SL2 ICS12

Functional Specification I•CODE UID-OTP Smart Label IC

1. General description

The I•CODE UID-OTP Smart Label IC is a dedicated chip for radio frequency identification on item level. This IC is a member of the well known and established in the contactless identification market I•CODE product family of smart label ICs.

The I•CODE system offers the possibility of operating smart labels simultaneously in the field of the reader antenna (Anticollision). It is designed for long range applications.

Due to the open marketing strategy of NXP Semiconductors there are various manufacturers available for both the smart labels as well as for the Read/Write devices (RWD).

1.1 Contactless energy and data transfer

Whenever connected to a very simple and cost effective type of antenna (as a result of the carrier frequency f_c = 13.56 MHz) made out of a few windings printed, winded, etched or punched coil the I•CODE UID-OTP Smart Label IC can be operated without any line of sight up to a distance of 1.5 m (gate width). No battery is needed. When the I•CODE UID-OTP smart label is positioned in the field of an RWD antenna, the high speed RF communication interface allows to transmit data with up to 53 kbit/s.

1.2 Anticollision

An intelligent anticollision function allows to operate many I•CODE UID-OTP smart labels in the field simultaneously. The anticollision algorithm enables that each smart label in the bulk of many labels can be identified individually.

1.3 Customer Application Support and Training

Accompanying Data Sheets and Application Notes:
http://www.nxp.com/products/identification/icode/index.html

2. Features

Integrated Circuit for Contactless Radio Frequency Identification Smart Label.

Integrated resonance capacitor of 23.5 pF with ± 5 % tolerance over full production.

2.1 I•CODE UID-OTP RF interface features

- Contactless transmission of data and supply energy (no battery needed)
- Operating distance: up to 1.5 m (depending on antenna geometry)
- Operating frequency: 13.56 MHz(ISM, world-wide licence free available)

NXP
founded by Philips

- Modulation Read/Write Device → Label: 10 % ASK
- Fast data transfer: up to 53 kbit/s
- High data integrity: 16-bit CRC, framing
- Anticollision with high identification speed (approx. 200 I•CODE UID-OTP smart labels per second)
- Label DESTROY command with 24-bit Destroy Code protection

2.2 Memory features

- 192 bits, organized in 24 blocks of 1 byte each
- Data retention of 5 years

3. Ordering information

See Delivery Type Addendum of Device

4. Block diagram

The I•CODE UID-OTP smart label requires no internal power supply. Its contactless interface generates the power supply and the system clock via the resonant circuitry by inductive coupling to the RWD. The interface also demodulates data that are transmitted from the RWD to the I•CODE UID-OTP smart label, and modulates the electromagnetic field for data transmission from the I•CODE UID-OTP smart label to the RWD.

Data are stored in a memory with a capacity of 192 bits which is organized in 24 blocks consisting of 1 byte each (1 block = 8 bits).
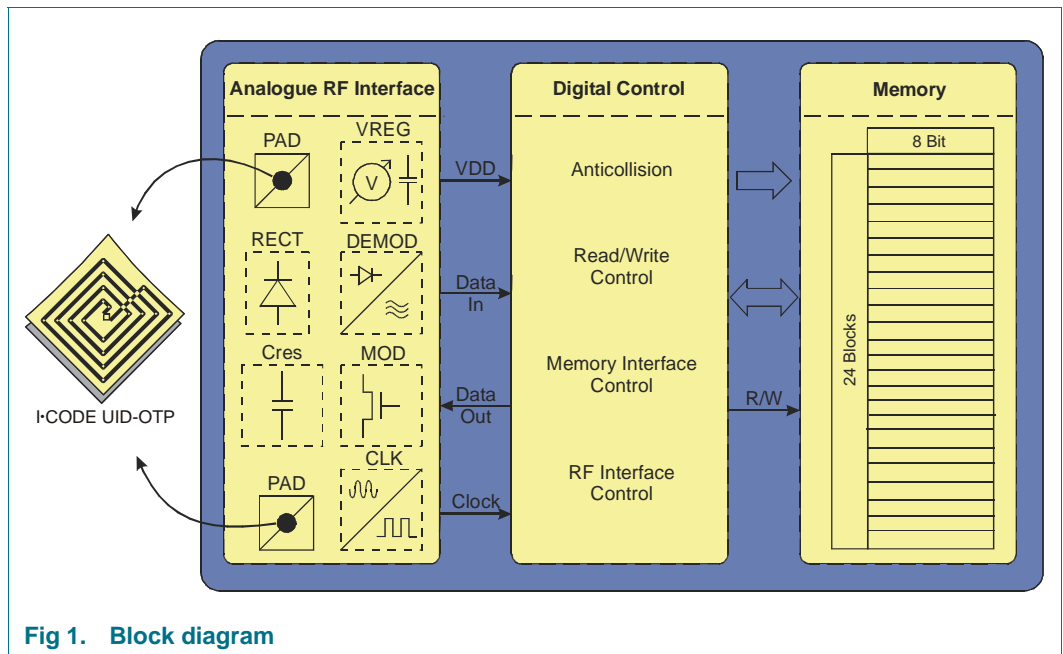

Fig 1. Block diagram

5. Pinning information

See Delivery Type Addendum of Device

6. Functional description

6.1 Memory organization

The memory has a capacity of 192 bits and is organized in 24 blocks, consisting of 1 byte each. All MSBytes of the different fields (UD, UD CRC 16, UID, CRC 16, Destroy Code) are located at the lowest block addresses of the corresponding memory blocks.
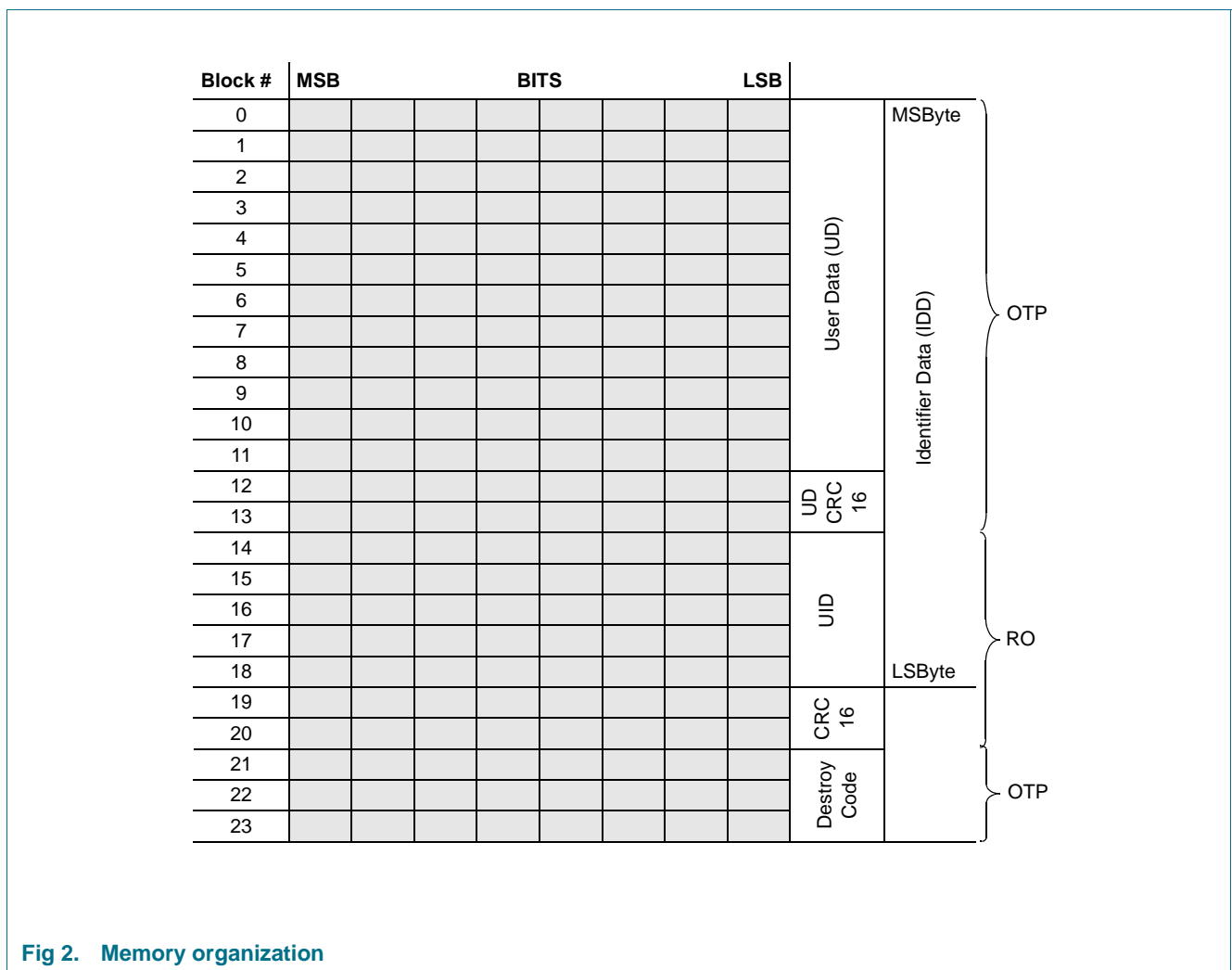
Fig 2. Memory organization

The I•CODE UID-OTP Smart Label IC memory contains, as shown in Figure 2, following elements:

- The 152-bit Identifier Data (IDD) consists of 96 bit User Data, 16-bit Cyclic Redundancy Check for the User Data (UD CRC 16) and 40-bit Unique Identifier (UID). The 40-bit UID is programmed by NXP Semiconductors.
- A 16-bit Cyclic Redundancy Check for the Unique Identifier (CRC 16).

