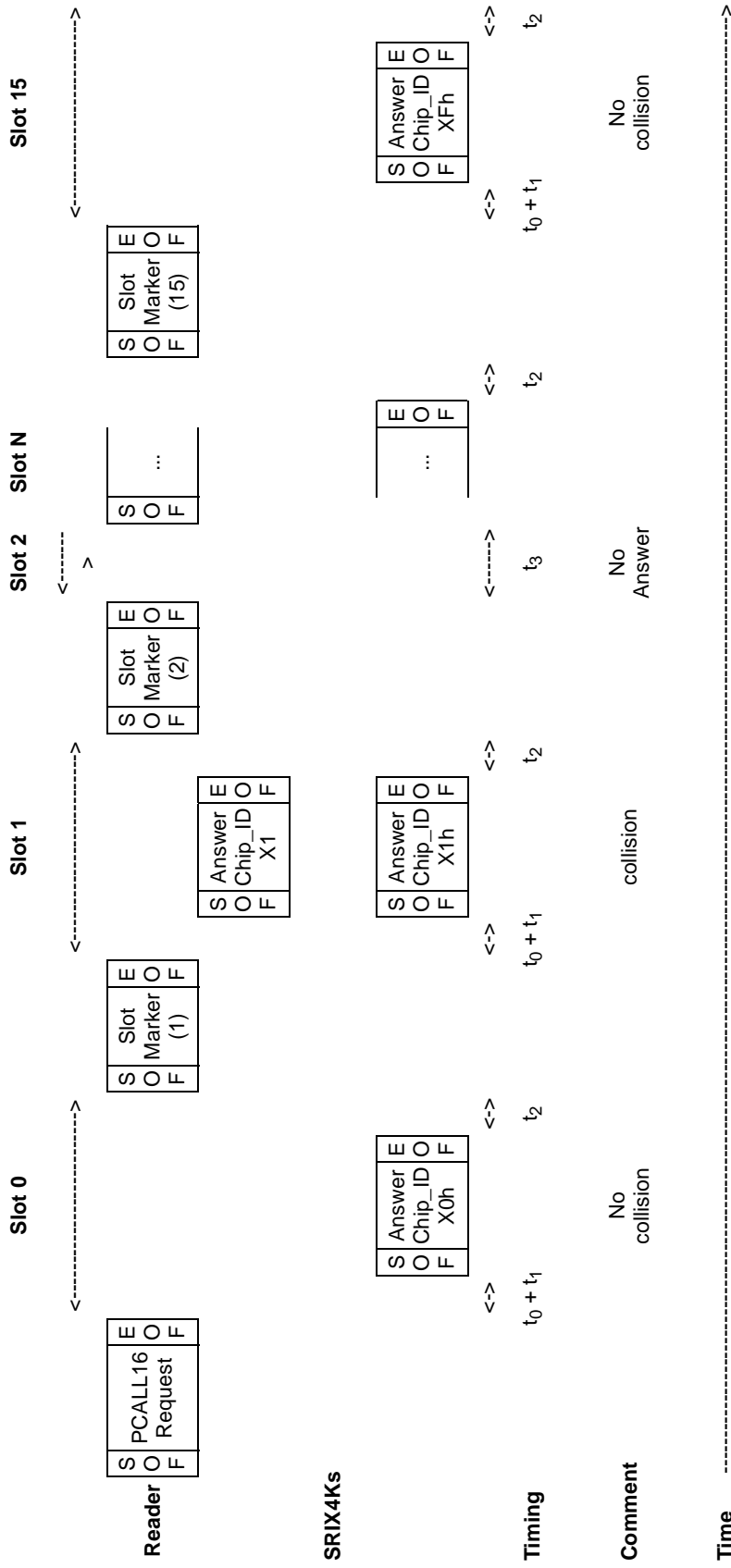
The 4 LSbit (b<sub>0</sub> to b<sub>3</sub>) of the Chip\_ID are also named CHIP\_SLOT\_NUMBER. This 4 bits value is used by the PCALL16() and SLOT\_MARKER() commands during the anti-collision sequence in INVENTORY state.

**Figure 10. SRIX4K Chip\_ID description**



Each time the SRIX4K receives a PCALL16() command, the CHIP\_SLOT\_NUMBER is set with a new 4 bits random value. If the new value is 0000<sub>b</sub>, the SRIX4K send back its complete 8 bits Chip\_ID on its PCALL16() answer. The PCALL16() command is also used to define the slot number 0 of the anti-collision sequence. When the SRIX4K receives the SLOT\_MARKER(SN) command, it first compares its CHIP\_SLOT\_NUMBER with the SLOT\_NUMBER parameter (SN). If values are matching, the SRIX4K send back its Chip\_ID after the command. If they are not matching, the SRIX4K do not answer after the command. The SLOT\_MARKER(SN) command allows to define all the anti-collision slot number from 1 to 15.

Figure 11. TDescription of a possible anti-collision sequence



Note: The value X in the Answer Chip\_ID means a random hexadecimal character from 0 to 9, A, B, C, D, E, F





### Anti-Collision sequence description

The anti-collision sequence is initiated by the INITIATE() command which trigger all the SRIX4Ks available in the reader field range in the INVENTORY state. Only SRIX4K in INVENTORY state can be handled by the anti-collision commands PCALL16() and SLOT\_MARKER(SN).

A new SRIX4K entered in the field range during the anti-collision sequence will not be taken into account as it will not answer the PCALL16() or SLOT\_MARKER(SN) commands (READY state). In order to add it in the anti-collision sequence, it must receive the INITIATE() to switch to the INVENTORY state.

A standard anticollision sequence is done as followed and as described in Figure 11:

- Step1 Init: Send INITIATE()
  - If no answer is detected, go to step1
  - If only 1 answer is detected, select and access the SRIX4K. After the access, deselect the tag and go to step1
  - If a collision (many answers) is detected, go to step2
- Step2 Slot 0: Send PCALL16()
  - If no answer or collision are detected, go to step3
  - If 1 answer is detected, store the Chip\_ID, Send SELECT() and go to step3
- Step3 Slot 1: Send SLOT\_MARKER(1)
  - If no answer or collision are detected, go to step4
  - If 1 answer is detected, store the Chip\_ID, Send SELECT() and go to step4
- Step4 Slot 2: Send SLOT\_MARKER(2)
  - If no answer or collision are detected, go to step5
  - If 1 answer is detected, store the Chip\_ID, Send SELECT() and go to step5
- StepN Slot N: Send SLOT\_MARKER(3 up to 14) .....
  - If no answer or collision are detected, go to stepN+1
  - If 1 answer is detected, store the Chip\_ID, Send SELECT() and go to stepN+1
- Step17 Slot 15: Send SLOT\_MARKER(15)
  - If no answer or collision are detected, go to step18
  - If 1 answer is detected, store the Chip\_ID, Send SELECT() and go to step18
- Step18
  - All slots were generated and Chip\_ID values should be stored in the reader memory. Send SELECT(Chip\_ID) and access each identified SRIX4K one by one. After each SRIX4K access, set them in DESELECTED or DESACTIVATED state depending on the application needs.
  - If collisions were detected between Step2 to Step17, go to Step2
  - If no collision was detected between Step2 to Step17, go to Step1,