- A 24-bit Destroy Code.

6.2 User data

The memory blocks 0 to 11 can be used to store user data.

6.3 User data cyclic redundancy check

To provide protection from undetected errors during the data transmission from the I•CODE UID-OTP smart Label to the Read/Write device a sixteen bit Cyclic Redundancy Check of the User Data (UD CRC 16) must be stored in the memory blocks 12 and 13. The CRC 16 algorithm is described in detail in Section 11.2.

6.4 Unique identifier

The 40-bit Unique Identifier (UID) is programmed during the production process and cannot be changed afterwards (Read Only = RO).

Table 1.   Unique identifier

MSByte		Unique Identifier		LSByte
Block 14	Block 15	Block 16	Block 17	Block 18

6.5 Cyclic redundancy check

A 16-bit Cyclic Redundancy Check (CRC 16) for the UID is located in the blocks 19 and 20 and is also programmed by NXP. The CRC 16 algorithm is described in detail in Section Section 11.2.

6.6 Destroy code

The 24-bit Destroy Code is located in the blocks 21 to 23 and is not programmed by NXP Semiconductors. The Destroy Code needs to be programmed during label personalization.

6.7 Configuration of delivered ICs

The I•CODE UID-OTP Smart Label ICs are delivered with following configuration by NXP:

- The User Data bytes are defined as 00 hex and One Time Programmable (OTP)
- In accordance with the User Data the UD CRC 16 is defined as 7B06 hex and One Time Programmable(OTP)
- The Unique Identifier is unique and Read Only (RO)
- The CRC 16 is defined according to the Unique Identifier and Read Only (RO)
- The Destroy Code is not defined and One Time Programmable (OTP)

6.8 Basic system configuration

The following block diagram shows the I•CODE system configuration.
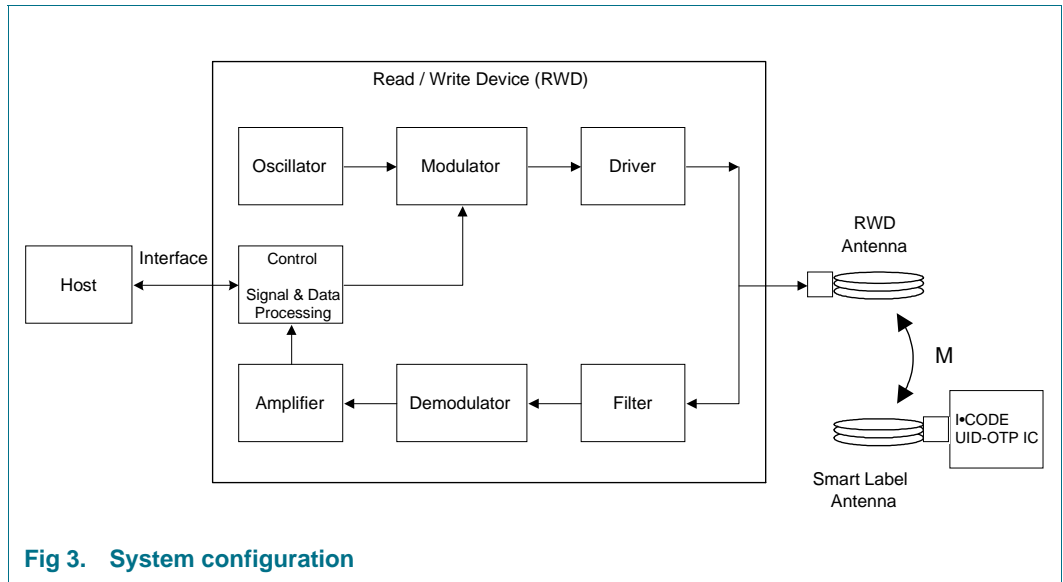
Fig 3. System configuration

A control and data processing unit controls the modulation of the carrier signal and processes data coming back from the demodulator circuit.

6.9 Energy transmission

Passive I•CODE smart labels must somehow be supplied with energy to be able to operate. In the I•CODE System, this is achieved by using the principle of a loose coupled transformer:

The RWD antenna generates a magnetic field. As shown in Figure 4 some of the generated magnetic flux passes through the I•CODE UID-OTP smart label antenna and induces a voltage there. The voltage drives a current and the I•CODE UID-OTP smart label will start operating. As this current will be very small when the smart label is far away from the antenna, the I•CODE UID-OTP Smart Label IC is designed for low power consumption.

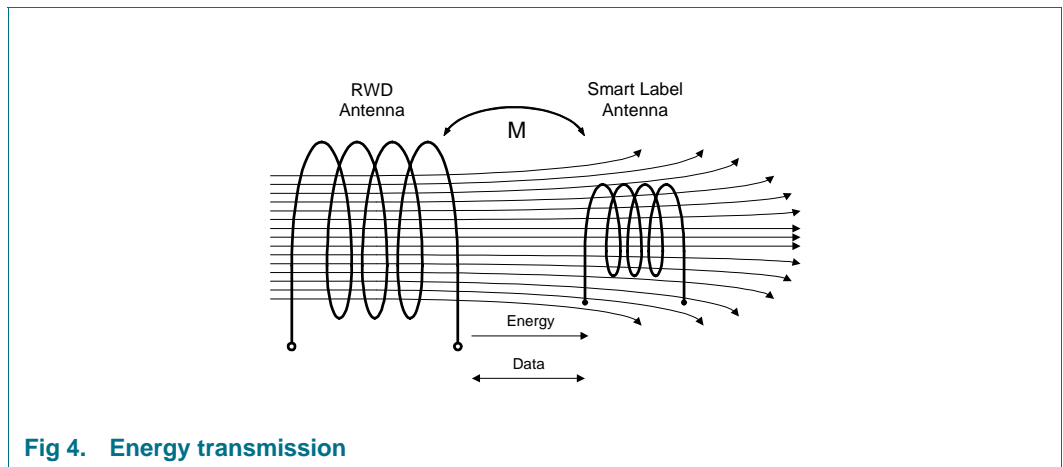The principle of a loose coupled transformer enables also a bi-directional data transmission.

Fig 4. Energy transmission

127730

Product data sheet **Rev. 03.00 — 6 December 2006** 5 of 33

6.10 Data transmission: I•CODE UID-OTP Smart label to RWD

For data transmission from the I•CODE UID-OTP smart label to the RWD, the implemented method is called 'load modulation'. Here the I•CODE UID-OTP smart label continuously changes the load on the magnetic field by switching on/off a resistor, according to the information to transmit.

The principle of load modulation is shown in Figure 5, which illustrates the label tuned circuit voltage during modulation. The black regions indicate regions of RWD carrier wave oscillations of period too small to be resolved.
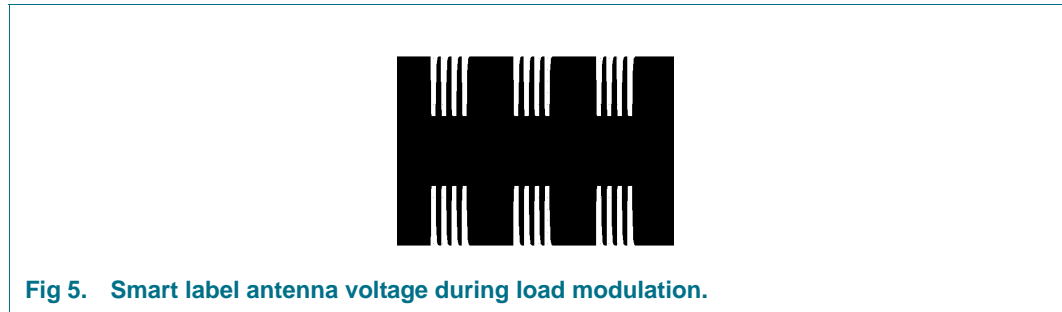

Fig 5.   Smart label antenna voltage during load modulation.

In the following description the physical state "modulator on" is named "high damping" and the physical state "modulator off" is named "low damping".

The receiver of the RWD detects the alteration of the magnetic field produced by the load modulation. The modulation ratio of the RWD antenna voltage depends on the coupling factor of the antenna configuration and is influenced by a lot of system design parameters such as the RWD antenna size, the smart label antenna size and the distance between the antennas.

6.10.1 Coding

A logic 0 starts with 4 pulses of $f_c/32$ (~423.75 kHz) sub-carrier, followed by an un-modulated time of $128/f_c$ (~9.44 ms), as shown in following Figure 6.
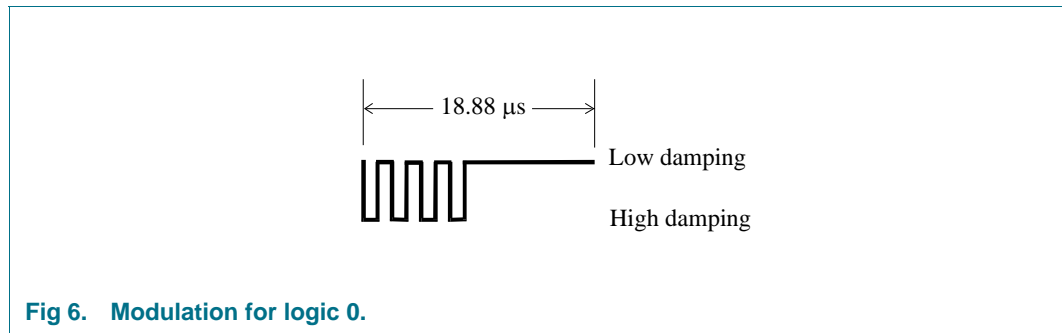

Fig 6.   Modulation for logic 0.

The waveforms in Figure 6, and similar diagrams to follow, represent logic states illustrating the periods of high damping and low damping of the antenna tuned circuit. They are not amplitudes of signals, but high oscillation amplitude will in fact occur during the high parts of the waveforms.

A logic 1 starts with an un-modulated time of $128/f_c$ (~9.44 μs) followed by 4 pulses of $f_c/32$ (~423.75 kHz) sub-carrier, as shown in following Figure 7.

Fig 7.   Modulation for logic 1.
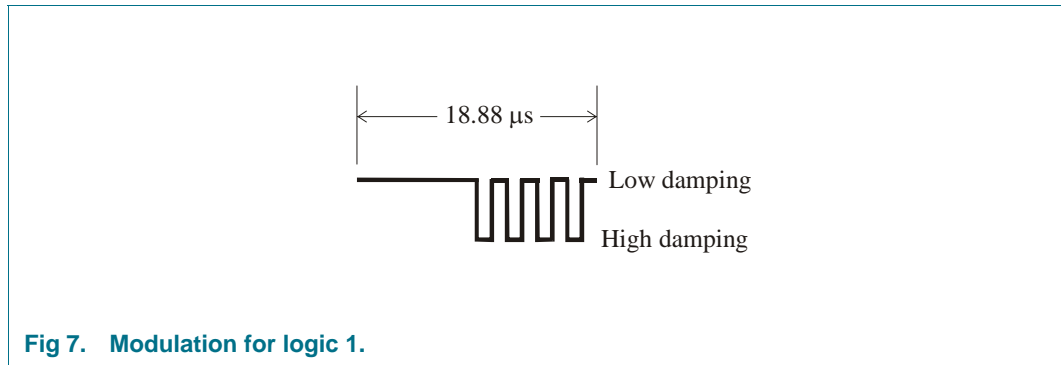
6.10.2   Baud rate

The baud rate in the I•CODE UID-OTP smart label to RWD link is 52.969 kbit/s (f_c/256).

6.10.3   Reply framing

The Reply Start of Frame (RSOF) starts with 12 pulses of sub-carrier of f_c/32 (~423.75 kHz), followed by an un-modulated time of 128/f_c (~9.44 µs) as shown in Figure 8. The significance of high levels and low levels in the diagram are the same as for Figure 6 and Figure 7.

Fig 8.   Reply start of frame (RSOF).
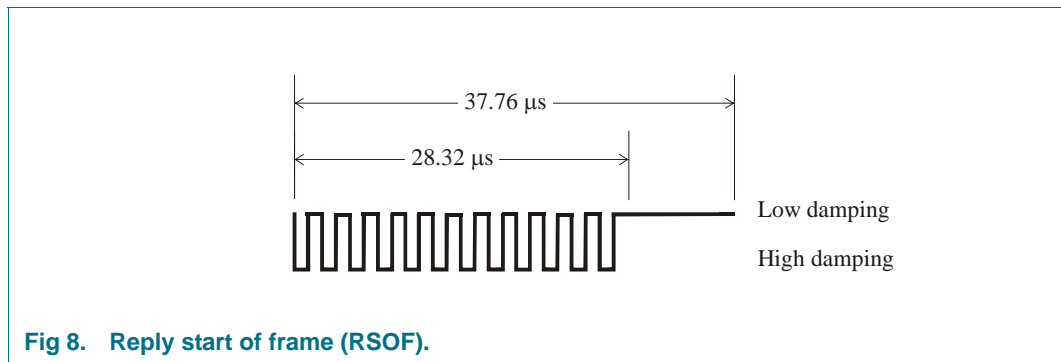
The Reply End of Frame (REOF) shown in Figure 9 starts with an un-modulated time of 128/f_c (~9.44 µs), followed by 12 pulses of sub-carrier of f_c/32 (~423.75 kHz).

Fig 9.   Reply start of frame (REOF).
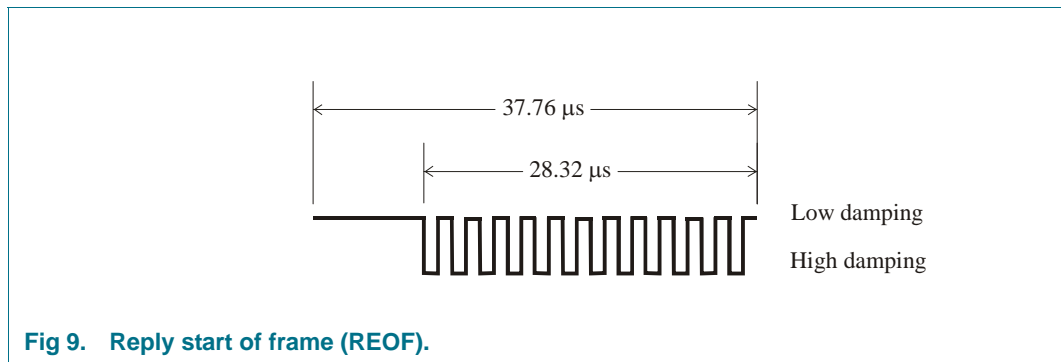
6.11   Data transmission: RWD to I•CODE UID-OTP smart label

Data are transmitted to the I•CODE UID-OTP smart label using Amplitude Shift Keying (ASK) modulation with a Modulation Index m between 10 % and 30 %.

6.11.1 Unitary pulse

All of the signalling from the RWD to the I•CODE UID-OTP smart label makes use of the Unitary Pulse with the nominal width of 9.44 μs, of which two instances are shown in Figure 10 below.

The pulse is described in terms of the Modulation Index m defined as: $m = (a - b)/(a + b)$.

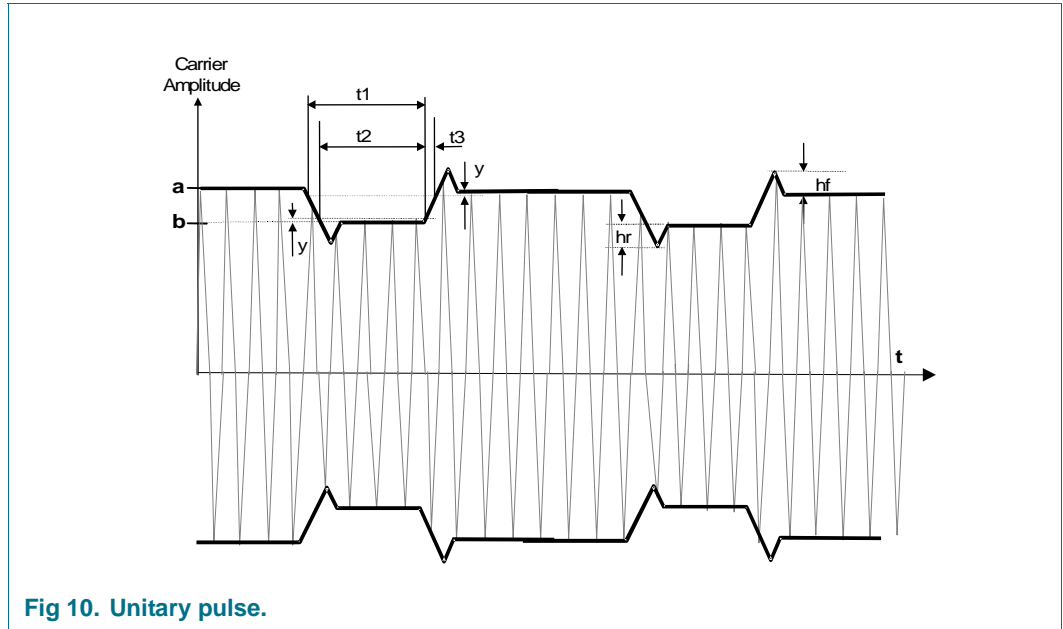$y = 0.05\ (a-b)$, $h_{r,max} = h_{f,max} = 0.1\ (a-b)$

Fig 10. Unitary pulse.

Table 2.   Unitary Pulse Parameter

SYMBOL	MIN	MAX	UNIT
m	10	30	%
t_1	6	9.44	μs
t_2	3	t_1	μs
t_3	0	4.5	μs

6.11.2 Baud rate

The baud rate in the RWD to I•CODE UID-OTP smart label link is 26.48 kbit/sec (f_c/512).

6.11.3 Set of symbols

All RWD to I•CODE UID-OTP smart label signalling is composed of symbols consisting of Unitary Pulse sequences as shown in Figure 11.