After each SLOT\_MARKER() command, many or one or no SRIX4K may answer. The reader must handle all this cases and stores all Chip\_ID correctly decoded. At the end of the anti-collision sequence, after SLOT\_MARKER(15), the reader can start working with one SRIX4K by issuing a SELECT() command with one of the Chip\_ID stored. If a collision is detected during the anti-collision sequence, the reader has to generate a new sequence in order to identify all unidentified SRIX4K in the field. The anti-collision sequence can stop when all SRIX4K were identified.

**Exemple of anti-collision sequence**

Command	Tag 1 Chip_ID	Tag 2 Chip_ID	Tag 3 Chip_ID	Tag 4 Chip_ID	Tag 5 Chip_ID	Tag 6 Chip_ID	Tag 7 Chip_ID	Tag 8 Chip_ID	Comments
READY State	40h	13h	3Fh	4Ah	50h	48h	52h	7Ch	each tag get a random Chip_ID
INITIATE()	40h	13h	3Fh	4Ah	50h	48h	52h	7Ch	All tags answer: collisions
PCALL16()	45h	12h	30h	43h	55h	43h	53h	73h	All CHIP_SLOT_NUMBER get a new random value
SELECT(30h)			30h						Slot0: only one answer Tag 3 is identified
SLOT_MARKER(1)									Slot1: No answer
SLOT_MARKER(2)		12h							Slot2: only one answer
SELECT(12h)		12h							Tag2 is identified
SLOT_MARKER(3)				43h		43h	53h	73h	Slot3: collisions
SLOT_MARKER(4)									Slot4: No answer
SLOT_MARKER(5)	45h				55h				Slot5: collisions
SLOT_MARKER(6)									Slot6: No answer
SLOT_MARKER(N)									SlotN: No answer
SLOT_MARKER(F)									Slot15: No answer
PCALL16()	40h			41h	53h	42h	50h	74h	All CHIP_SLOT_NUMBER get a new random value
SLOT_MARKER(1)	40h			41h			50h		Slot0: collisions
SELECT(41h)				41h					Slot1: only one answer Tag4 is identified
SLOT_MARKER(2)						42h			Slot2: only one answer
SELECT(42h)						42h			Tag6 is identified
SLOT_MARKER(3)					53h				Slot3: only one answer
SELECT(53h)					53h				Tag5 is identified
SLOT_MARKER(4)							74h		Slot4: only one answer
SELECT(53h)							74h		Tag8 is identified
SLOT_MARKER(N)									SlotN: No answer
PCALL16()	41h						50h		All CHIP_SLOT_NUMBER get a new random value
SELECT(50h)							50h		Slot0: only one answer Tag 7 is identified
SLOT_MARKER(1)	41h								Slot1: only one answer but already found for tag 4
SLOT_MARKER(N)									SlotN: No answer
PCALL16()	43h								All CHIP_SLOT_NUMBER get a new random value
SLOT_MARKER(3)	43h								Slot0: No answer Slot3: only one answer
SELECT(43h)									Tag 1 is identified

All tag are identified



**ANTI-CLONE FUNCTION**

The SRIX4K provides an anti-clone function which allows the application to authenticate it. This function uses data stored into the SRIX4K memory.

The Authenticate system is based on a proprietary challenge/response mechanism which allows the application software to authenticate a STMicroelectronics secure memory tag from the SRX--- family like the SRIX4K. A reader system designed with the integrated chip coupler CRX14 and the tag SRIX4K can check the authenticity of the tag and protect the application system against silicon copy or emulator.

Complete description of the Authenticate system is available under Non Disclosure Agreement (NDA) with STMicroelectronics. To get more details about this SRIX4K function, please contact the nearest STMicroelectronics sales office.

## SRIX4K

### INITIATE()

Command Code = 06h - 00h

INITIATE() is used to initiate the anti-collision sequence of the SRIX4K. When receiving the INITIATE() command, all SRIX4K in the READY state switch to the INVENTORY state and send back their Chip\_ID value. This command is usefull in the case there is only one SRIX4K in the READY state available in the reader field range. It allows to speed up the search for the Chip\_ID. CHIP\_SLOT\_NUMBER is not used during the INITIATE() command access.

**Table 11. INITIATE Request Format**

SOF	INITIATE		CRC <sub>L</sub>	CRC <sub>H</sub>	EOF
	06h	00h	8 bits	8 bits	

Request parameter:

- No parameter

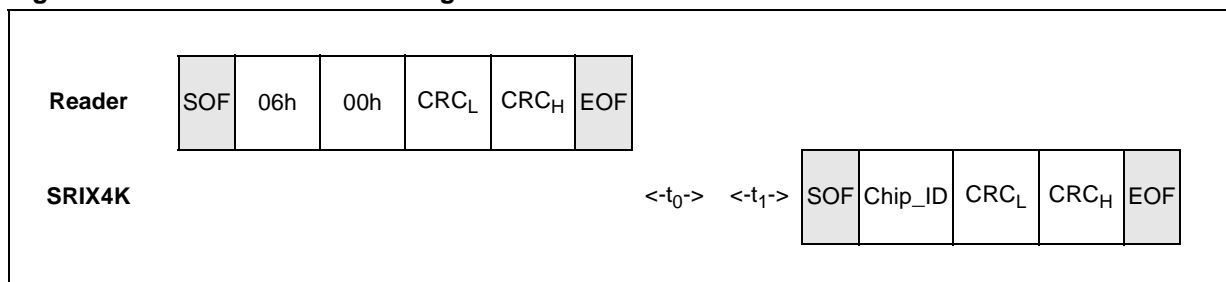
**Table 12. INITIATE Response Format**

SOF	Chip_ID	CRC <sub>L</sub>	CRC <sub>H</sub>	EOF
	8 bits	8 bits	8 bits	

Response parameters:

- Chip\_ID of the SRIX4K.

**Figure 12. INITIATE Frame exchange between Reader and SRIX4K**



**PCALL16()**

Command Code = 06h - 04h

The SRIX4K must be in the INVENTORY mode before interpreting the PCALL16() command.