127730

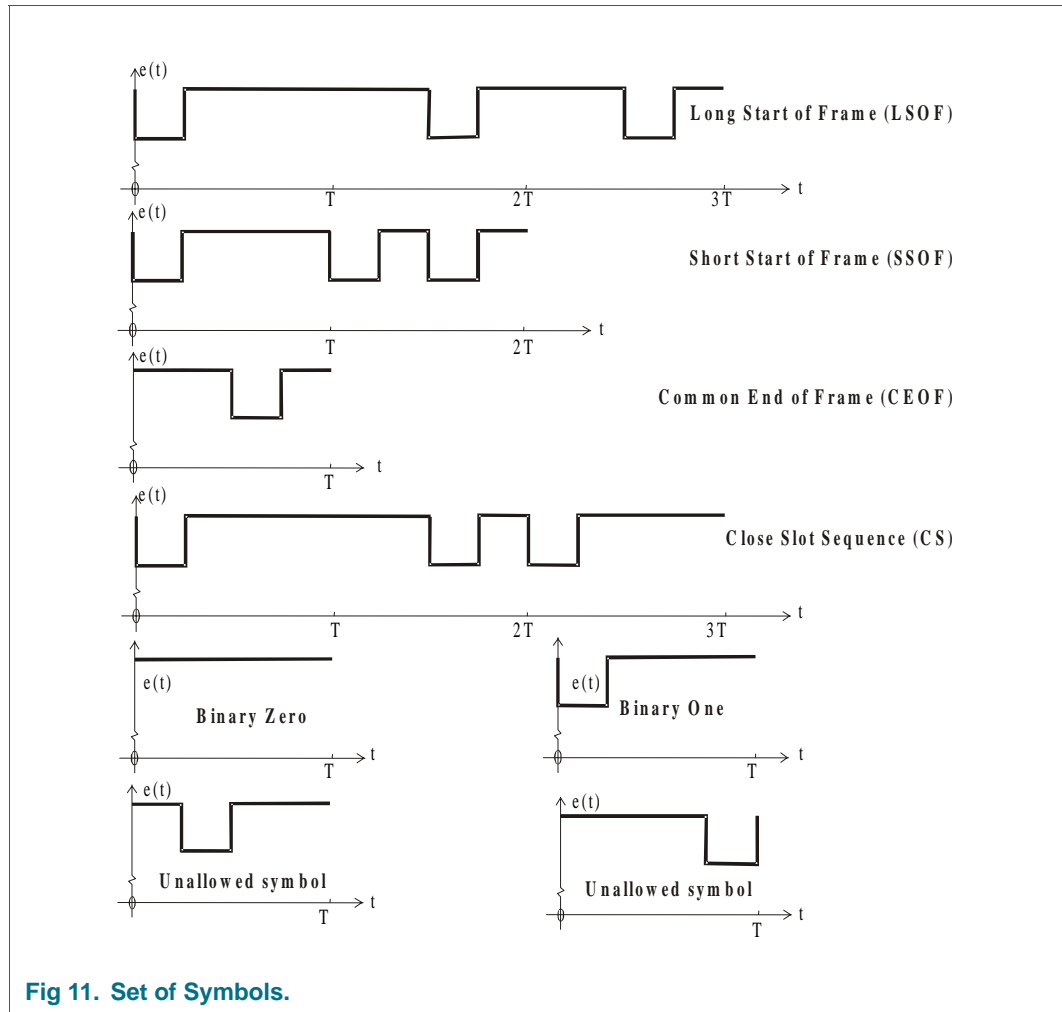**Product data sheet** **Rev. 03.00 — 6 December 2006** **8 of 33**

Fig 11. Set of Symbols.

The set of symbols consist of

- A Long Start of Frame that is used for beginning of some commands.
- A Short Start of Frame used for beginning of other commands.
- A Common End of Frame used for ending of all commands.
- A special Close Slot Sequence that is used for closing slots from which no information was collected.
- Binary symbols for Zero and One used within commands.

These symbols are built up into commands as described in Section 10 .

6.11.4 Symbol coding

The leading edges of all these pulses in such a pulse sequence are at a multiple of a basic time of exactly $128/f_c$ (~ 9.44 μs) from one another.

Binary symbols (Binary Zero and Binary One) and the Common End of Frame have a duration of $T = 512/f_c$ (~ 37.76 μs).

The Short Start of Frame has a duration of $2T = 1024/f_c$ (~ 75.52 μs).

127730

Product data sheet **Rev. 03.00 — 6 December 2006** **9 of 33**

The Long Start of Frame and the Close Slot Sequence have a duration of 3T = 1536/f_c (~ 113.27 μs).

## 6.12 Collision detection

6.12.1 Concept

The used collision detection methodology depends upon the facts that:

- the data content of all replying I•CODE UID-OTP smart labels will be different.
- the bit boundaries in the reply signals from all simultaneously replying I•CODE UID-OTP smart labels will be closely aligned.
- each bit period contains a period of no modulation, positioned in a data dependent way.

In consequence, two simultaneously replying I•CODE UID-OTP smart labels will exhibit in the receiver of the RWD the interference pattern as illustrated in the Figure 12 below.

6.12.2 Illustration

Fig 12. Illustrations of reply collision.
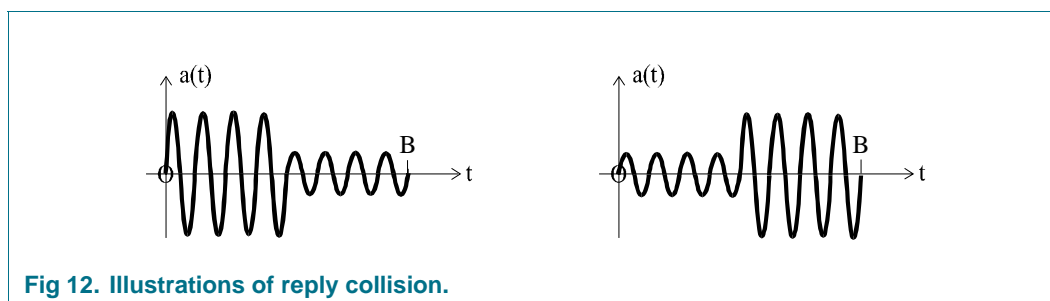
The presence of modulation in both halves of the reply period can be detected by the RWD.

The dynamic range of reliable detection is suitably large when the four cycles of sub-carrier occupy each half of the bit period B = 256/f_c (~18.88 μs).

7. Protocol

7.1 General description

The I•CODE UID-OTP uses the Slotted Terminating Adaptive Collection (STAC) protocol which is a Reader Talks First (RTF) protocol. In the STAC protocol, smart labels reply with randomly selected positions or time intervals referred to as slots, which have their beginning and end under RWD control. An RWD command signals both the end of the current and the beginning of the next slot. A number of slots form a Reply Round.
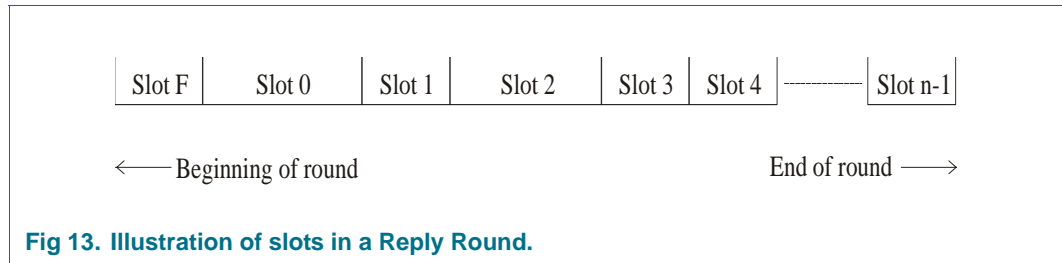
Fig 13. Illustration of slots in a Reply Round.

In relation to Figure 13, it is noted that:

- The figure is not to scale, but does correctly indicate that slots are not necessarily of equal sizes.
- There is a special slot (Slot F) at the beginning of a Reply Round. This slot is followed by n further slots to complete the Reply Round.
- The number n of further slots is a power of two and is determined by the RWD.
- Slot F is of a special and fixed size, but the duration of other slots is determined by RWD signalling.
- Slots, which contain no reply, can be closed early by the RWD.

The issuing of a BEGIN ROUND command starts the Reply Round and causes a subset of the I•CODE UID-OTP smart labels, which were waiting in the READY state, to enter the SLOTTED READ state (please refer to Chapter Section 9). Those I•CODE UID-OTP smart labels calculate for themselves a proposed reply slot. Each I•CODE UID-OTP smart label is equipped with a slot counter. The slot counters of all I•CODE UID-OTP smart labels in the SLOTTED READ state will advance each time the RWD indicates the end of a slot and the beginning of a new one. As soon as the slot counter reaches the proposed reply slot position the related I•CODE UID-OTP smart label will reply during this slot.

The reply conditions within a slot can then be separated into following three categories:

- no smart label reply present
- one smart label reply present
- two or more smart label replies present.

In first case above the RWD detects that no smart label reply is present and may close that empty slot by issuing a Close Slot Sequence which signals to all smart labels in the SLOTTED READ state to increment their current slot number.

In the second and third cases above it will be clear to the RWD that one or more smart labels are replying, and the RWD will continue to keep the slot open for a time sufficient for the reply or replies to conclude and be evaluated.

The evaluation may take several forms. One of these is based on a special feature of reply coding described in Section 6.12, and makes the detection of collisions, when they occur, highly probable.

Another of these is the checking of a CRC 16 present in the reply, against an expected value that may be calculated from information present in the RWD to smart label and smart label to RWD signalling. More detail of the checking is provided in Section 11.2.

If there is evidence from these checks that the smart label data has not been correctly collected, the slot is closed by a Close Slot Sequence as already described for the case of an empty slot. All those smart labels, which have replied in that closed slot, return to the READY state.

If however it appears that the smart label information has been correctly collected, the slot is closed in another way. In this case the RWD issues a FIX SLOT command, described in Section 10.3 . The effect of this command is to move the smart label from the SLOTTED READ state to the FIXED SLOT state. In the FIXED SLOT state the smart label replies once per Reply Round always in the fixed slot F. The smart labels in the Slot F reply only with the RSOF, as described in Section 8.1.

7.2 I•CODE UID-OTP smart label selection

The I•CODE UID-OTP Smart Label IC supports a selection feature in which groups of I•CODE UID-OTP smart labels may be selected. Therefore the BEGIN ROUND command includes the Selection Mask as an optional parameter.

While the BEGIN ROUND command is being processed by all I•CODE UID-OTP smart labels in the READY state, the Selection Mask is checked against the IDD content on a bit by bit basis, Most Significant Bit (MSB) first. In case of a match the related I•CODE UID-OTP smart label is selected and moves from the READY state to the SLOTTED READ state.

I•CODE UID-OTP smart labels, which have a mis-match in one or more Selection Mask bits, are not selected.

When no effort is made to select I•CODE UID-OTP smart labels, as a result of the BEGIN ROUND command not having a Selection Mask parameter, all I•CODE UID-OTP smart labels in a READY state are automatically selected and move into the SLOTTED READ state.

7.3 Truncated reply

As shown in Figure 14 a selected I•CODE UID-OTP smart label replies with the remaining contents of its memory, rounded up to an integral number of bytes (omitting all or most that part which participated in the selection process) and a CRC 16 that is calculated over the UID (memory blocks 14 to 18).

Fig 14. Illustration of selection and reply.
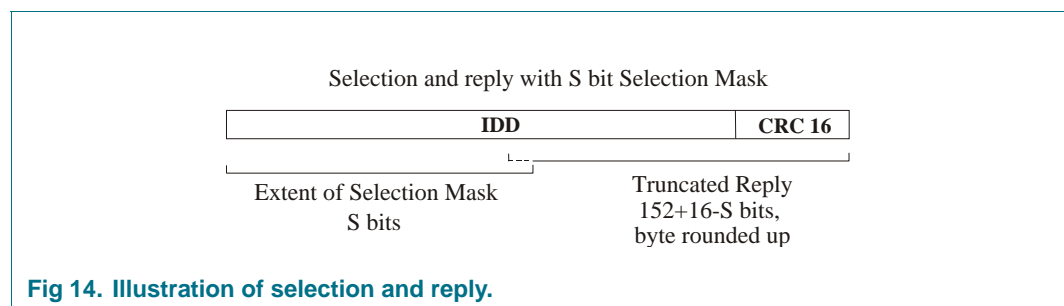
The Truncated Reply starts with the MSB at the preceding byte border of the truncated IDD and stops with the LSB of the CRC 16.

In case $S = 152$ bits only the CRC 16 is replied.

An advantage of the above selection implementation is that the smart label reply time is reduced as the number of bits transmitted is less than a full reply.

8. Protocol timing

8.1 Reply round timing

The following figure shows the timing within a Reply Round. Note that the diagram is not to scale, but does show the elements that make up a slot.
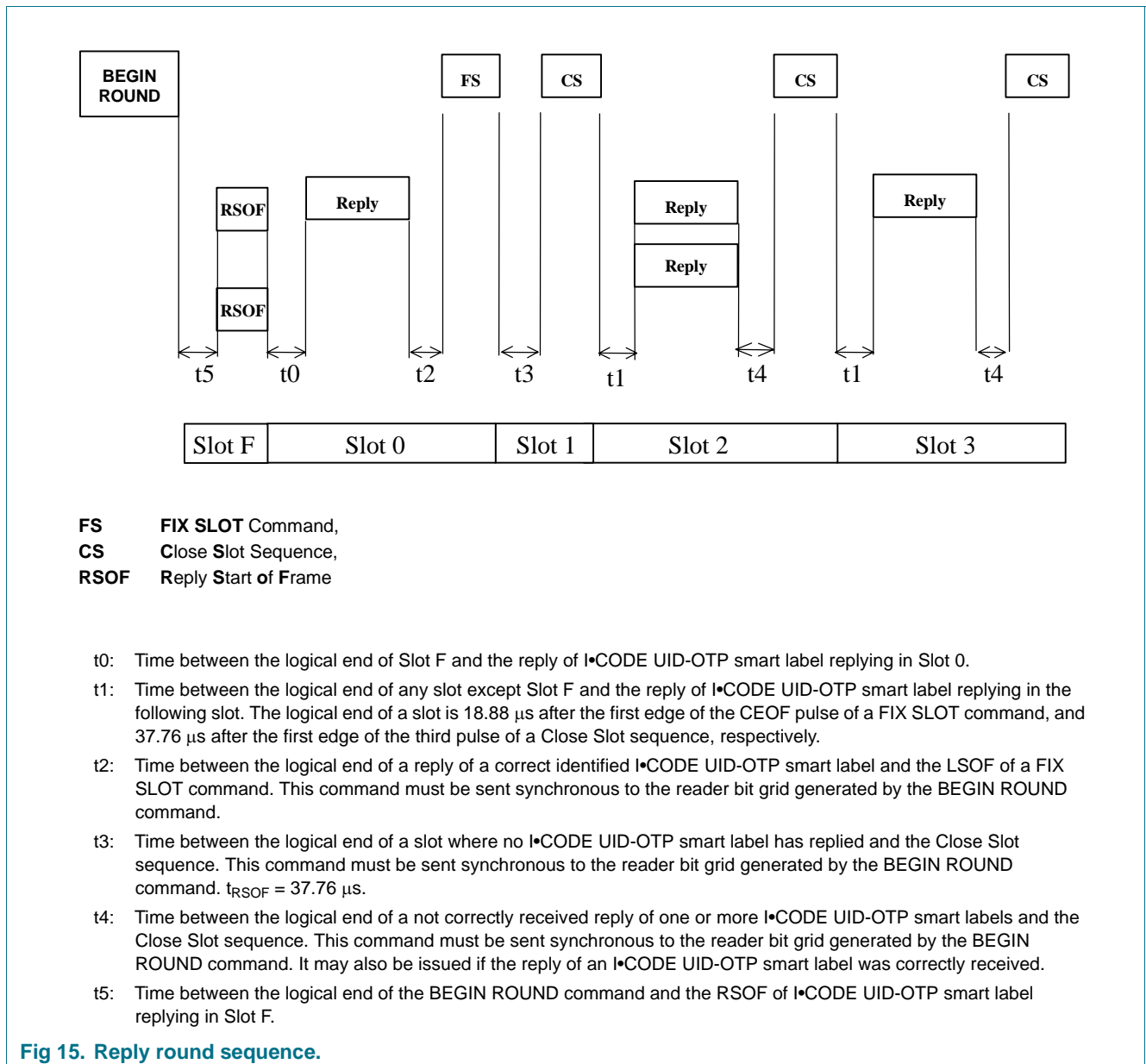

FS   **FIX SLOT** Command,
CS   **C**lose **S**lot Sequence,
RSOF   **R**eply **S**tart **o**f **F**rame

t0:   Time between the logical end of Slot F and the reply of I•CODE UID-OTP smart label replying in Slot 0.

t1:   Time between the logical end of any slot except Slot F and the reply of I•CODE UID-OTP smart label replying in the following slot. The logical end of a slot is 18.88 μs after the first edge of the CEOF pulse of a FIX SLOT command, and 37.76 μs after the first edge of the third pulse of a Close Slot sequence, respectively.

t2:   Time between the logical end of a reply of a correct identified I•CODE UID-OTP smart label and the LSOF of a FIX SLOT command. This command must be sent synchronous to the reader bit grid generated by the BEGIN ROUND command.

t3:   Time between the logical end of a slot where no I•CODE UID-OTP smart label has replied and the Close Slot sequence. This command must be sent synchronous to the reader bit grid generated by the BEGIN ROUND command. t_{RSOF} = 37.76 μs.

t4:   Time between the logical end of a not correctly received reply of one or more I•CODE UID-OTP smart labels and the Close Slot sequence. This command must be sent synchronous to the reader bit grid generated by the BEGIN ROUND command. It may also be issued if the reply of an I•CODE UID-OTP smart label was correctly received.

t5:   Time between the logical end of the BEGIN ROUND command and the RSOF of I•CODE UID-OTP smart label replying in Slot F.

Fig 15. Reply round sequence.

Table 3.   Reply Round Timing.

SYMBOL	MIN	MAX	UNIT
t0	113.27 - 2.36	113.27 + 2.36	μs
t1	302.06 - 2.36	302.06 + 2.36	μs
t2	302.06 - 2.36	302.06 + 2.36	μs
t3	377.58	Infinite	μs
t4	302.06 - 2.36	Infinite	μs
t5	151.03 - 2.36	151.03 + 2.36	μs

Remark: All timings refer to the logical end of the preceding command or symbol. The beginning of all RWD signals must be synchronous with the RWD bit grid which is at intervals of $512/f_c = 37.76$ μs.

Table 4.   Timings

302.06 μs is equal to $4096/f_c$ 151.03 μs is equal to $2048/f_c$

113.27 μs is equal to $1536/f_c$ 18.88 μs is equal to $256/f_c$

2.36 μs is equal to $32/f_c$

8.2 Start-up time

After switching on the powering field, the RWD has to wait at least the minimum time t_{wfc} = 1 μs before sending the first command.

8.3 Reset time

The powering field must be switched off for at least t_{reset} = 5 μs to generate a reset of the I•CODE UID-OTP smart label and to bring the label into the UNPOWERED State.

8.4 Waiting time after reply round sequence

After the logical end of a Reply Round Sequence, the RWD has to wait at least the minimum time t_{wsc} = 302.06 μs before sending a subsequent command.

8.5 Waiting time after WRITE/DESTROY

After the RWD has sent a WRITE command or a DESTROY command, the RWD has to wait at least the minimum time t_{wwr} = 6.4 μs before sending a subsequent command or switching off the RF field.

9. State Diagram

The following figure shows the State Diagram of the I•CODE UID-OTP smart label IC.