When receiving the PCALL16() command, the SRIX4K generates first a new random CHIP\_SLOT\_NUMBER value (4 LSbits of the Chip\_ID). CHIP\_SLOT\_NUMBER can take a value between 0 and 15 (1111<sub>b</sub>). This value is stable up to the next PCALL16() command or up to the power-off of the SRIX4K. This new CHIP\_SLOT\_NUMBER value is then compared with the value 0000<sub>b</sub>. If it matches, the SRIX4K sends back its Chip\_ID value. If not, the SRIX4K does not send any response.

This command, in addition to SLOT\_MARKER(), is requested in the case there is more than one SRIX4K in the INVENTORY state available in the reader field range. It allows to search for all Chip\_ID available.

**Table 13. PCALL16 Request Format**

SOF	PCALL16		CRC <sub>L</sub>	CRC <sub>H</sub>	EOF
	06h	04h	8 bits	8 bits	

Request parameter:

- No parameter

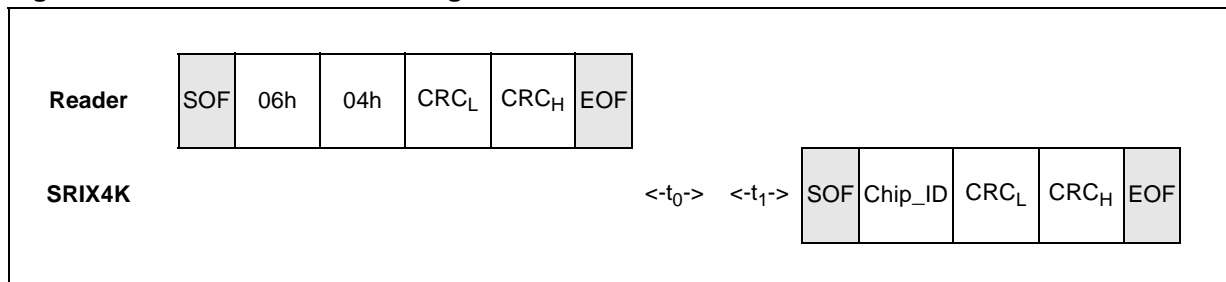
**Table 14. PCALL16 Response Format**

SOF	Chip_ID	CRC <sub>L</sub>	CRC <sub>H</sub>	EOF
	8 bits	8 bits	8 bits	

Response parameters:

- Chip\_ID of the SRIX4K.

**Figure 13. PCALL16 Frame exchange between Reader and SRIX4K**



## SRIX4K

### SLOT\_MARKER(SN)

Command Code = x6h

The SRIX4K must be in the INVENTORY mode before interpreting the SLOT\_MARKER(SN) command.

The SLOT\_MARKER byte code is divided in two parts:

- b<sub>3</sub> to b<sub>0</sub>: 4 bits command code fixed at the value 6
- b<sub>7</sub> to b<sub>4</sub>: 4 bits named SLOT\_NUMBER (SN) which gives a value between 1 and 15. The value 0 is reserved by the PCALL16() command.

When receiving the SLOT\_MARKER() command, the SRIX4K compares its CHIP\_SLOT\_NUMBER value with the SLOT\_NUMBER value given in the command code. If it matches, the SRIX4K send back its Chip\_ID value. If not, the SRIX4K does not send any response.

This command, in addition to PCALL16(), is requested in the case there are more than one SRIX4K in the INVENTORY state available in the reader field range. It allows to search for all Chip\_ID available.

**Table 15. SLOT\_MARKER Request Format**

SOF	SLOT_MARKER	CRC <sub>L</sub>	CRC <sub>H</sub>	EOF
	x6h	8 bits	8 bits	

Request parameter:

- x: Slot number

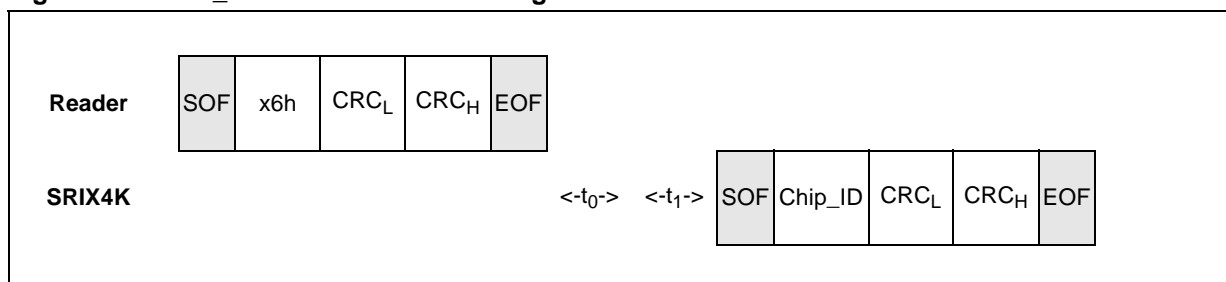
**Table 16. SLOT\_MARKER Response Format**

SOF	Chip_ID	CRC <sub>L</sub>	CRC <sub>H</sub>	EOF
	8 bits	8 bits	8 bits	

Response parameters:

- Chip\_ID of the SRIX4K.

**Figure 14. SLOT\_MARKER Frame exchange between Reader and SRIX4K**



**SELECT(Chip\_ID)**

Command Code = 0Eh

Prior to any other command, the SRIX4K must have been set in the SELECTED state by a SELECT() command. All other commands send to the SRIX4K before the SELECT() are ignored except INITIATE(), PCALL16() and SLOT\_MARKER(). The SELECT() command sends back the 8 bits of the Chip\_ID value. A SRIX4K in SELECTED state which receives a SELECT() command and does not match the Chip\_ID is automatically putted in the DESELECTED state.

**Table 17. SELECT Request Format**

SOF	SELECT	Chip_ID	CRC <sub>L</sub>	CRC <sub>H</sub>	EOF
	0Eh	8 bits	8 bits	8 bits	

Request parameter:

- 8 bits Chip\_ID get during the anti-collision sequence

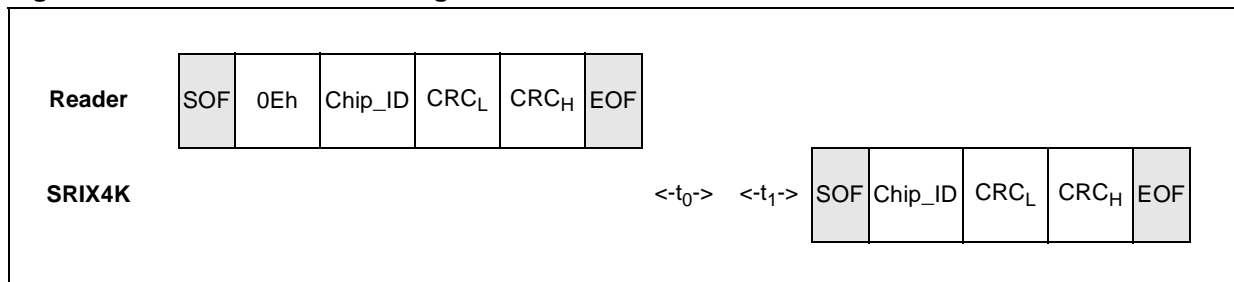
**Table 18. SELECT Response Format**

SOF	Chip_ID	CRC <sub>L</sub>	CRC <sub>H</sub>	EOF
	8 bits	8 bits	8 bits	

Response parameters:

- Chip\_ID of the selected tag. Must be equal to the transmitted Chip\_ID

**Figure 15. SELECT Frame exchange between Reader and SRIX4K**



## SRIX4K

---

### COMPLETION()

Command Code = 0Fh

When receiving the COMPLETION(), a SRIX4K in SELECTED state is put in the DESACTIVATED state and does not decode any new commands. The SRIX4K is then locked in this state and waits for a complete reset (tag out of the field range). This allows to access a new SRIX4K by a SELECT() without the needs of removing the previous one from the field. The COMPLETION() does not generate any response. All SRIX4K which are not in the SELECTED state ignore the COMPLETION() command.

**Table 19. COMPLETION Request Format**

SOF	COMPLETION	CRC <sub>L</sub>	CRC <sub>H</sub>	EOF
	0Fh	8 bits	8 bits	

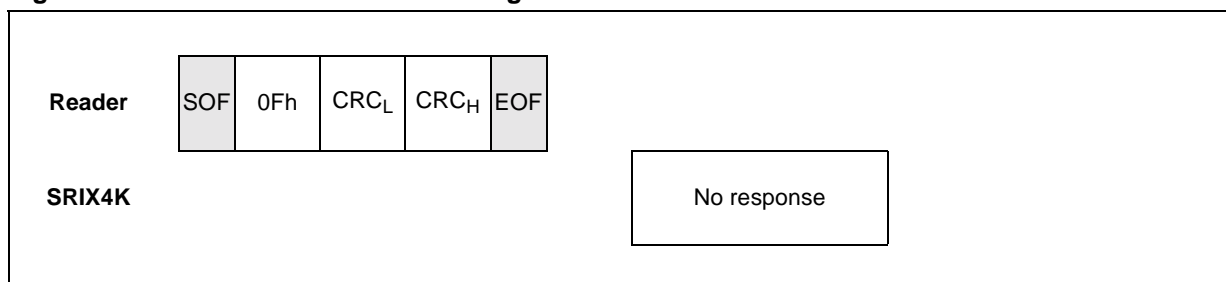
Request parameters:

- No parameter

**Table 20. COMPLETION Response Format**

No Response

**Figure 16. COMPLETION Frame exchange between Reader and SRIX4K**





**RESET\_TO\_INVENTORY()**

Command Code = 0Ch

When receiving the RESET\_TO\_INVENTORY(), SRIX4K in SELECTED state are put back in the INVENTORY state. This allows to re-introduce the SRIX4K into the anti-collision sequence. This command is usefull when two SRIX4K, with the same 8 bit Chip\_ID, are found in the SELECTED state at the same time. Forcing them to go back to the anti-collision sequence allows the reader to generates new PCALL16() commands and so to set new random Chip\_ID.

The RESET\_TO\_INVENTORY() does not generate any response.

All SRIX4K which are not in the SELECTED state ignore the RESET\_TO\_INVENTORY() command.

**Table 21. RESET\_TO\_INVENTORY Request Format**

SOF	RESET_TO_INVENTORY	CRC <sub>L</sub>	CRC <sub>H</sub>	EOF
	0Ch	8 bits	8 bits	

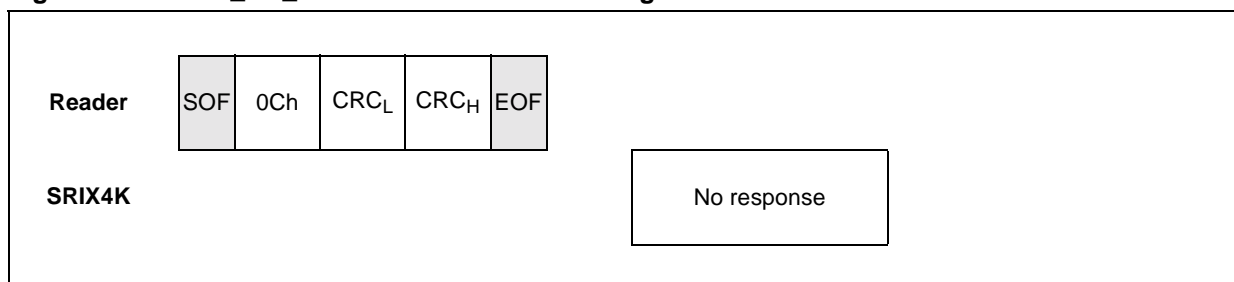
Request parameters:

- No parameter

**Table 22. RESET\_TO\_INVENTORY Response Format**

No Response

**Figure 17. RESET\_TO\_INVENTORY Frame exchange between Reader and SRIX4K**



## SRIX4K

### READ\_BLOCK(Addr)

Command Code = 08h

When receiving the READ\_BLOCK command, the SRIX4K read the requested block and send back its 4 data bytes value in the response. Data bytes are transmitted Least Significant Byte (LSByte) first and each byte are transmitted Least Significant Bit (LSBit) first.

The address byte allow to access to the 128 blocks of the SRIX4K (addresses between 0 to 127). All access above address 127 are not interpreted and the SRIX4K does not generate any response.

Prior to any READ\_BLOCK() command, the SRIX4K must have been set in the SELECTED state by a SELECT(). All commands send to the SRIX4K before the SELECT() are ignored.

**Table 23. READ\_BLOCK Request Format**

SOF	READ_BLOCK	ADDRESS	CRC <sub>L</sub>	CRC <sub>H</sub>	EOF
	08h	8 bits	8 bits	8 bits	

Request parameter:

- ADDRESS: address block from 0 to 127

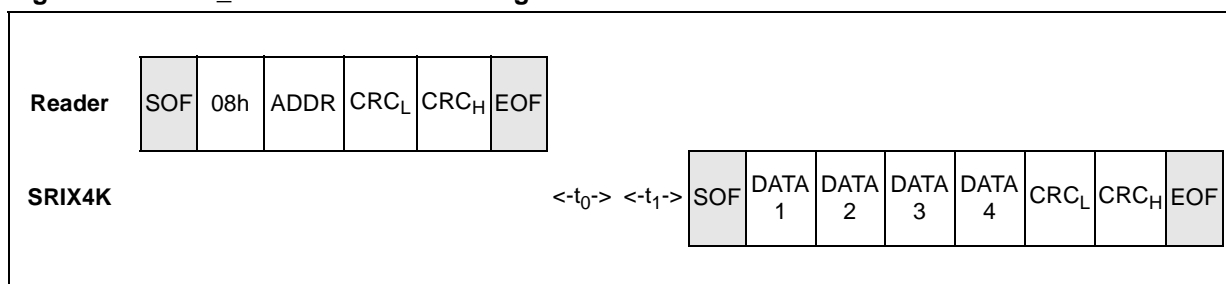
**Table 24. READ\_BLOCK Response Format**

SOF	DATA 1	DATA 2	DATA 3	DATA 4	CRC <sub>L</sub>	CRC <sub>H</sub>	EOF
	8 bits	8 bits	8 bits	8 bits	8 bits	8 bits	

Response parameters:

- DATA 1: Less significant data byte
- DATA 2: Data byte
- DATA 3: Data byte
- DATA 4: Most significant data byte

**Figure 18. READ\_BLOCK Frame exchange between Reader and SRIX4K**



**WRITE\_BLOCK(Addr, Data)**

Command Code = 09h

When receiving the WRITE\_BLOCK, the SRIX4K write the addressed block with the 4 bytes values send in the command if the block is available and not write protected. Data bytes are transmitted Least Significant Byte (LSByte) first and each byte are transmitted Least Significant Bit (LSBit) first.

The address byte allow to access to the 128 blocks of the SRIX4K (addresses between 0 to 127). All access above address 127 are not interpreted and the SRIX4K does not generate any response.

The WRITE\_BLOCK result is submitted to the addressed block. See the following paragraphs for complete description of WRITE\_BLOCK behaviour:

- Resettable EPROM (addresses 0 to 4) on page 9
- Binary Counter (addresses 5 to 6) on page 10
- EEPROM (addresses 7 to 127) on page 11

The WRITE\_BLOCK does not generate any response. The reader must check after the programming time  $t_W$  that datas were correctly programmed. Prior to any WRITE\_BLOCK command, the SRIX4K must have been set in the SELECTED state by a SELECT(). All commands send to the SRIX4K before the SELECT() are ignored.

**Table 25. WRITE\_BLOCK Request Format**

SOF	WRITE_BLOCK	ADDRESS	DATA 1	DATA 2	DATA 3	DATA 4	CRC <sub>L</sub>	CRC <sub>H</sub>	EOF
	09h	8 bits	8 bits	8 bits	8 bits	8 bits	8 bits	8 bits	

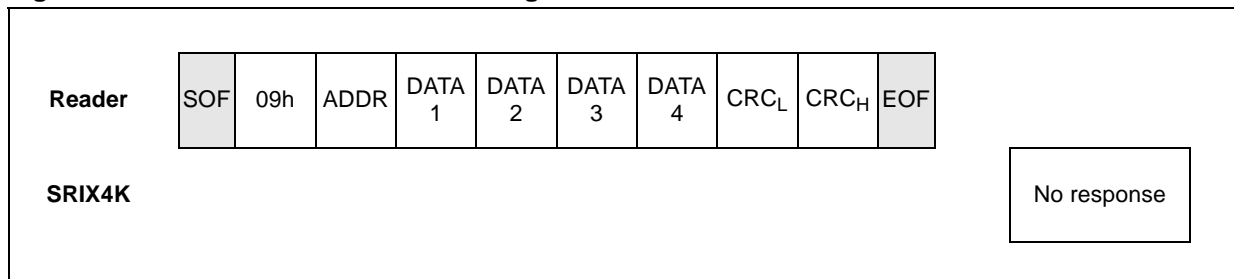
Request parameters:

- ADDRESS: address block from 0 to 127
- DATA 1: Less significant data byte
- DATA 2: Data byte
- DATA 3: Data byte
- DATA 4: Most significant data byte

**Table 26. WRITE\_BLOCK Response Format**

No Response

**Figure 19. WRITE\_BLOCK Frame exchange between Reader and SRIX4K**



## SRIX4K

### AUTHENTICATE(RND)

Command Code = 0Ah

AUTHENTICATE() compute the anti-clone algorithm provided in the SRIX4K. It uses a 6 data bytes as input and compute a 3 bytes signature.

Data bytes (Data) and signature bytes (SIG) are transmitted Least Significant Byte (LSByte) first and each byte are transmitted Least Significant Bit (LSBit) first.

Prior to any AUTHENTICATE() command, the SRIX4K must have been set in the SELECTED state by a SELECT(). All commands send to the SRIX4K before the SELECT() are ignored.

**Table 27. AUTHENTICATE Request Format**

SOF	AUTHENTICATE	Data 1	Data 2	Data 3	Data 4	Data 5	Data 6	CRC <sub>L</sub>	CRC <sub>H</sub>	EOF
	0Ah	8 bits	8 bits	8 bits	8 bits	8 bits	8 bits	8 bits	8 bits	

Request parameters:

- Data 1: Less significant Data byte
- Data 2, Data 3, Data 4, Data 5: Data byte
- Data 6: Most significant Data byte

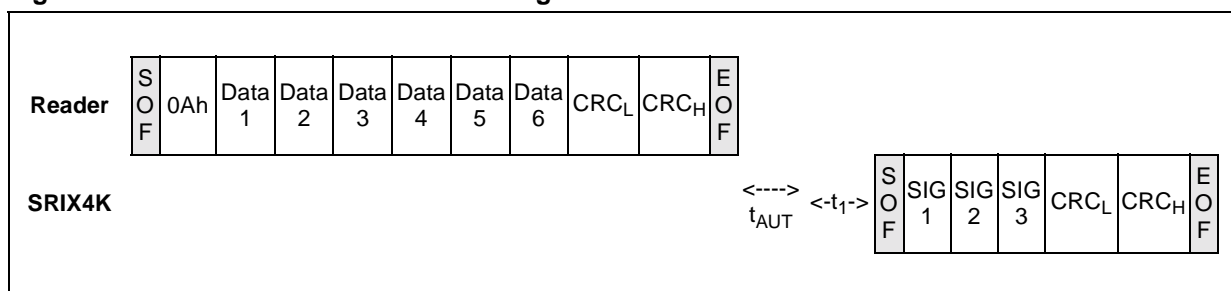
**Table 28. AUTHENTICATE Response Format**

SOF	SIG 1	SIG 2	SIG 3	CRC <sub>L</sub>	CRC <sub>H</sub>	EOF
	8 bits	8 bits	8 bits	8 bits	8 bits	

Response parameters:

- SIG 1: Less significant signature byte
- SIG 2: signature byte
- SIG 3: Most significant signature byte

**Figure 20. AUTHENTICATE Frame exchange between Reader and SRIX4K**



**GET\_UID()**

Command Code = 0Bh

When receiving the GET\_UID command, the SRIX4K send back its 8 UID bytes value in the response. UID bytes are transmitted Least Significant Byte (LSByte) first and each byte are transmitted Least Significant Bit (LSBit) first.