Fig 16. State Diagram
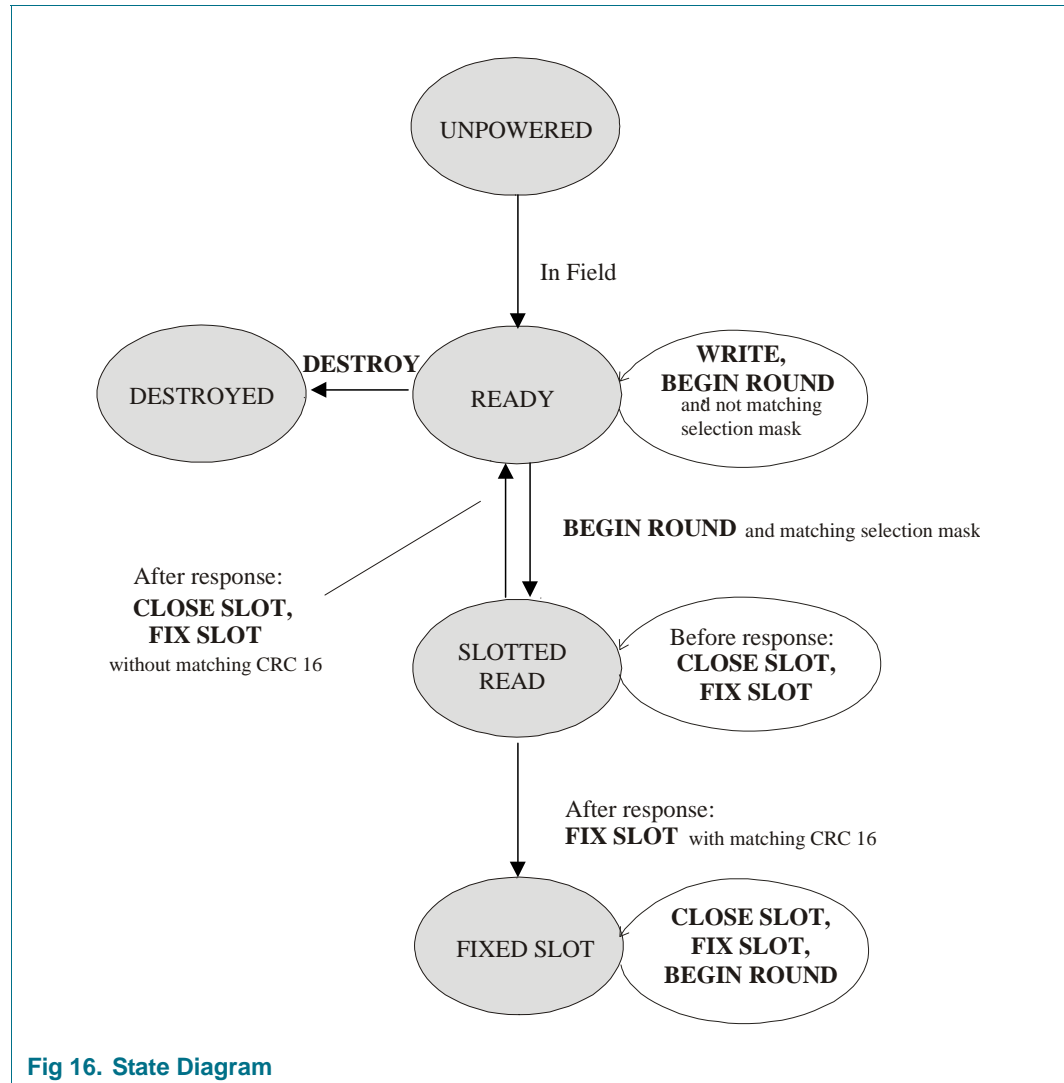
9.1 General description of states

9.1.1 Unpowered

The powering magnetic field of the RWD is switched off or the I•CODE UID-OTP smart label is out of field. In case of a smart label leaves the field and is not powered anymore, this smart label moves from all states except the DESTROYED state into the UNPOWERED state.

9.1.2 Ready

After start-up phase, the I•CODE UID-OTP smart label is ready to receive commands such as WRITE, DESTROY and BEGIN ROUND.

9.1.3 Slotted Read

The I•CODE UID-OTP smart label enters this state after the BEGIN ROUND command with a matching Selection Mask. In this state the I•CODE UID-OTP smart label calculates a reply slot and waits until its slot counter reaches this reply slot position, whereupon the I•CODE UID-OTP smart label will reply during the reply slot.

Until this reply slot is reached the slot counter is increased by one due to a Close Slot Sequence or a FIX SLOT command.

After its reply the I•CODE UID-OTP smart label can be moved back to the READY state by an issued Close Slot Sequence or an issued FIX SLOT command with a not matching CRC 16.

9.1.4 Fixed slot

The I•CODE UID-OTP smart label moves from the SLOTTED READ state to the FIXED SLOT state after a FIX SLOT command with a matching CRC 16. In this state the smart label replies once per Reply Round always in the fixed slot F only with the RSOF. The smart label stays in that state as long as it is powered by the RF field.

9.1.5 Destroyed

The I•CODE UID-OTP smart label moves from the READY state to the DESTROYED state after a DESTROY command with a matching Destroy Code. The smart label stays in this state permanently.

10. Command set

All bytes are transmitted MSB first.

10.1 Begin round command

10.1.1 Structure

The structure and parameters of the BEGIN ROUND command are defined in <u>Figure 17</u> below.

SSOF	Command Code	Mask Length	Selection Mask	Number of Slots	CRC 8	CEOF

Fig 17. BEGIN ROUND command.

The parameters of the BEGIN ROUND command are defined as follows:

- The Command Code has 8 bits and value of 30 hex.
- The Mask Length parameter is always present and has 8 bits. It specifies the number of bits included in the selection mask. If the value is zero, the selection mask is not present.

127730

Product data sheet **Rev. 03.00 — 6 December 2006** 16 of 33

- The Selection Mask parameter is present if the mask length is non-zero; otherwise it is absent. When present, it specifies the bits to be compared with the I•CODE UID-OTP smart label Identifier Data in the selection process. If all bits of the Selection Mask match, I•CODE UID-OTP smart label will be selected and will respond to a BEGIN ROUND command.

The Number of Slots parameter is always present and has 8 bits.

The coding of the number of slots is as follows:

1 slot:	00 hex	64 slots:	1F hex
4 slots:	01 hex	128 slots:	3F hex
8 slots:	03 hex	256 slots:	7F hex
16 slots:	07 hex	512 slots:	FF hex
32 slots:	0F hex		

The CRC 8 parameter has 8 bits and is calculated over the Command Code, the Mask Length parameter, the Selection Mask parameter (if present) and the Number of Slots parameter as described in Section Section 11.1.

10.1.2 I•CODE UID-OTP smart label reaction

All I•CODE UID-OTP smart labels will calculate a response position in the range of the Reply Round size, excluding Slot F. When the slot count maintained within the I•CODE UID-OTP smart label reaches that count, the I•CODE UID-OTP smart label will reply as described in Section 7.3 .

RSOF	Truncated Reply	REOF
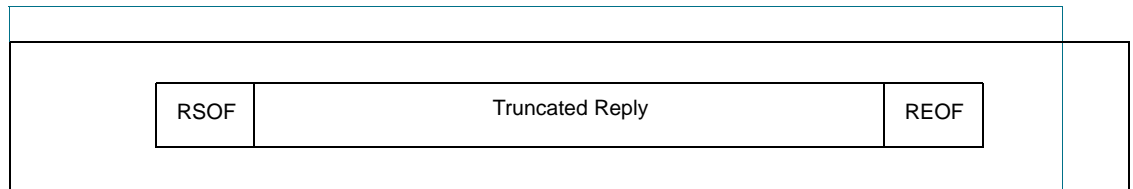
Fig 18. BEGIN ROUND reply

10.2 Close slot sequence

10.2.1 Structure

The form of the Close Slot Sequence has been defined in Figure 11.

10.2.2 I•CODE UID-OTP smart label reaction

Upon receiving a Close Slot Sequence, an I•CODE UID-OTP smart label that is in the SLOTTED READ state and has not replied will advance its count of slot number and will remain in the SLOTTED READ state.

Alternatively, upon receiving a Close Slot Sequence, an I•CODE UID-OTP smart label that is in the SLOTTED READ state and has just replied will move to the READY state and will be available for reading in another Reply Round.

There is no reply from the I•CODE UID-OTP smart label on the Close Slot Sequence.

10.3 Fix slot command

If there is no evidence of a collision, either through the CRC 16 check or through the mechanisms of reply signal examination described in Section 6.12 , it can be concluded that a successful collection of data has occurred, and a FIX SLOT command may be issued.

10.3.1 Structure

The FIX SLOT command is targeted at an I•CODE UID-OTP smart label, and normally that is an I•CODE UID-OTP smart label that has just replied. The FIX SLOT command consists of a Long start of frame, sixteen binary digits that are the CRC 16 of the I•CODE UID-OTP smart label being targeted, and a Common end of frame, as shown in Figure 19 below.

LSOF	CRC 16 (of I•CODE UID-OTP smart label response)	CEOF

Fig 19. FIX SLOT command
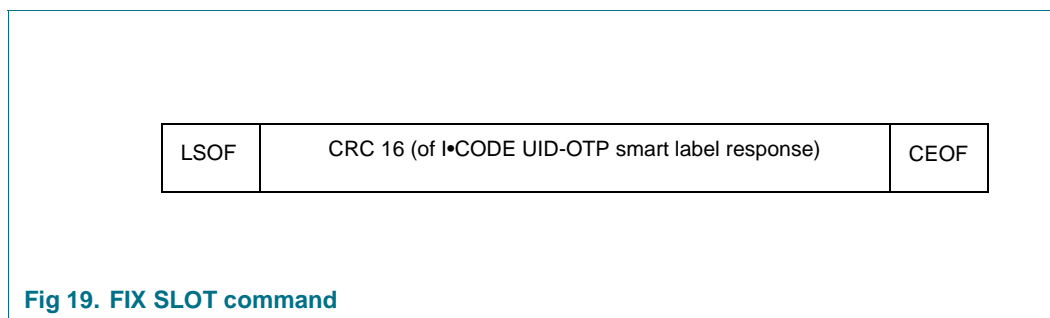
10.3.2 I•CODE UID-OTP smart label reaction

Only the I•CODE UID-OTP smart label that has just replied in the current slot, and has also a CRC 16 matching the data sent in the command will move to the FIXED SLOT state.