Prior to any GET\_UID() command, the SRIX4K must have been set in the SELECTED state by a SELECT(). All commands send to the SRIX4K before the SELECT() are ignored.

**Table 29. GET\_UID Request Format**

SOF	GET_UID	CRC <sub>L</sub>	CRC <sub>H</sub>	EOF
	0Bh	8 bits	8 bits	

Request parameter:

- No parameter

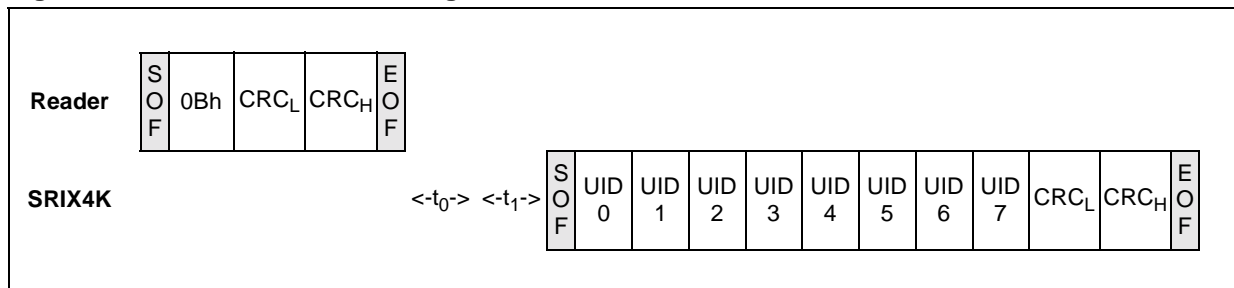
**Table 30. GET\_UID Response Format**

SOF	UID 0	UID 1	UID 2	UID 3	UID 4	UID 5	UID 6	UID 7	CRC <sub>L</sub>	CRC <sub>H</sub>	EOF
	8 bits	8 bits	8 bits	8 bits	8 bits	8 bits	8 bits	8 bits	8 bits	8 bits	

Response parameters:

- UID 0: Less significant UID byte
- UID 1 to UID 6: UID bytes
- UID 7: Most significant UID byte

**Figure 21. GET\_UID Frame exchange between Reader and SRIX4K**



**Power On State**

After a Power On, the SRIX4K is in the following state:

- The device is in the low power mode.
- The device is in the READY state.
- The device present its highest impedance to the reader antenna field.
- It will not answer to any command except INITIATE().

# SRIX4K

## SRIX4K COMMANDS BRIEF

Figure 22. INITIATE Frame exchange between Reader and SRIX4K

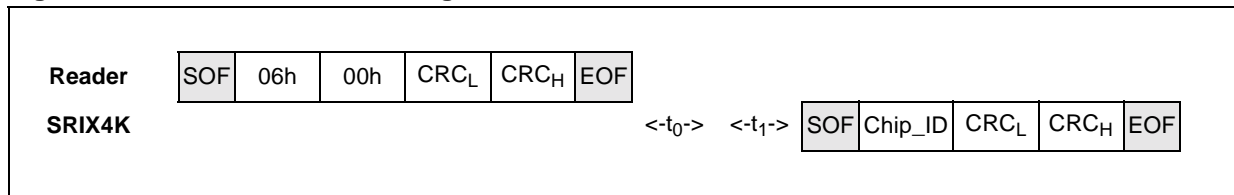


Figure 23. PCALL16 Frame exchange between Reader and SRIX4K

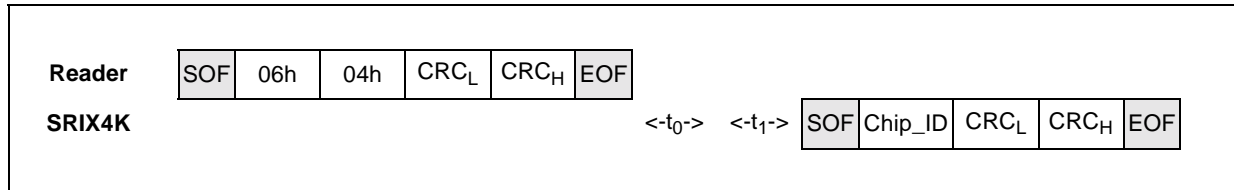


Figure 24. SLOT\_MARKER Frame exchange between Reader and SRIX4K

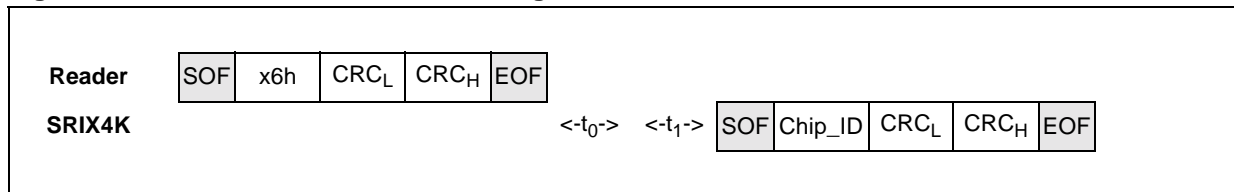


Figure 25. SELECT Frame exchange between Reader and SRIX4K



Figure 26. COMPLETION Frame exchange between Reader and SRIX4K

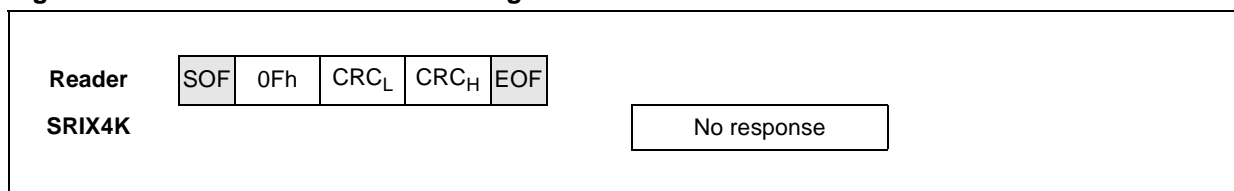
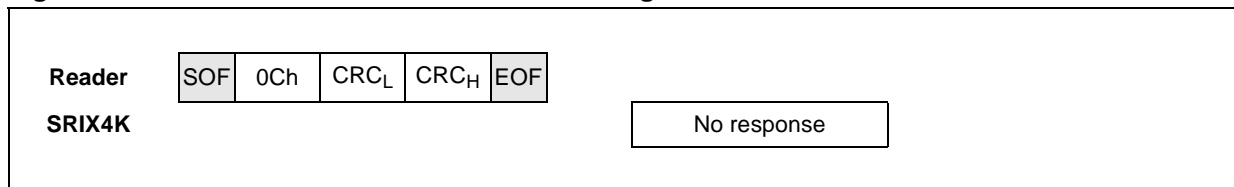
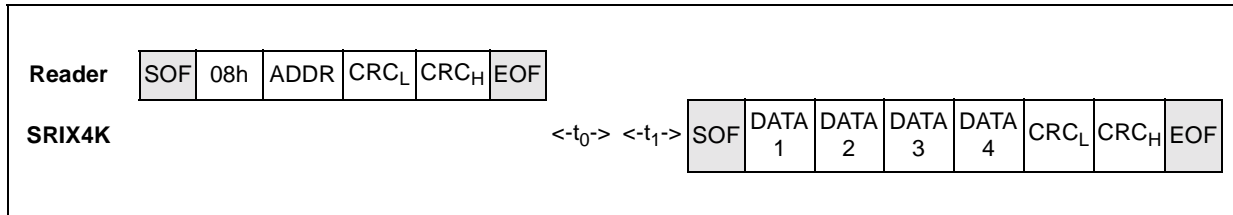


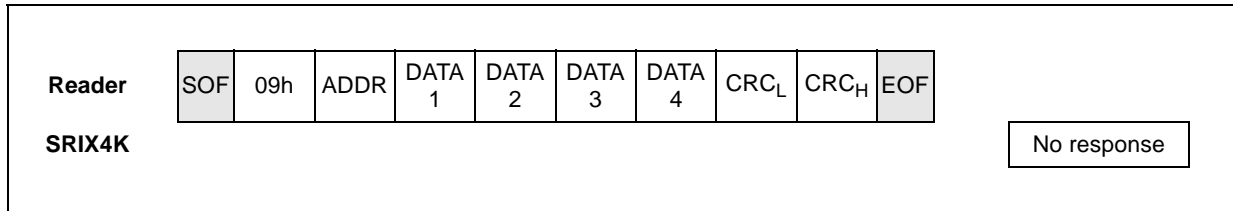
Figure 27. RESET\_TO\_INVENTORY Frame exchange between Reader and SRIX4K



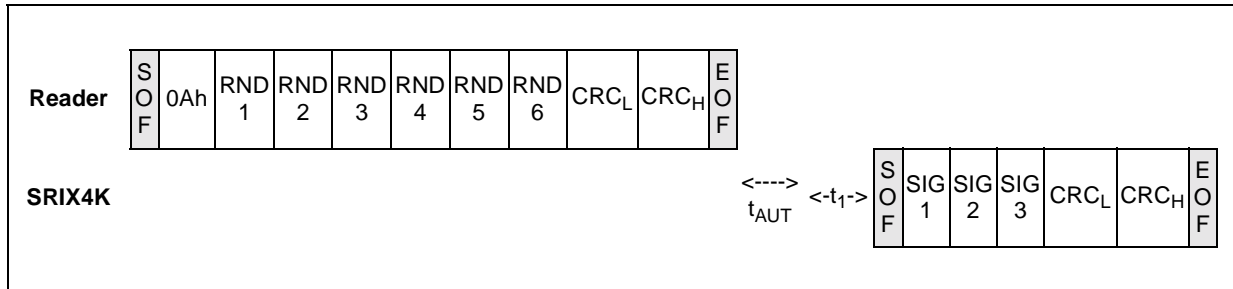
**Figure 28. READ\_BLOCK Frame exchange between Reader and SRIX4K**



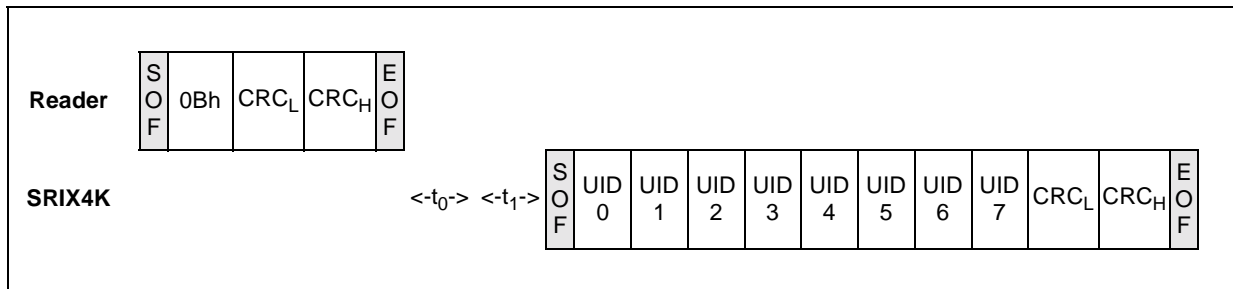
**Figure 29. WRITE\_BLOCK Frame exchange between Reader and SRIX4K**



**Figure 30. AUTHENTICATE Frame exchange between Reader and SRIX4K**



**Figure 31. GET\_UID Frame exchange between Reader and SRIX4K**



## ANNEX: ISO14443 TYPE B CRC CALCULATION

```
#include <stdio.h>
#include <stdlib.h>
#include <string.h>
#include <ctype.h>
#define BYTE unsigned char
#define USHORTunsigned short

unsigned short UpdateCrc(BYTE ch, USHORT *lpwCrc)
{
    ch = (ch^(BYTE)((*lpwCrc) & 0x00FF));
    ch = (ch^(ch<<4));
    *lpwCrc = (*lpwCrc >> 8)^((USHORT)ch << 8)^((USHORT)ch<<3)^((USHORT)ch>>4);
    return(*lpwCrc);
}

void ComputeCrc(char *Data, int Length, BYTE *TransmitFirst, BYTE *TransmitSecond)
{
    BYTE chBlock; USHORT wCrc;
    wCrc = 0xFFFF; // ISO 3309
    do
    {
        chBlock = *Data++;
        UpdateCrc(chBlock, &wCrc);
    } while (--Length);
    wCrc = ~wCrc; // ISO 3309
    *TransmitFirst = (BYTE) (wCrc & 0xFF);
    *TransmitSecond = (BYTE) ((wCrc >> 8) & 0xFF);
    return;
}

int main(void)
{
    BYTE BuffCRC_B[10] = {0x0A, 0x12, 0x34, 0x56}, First, Second, i;
    printf("Crc-16 G(x) = x^16 + x^12 + x^5 + 1");
    printf("CRC_B of [ ");
    for(i=0; i<4; i++)
        printf("%02X ",BuffCRC_B[i]);
    ComputeCrc(BuffCRC_B, 4, &First, &Second);
    printf("] Transmitted: %02X then %02X.", First, Second);
    return(0);
}
```