An I•CODE UID-OTP smart label that has just replied but has a CRC 16 not matching the data sent in the command will move to the READY state.

All other I•CODE UID-OTP smart labels will ignore this command except that such I•CODE UID-OTP smart labels in the SLOTTED READ state will increment their count of slot number.
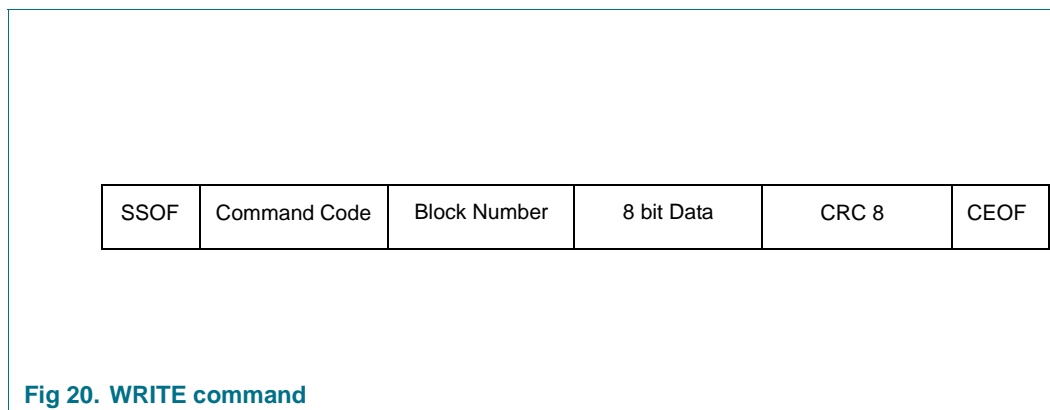
There is no reply from the I•CODE UID-OTP smart label on the FIX SLOT command.

10.4 Write command

The WRITE command writes data into the I•CODE UID-OTP Smart Label IC memory in 8-bit blocks.

10.4.1 Structure

The structure of the WRITE command is as shown in Figure 20 below.

SSOF	Command Code	Block Number	8 bit Data	CRC 8	CEOF

Fig 20. WRITE command

Writing may occur to the User Data, the UD CRC 16 and the Destroy Code.

The parameters of the Write command are defined as follows:

- The Command Code has 8 bits and a value of 01 hex.
- The Block Number parameter takes the values:
    - 00 hex (MSByte of UD), 01 hex, … 0B hex (LSByte of UD);
    - 0C hex (MSByte of UD CRC 16), 0D hex (LSByte of UD CRC 16);
    - 15 hex … 17 hex 24-bit Destroy Code starting with MSByte.

The 8-bit Data parameter is the eight bits of data to be written, MSB first.

The CRC 8 parameter is calculated over the Command Code, Block Number, and 8-bit Data as described in Section 6.10 .

As the memory is only one time programmable, writing of data needs to take place in a secure environment.

10.4.2 I•CODE UID-OTP smart label reaction

There is no reply from the I•CODE UID-OTP smart label on the WRITE command.

10.5 Destroy command

The DESTROY command will render the I•CODE UID-OTP smart label permanently unable to give any replies.

10.5.1 Structure

The structure of the DESTROY command is as shown in Figure 21 below.

SSOF	Command Code	IDD	Destroy Code	CRC 8	CEOF

Fig 21. DESTROY command

The parameters of the DESTROY command are defined as follows:

- The **Command Code** has 8 bits and a value of 02 hex.

- The IDD parameter includes the entire 152 bits of the IDD.

- The Destroy Code has 24 bits and can have any content previously written into the relevant section of memory.

- The CRC 8 parameter is calculated over the Command Code, the IDD parameter, and the Destroy Code as described in Section 6.10 .

10.5.2 I•CODE UID-OTP smart label reaction

There is no reply from the I•CODE UID-OTP smart label on the DESTROY command.

11. Data Integrity/calculation of CRC

The following explanations show the features of the I•CODE UID-OTP Smart Label IC protocol to protect read and write access to I•CODE UID-OTP smart labels from undetected errors.

11.1 Data transmission: RWD to I•CODE UID-OTP smart label

In the RWD to I•CODE UID-OTP smart label link an 8-bit CRC is used.

Table 5.   CRC 8 definition

CRC Definition		
Length	**Polynomial**	**Pre-set**
8 bits	$x^8 + x^4 + x^3 + x^2 + 1$	FD hex

If the I•CODE UID-OTP smart label detects a CRC 8 error, it will not execute the command.

For CRC calculation see Section 15

11.2 Data transmission: I•CODE UID-OTP smart label to RWD

The I•CODE UID-OTP smart label to RWD link uses a 16-bit CRC algorithm, which is defined in Table 4 and following text, and is stored in the I•CODE UID-OTP smart label.

Table 6. **CRC 16 definition**

CRC Definition				
CRC Type	**Length**	**Polynomial**	**Pre-set**	**Residue**
ISO/IEC 13239	16 bits	$x^{16} + x^{12} + x^5 + 1$	FFFF hex	1D0F hex

The Cyclic Redundancy Check 16 (CRC 16) is calculated on all 40 bits of the UID starting with the MSB of the MSByte thereof. The UD CRC 16 is calculated on all 96 bits of the UD starting with the MSB of the MSByte thereof.

A further transformation on the calculated CRC 16 is made. The value stored in the I•CODE UID-OTP smart label and attached to the message for transmission is the One's Complement of the CRC 16 calculated as defined in Table 6.

For ease of checking of received messages, the two CRC 16 bytes are often also included in the re-calculation without inverting them. In this case, the expected value for the residue of the CRC 16 generated in the receiver is 1D0F hex.

For CRC calculation see Section 15

12. Limiting values

See Delivery Type Addendum of Device

13. Characteristics

13.1 I•CODE UID-OTP inlet/smart label characterization and test

The parameters recommended to be characterized for the I•CODE UID-OTP Inlet/Smart Label are:

Table 7. **Parameters**

Symbol	Parameter	Conditions
f_{res}	Resonant frequency	Resonant frequency @ T_{amb} = 22 °C @ H_{min}
H_{min}	Threshold value of the field strength for BEGIN ROUND command	Response at BEGIN ROUND command

To detect damage of memory cells during production of inlets/labels a final test of the memory after assembly of the inlet/label is recommended (BEGIN ROUND). This is necessary to achieve lowest failure rates.

14. Package outline

15. Appendix

15.1 Example routines for CRC 8 and CRC 16 checksum calculation

The following lines show C-code examples for the calculation of the CRC 8 and the CRC 16 checksum.

15.1.1 CRC8_UID.C

```c
/* crc8_UID.c */
#include <ctype.h>
#include <stdlib.h>
#include <stdio.h>
#include <conio.h>


#define PRESET     0xFD
#define POLYNOMIAL 0x1D // x^8 + x^4 + x^3 + x^2 + 1 (MSB first)


// CRC8 (MSB first)


void crc8(unsigned char in, unsigned char *crc)
{
  int i;

  *crc = *crc ^ in;

  for (i = 0; i < 8; i++)
  {
    *crc = (*crc & 0x80)? (*crc << 1) ^ POLYNOMIAL : (*crc << 1);
  }
}


unsigned char a2x(unsigned char a[2])
{
  unsigned char value;

  if (isdigit(a[1]))  value =   a[1] - '0';
  else                value =   toupper(a[1]) - 'A' + 10;

  if (isdigit(a[0]))  value += (a[0] - '0') << 4;
```

```
    else                    value += (toupper(a[0]) - 'A' + 10) << 4;

    return (value);
}


unsigned int GetNextByte (FILE *fp, char *line,
                                unsigned char *byte)
{
    int i;

    fgets (line, 80, fp);
    if (feof(fp))  return (1);

    for (i = 0; i < 80 && !isspace(line[i]) && line[i] != '\0' ; i++)
    {
        if (!isxdigit(toupper(line[i])))  return -1;

        *byte = a2x((unsigned char *)line);
    }
    return 0;
}


main (int argc, char **argv)
{
    FILE        *fp;
    int         n, i;
    char        input_line[80], st;
    unsigned char byte[100], crc_reg;

    if (argc == 2)
    {
        fp = fopen(argv[1], "r");
        if (!fp)
        {
            perror(argv[1]);
            exit(0);
        }
    }
    else
    {
        printf("\n\nSyntax: CRC8_UID <file_with_input_data>\n\n");
        exit(0);
    }
```

```
    for (n = 0; n < 100; n++)

    {

      st = GetNextByte (fp, input_line, &(byte[n]));

      if (st == -1)

      {

        printf("\nWrong Number Format in %s", argv[1]);

        printf("\nHas to be a 2 number hex-value like A3 or 0C");

        printf("\n\n");

fclose(fp);

        return 0;

      }

      else if (st == 1) break;

      else            continue;

    }

    crc_reg = PRESET;

    printf("\n                Preset  ==>  ");

    printf("CRC8-Register: %02X hex\n", crc_reg);


  for (i = 0; i < n; i++)

    {

      printf("%2d. Input-Byte: %02X hex  ==>  ", i + 1, byte[i]);

      crc8(byte[i], &crc_reg);

      printf("CRC8-Register: %02X hex\n", crc_reg);

    }

    fclose(fp);

    printf("\nPress any key to continue ...\n");


    if (!getch()) getch();  return 0;

}
```

CRC8_UID Example Input File

```
30

00

00

25
```

CRC8_UID Example Output

```
                Preset  ==>  CRC8-Register: FD hex

1. Input-Byte: 30 hex  ==>  CRC8-Register: B4 hex
2. Input-Byte: 00 hex  ==>  CRC8-Register: 18 hex
3. Input-Byte: 00 hex  ==>  CRC8-Register: 25 hex
```

```
    4. Input-Byte: 25 hex  ==>  CRC8-Register: 00 hex
```

Press any key to continue ...

15.1.2 CRC16_UID.C

```c
/* crc16_UID.c */

#include <ctype.h>
#include <stdlib.h>
#include <stdio.h>
#include <conio.h>

#define PRESET      0xFFFF
#define POLYNOMIAL  0x1021  // x^16 + x^12 + x^5 + 1 (MSB first)

// CRC16 (MSB first)

void crc16(unsigned char in, unsigned int *crc)
{
  int i;



  *crc = *crc ^ ((unsigned int)in << 8);

  for (i = 0; i < 8; i++)
  {
    *crc = (*crc & 0x8000)? (*crc << 1) ^ POLYNOMIAL : (*crc << 1);
  }
}


unsigned char a2x(unsigned char a[2])
{
  unsigned char value;

if (isdigit(a[1]))  value =   a[1] - '0';
  else               value =   toupper(a[1]) - 'A' + 10;

  if (isdigit(a[0]))  value += (a[0] - '0') << 4;
  else               value += (toupper(a[0]) - 'A' + 10) << 4;

  return (value);
}


unsigned int GetNextByte (FILE *fp, char *line,
```

```
                          unsigned char *byte)
{
  int i;


  fgets (line, 80, fp);
  if (feof(fp))  return (1);


  for (i = 0; i < 80 && !isspace(line[i]) && line[i] != '\0' ; i++)
  {
    if (!isxdigit(toupper(line[i])))  return -1;


    *byte = a2x((unsigned char *)line);
  }
  return 0;
}


main (int argc, char **argv)
{
  FILE         *fp;
  int          n, i;
  char         input_line[80], st;
  unsigned char byte[100];
  unsigned int  crc_reg;


  if (argc == 2)
  {
    fp = fopen(argv[1], "r");
    if (!fp)
    {
      perror(argv[1]);
      exit(0);
    }
  }
  else
  {
    printf("\n\nSyntax: CRC16_UID <file_with_input_data>\n\n");
    exit(0);
  }


  for (n = 0; n < 100; n++)
  {
    st = GetNextByte (fp, input_line, &(byte[n]));
    if (st == -1)
```

```
        {

printf("\nWrong Number Format in %s", argv[1]);

        printf("\nHas to be a 2 number hex-value like A3 or 0C");

        printf("\n\n");

        fclose(fp);

        return 0;

      }

      else if (st == 1) break;

      else            continue;

    }

    crc_reg = PRESET;

    printf("\n            Preset  ==>  ");

    printf("CRC16-Register: %04X hex\n", crc_reg);


    for (i = 0; i < n; i++)

    {

      printf("%2d. Input-Byte: %02X hex  ==>  ", i + 1, byte[i]);

      crc16(byte[i], &crc_reg);

      printf("CRC16-Register: %04X hex\n", crc_reg & 0xFFFF);

    }

    fclose(fp);

    printf("\nPress any key to continue ...\n");


    if (!getch()) getch();  return 0;

}
```

CRC16_UID Example Input File


```
01

02

03
```

```
04

05

06

07

08

09

10

11

12

64

32
```

CRC16_UID Example Output

```
                Preset  ==>  CRC16-Register: FFFF hex

 1. Input-Byte: 01 hex  ==>  CRC16-Register: F1D1 hex
 2. Input-Byte: 02 hex  ==>  CRC16-Register: 0E7C hex
 3. Input-Byte: 03 hex  ==>  CRC16-Register: ADAD hex
 4. Input-Byte: 04 hex  ==>  CRC16-Register: 89C3 hex
 5. Input-Byte: 05 hex  ==>  CRC16-Register: 9304 hex
 6. Input-Byte: 06 hex  ==>  CRC16-Register: D71C hex
 7. Input-Byte: 07 hex  ==>  CRC16-Register: D77D hex
 8. Input-Byte: 08 hex  ==>  CRC16-Register: 4792 hex
 9. Input-Byte: 09 hex  ==>  CRC16-Register: 3B0A hex
10. Input-Byte: 10 hex  ==>  CRC16-Register: 9F09 hex
11. Input-Byte: 11 hex  ==>  CRC16-Register: 7946 hex
12. Input-Byte: 12 hex  ==>  CRC16-Register: 9BCD hex *
13. Input-Byte: 64 hex  ==>  CRC16-Register: D3F0 hex
14. Input-Byte: 32 hex  ==>  CRC16-Register: 1D0F hex

Press any key to continue ...
```

Note: * ...9BCD hex inverted = 6432 hex

16. Abbreviations

Table 8.   **Abbreviations**

Acronym	Description
ASK	Amplitude Shift Keying
CEOF	Common End of Frame
CRC	Cyclic Redundancy Check
CS	Close Slot
EOF	End of Frame
FS	FIX SLOT
IDD	Identifier Data
ISM	Industrial, Scientific, Medical
LSB	Least Significant Bit
LSByte	Least Significant Byte
LSOF	Long Start of Frame
m	Modulation Index
MSB	Most Significant Bit
MSByte	Most Significant Byte
OTP	One Time Programmable
REOF	Reply End of Frame
RO	Read Only
RSOF	Reply Start of Frame
RTF	Reader Talks First
R/W	Read/Write
RWD	Read/Write Device
SOF	Start of Frame
SSOF	Short Start of Frame
STAC	Slotted Terminated Adaptive Collection
UD	User Data
UD CRC	User Data Cyclic Redundancy Check
UID	Unique Identifier

127730

Product data sheet   **Rev. 03.00 — 6 December 2006**   **29 of 33**

17. Revision history

Table 9.   Revision history

Document ID	Release date	Data sheet status	Change notice	Supersedes
127730	Dezember 2006	Product data sheet	Initial version	3.0
Modifications:	• The format of this data sheet has been redesigned to comply with the new identity guidelines of NXP Semiconductors. • Legal texts have been adapted to the new company name where appropriate.			

127730

Product data sheet **Rev. 03.00 — 6 December 2006** **30 of 33**

18. Legal information

18.1 Data sheet status

Document status[1][2]	Product status[3]	Definition
Objective [short] data sheet	Development	This document contains data from the objective specification for product development.
Preliminary [short] data sheet	Qualification	This document contains data from the preliminary specification.
Product [short] data sheet	Production	This document contains the product specification.

[1]   Please consult the most recently issued document before initiating or completing a design.

[2]   The term 'short data sheet' is explained in section "Definitions".

[3]   The product status of device(s) described in this document may have changed since this document was published and may differ in case of multiple devices. The latest product status information is available on the Internet at URL http://www.nxp.com.

18.2 Definitions

Draft — The document is a draft version only. The content is still under internal review and subject to formal approval, which may result in modifications or additions. NXP Semiconductors does not give any representations or warranties as to the accuracy or completeness of information included herein and shall have no liability for the consequences of use of such information.

Short data sheet — A short data sheet is an extract from a full data sheet with the same product type number(s) and title. A short data sheet is intended for quick reference only and should not be relied upon to contain detailed and full information. For detailed and full information see the relevant full data sheet, which is available on request via the local NXP Semiconductors sales office. In case of any inconsistency or conflict with the short data sheet, the full data sheet shall prevail.

18.3 Disclaimers

General — Information in this document is believed to be accurate and reliable. However, NXP Semiconductors does not give any representations or warranties, expressed or implied, as to the accuracy or completeness of such information and shall have no liability for the consequences of use of such information.

Right to make changes — NXP Semiconductors reserves the right to make changes to information published in this document, including without limitation specifications and product descriptions, at any time and without notice. This document supersedes and replaces all information supplied prior to the publication hereof.

Suitability for use — NXP Semiconductors products are not designed, authorized or warranted to be suitable for use in medical, military, aircraft, space or life support equipment, nor in applications where failure or

malfunction of a NXP Semiconductors product can reasonably be expected to result in personal injury, death or severe property or environmental damage. NXP Semiconductors accepts no liability for inclusion and/or use of NXP Semiconductors products in such equipment or applications and therefore such inclusion and/or use is at the customer's own risk.

Applications — Applications that are described herein for any of these products are for illustrative purposes only. NXP Semiconductors makes no representation or warranty that such applications will be suitable for the specified use without further testing or modification.

Limiting values — Stress above one or more limiting values (as defined in the Absolute Maximum Ratings System of IEC 60134) may cause permanent damage to the device. Limiting values are stress ratings only and operation of the device at these or any other conditions above those given in the Characteristics sections of this document is not implied. Exposure to limiting values for extended periods may affect device reliability.

Terms and conditions of sale — NXP Semiconductors products are sold subject to the general terms and conditions of commercial sale, as published at http://www.nxp.com/profile/terms, including those pertaining to warranty, intellectual property rights infringement and limitation of liability, unless explicitly otherwise agreed to in writing by NXP Semiconductors. In case of any inconsistency or conflict between information in this document and such terms and conditions, the latter will prevail.

No offer to sell or license — Nothing in this document may be interpreted or construed as an offer to sell products that is open for acceptance or the grant, conveyance or implication of any license under any copyrights, patents or other industrial or intellectual property rights.

18.4 Trademarks

Notice: All referenced brands, product names, service names and trademarks are the property of their respective owners.

19. Contact information

For additional information, please visit: **http://www.nxp.com**

For sales office addresses, send an email to: **salesaddresses@nxp.com**

20. Tables

21. Figures

22. Contents

founded by

PHILIPS

Date of release: 6 December 2006
Document identifier: 127730