## PACKAGE DESCRIPTION

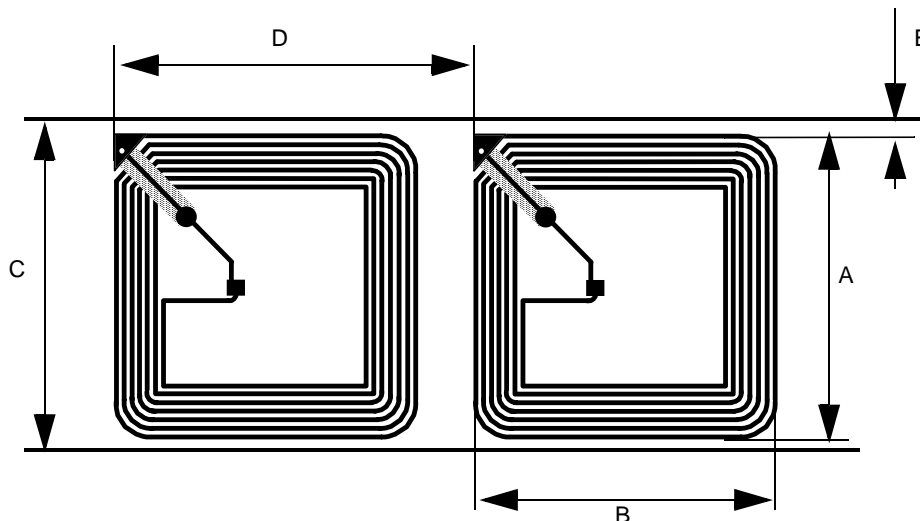
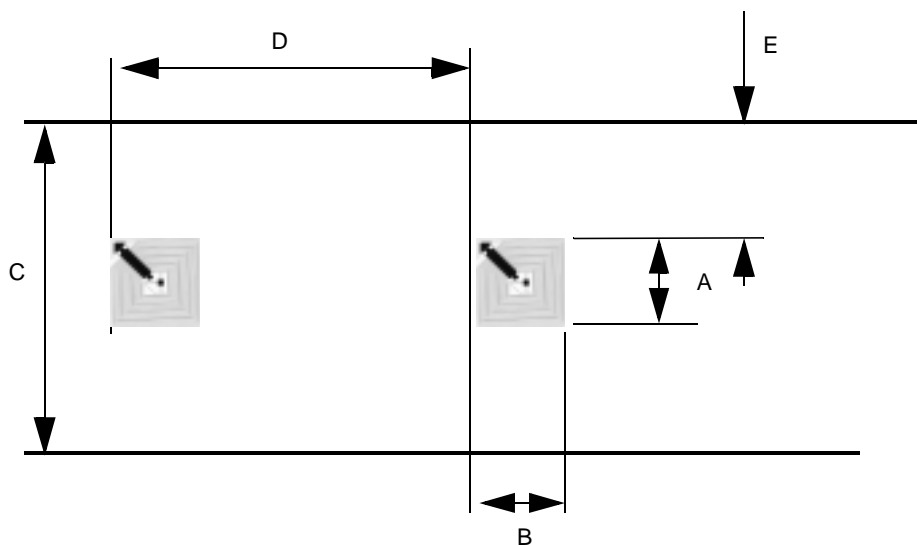


Table 31. 1/2 ISO Antenna specification

Symbol	Parameter	Type	Min	Max	Unit
A	Coil Width	45	44.5	45.5	mm
B	Coil Length	45	44.5	45.5	mm
C	Web Width	48	47.5	48.5	mm
D	Pitch				mm
E	Coil distance from Web edge	1.5	1	2	mm
	Overall Thickness of copper antenna coil	110	90	130	$\mu\text{m}$
	Silicon Thickness	180	165	195	$\mu\text{m}$
Q	Unloaded Q value	40			
F <sub>NOM</sub>	Unloaded free-air resonance	15.1			MHz

**PACKAGE DESCRIPTION**



**Table 32. 1/2 15x15mm Antenna specification**

Symbol	Parameter	Type	Min	Max	Unit
A	Coil Width	15			mm
B	Coil Length	15			mm
C	Web Width	48			mm
D	Pitch				mm
E	Coil distance from Web edge				mm
	Overall Thickness of copper antenna coil	110			μm
	Silicon Thickness	180			μm
Q	Unloaded Q value				
F <sub>NOM</sub>	Unloaded free-air resonance				MHz

## ORDERING INFORMATION SCHEME

Example

SRIX4K - W4 / XXX

<b>Delivery Form</b>
----------------------

W4	180µm ± 15µm unsawn wafer
A3T/S-ISOR	Half ISO Copper Antenna on tape
A4T/15R	15x15mm Copper Antenna on tape

For a full list of the available options, please see the current Memory Shortform Catalogue.

For further information on any aspect of this device, please contact your nearest ST Sales Office.

## CONFIDENTIALITY OBLIGATIONS

This document contains confidential information. Its distribution is subject to the signature of a non-disclosure agreement (NDA). It is classified "Restricted Distribution".

At all times you should comply with the following security rules (refer to the NDA for detailed obligations):

- Do not copy or reproduce all or part of this document
- Keep this document locked away

Further copies can be provided on a "need to know basis", please contact your local ST sales office, or the following address:

STMicroelectronics  
EEPROM and ASM Division  
Z.I. de Rousset  
B.P. 2  
13106 ROUSSET Cedex  
FRANCE

Fax number for confidential information enquiries: 04.42.68.87.93

E-mail: memories.contactless@st.com

Information furnished is believed to be accurate and reliable. However, STMicroelectronics assumes no responsibility for the consequences of use of such information nor for any infringement of patents or other rights of third parties which may result from its use. No license is granted by implication or otherwise under any patent or patent rights of STMicroelectronics. Specifications mentioned in this publication are subject to change without notice. This publication supersedes and replaces all information previously supplied. STMicroelectronics products are not authorized for use as critical components in life support devices or systems without express written approval of STMicroelectronics.

© 2002 STMicroelectronics - All Rights Reserved

STMicroelectronics GROUP OF COMPANIES

Australia - Brazil - China - France - Germany - Italy - Japan - Korea - Malaysia - Malta - Morocco - The Netherlands - Singapore - Spain - Sweden - Switzerland - Taiwan - Thailand - United Kingdom - U.S.A.

