DATA SHEET

SL2 ICS10
I•CODE EPC
Smart Label IC
Functional Specification

Product Specification
Revision 3.0

2004 January 30
080530

Philips
Semiconductors

PHILIPS

CONTENTS

1 FEATURES 4
1.1 I•CODE EPC RF Interface Features 4
1.2 Memory Features 4
1.3 Security Features 4
1.4 Supported Standards 4
2 DESCRIPTION 5
2.1 Contactless Energy and Data Transfer 5
2.2 Anticollision 5
2.3 Customer Application Support and Training 5
3 BLOCKDIAGRAM 6
4 MEMORY ORGANIZATION 7
4.1 Configuration of delivered ICs 7
5 FUNCTIONAL DESCRIPTION 8
5.1 Basic System Configuration 8
5.2 Energy Transmission 9
5.3 Data Transmission: I•CODE EPC Smart Label to RWD 9
5.3.1 Coding 10
5.3.2 Baud Rate 10
5.3.3 Reply Framing 11
5.4 Data Transmission: RWD to I•CODE EPC Smart Label 12
5.4.1 Unitary Pulse 12
5.4.2 Baud Rate 12
5.4.3 Set of Symbols 13
5.4.4 Symbol Coding 14
5.5 Collision Detection 14
5.5.1 Concept 14
5.5.2 Illustration 14
6 PROTOCOL 15
6.1 General Description 15
6.2 I•CODE EPC Smart Label Selection 16
6.3 Truncated Reply 16
7 PROTOCOL TIMING 17
7.1 Reply Round Timing 17
7.2 Start-up Time 18
7.3 Reset Time 18
7.4 Waiting Time after Reply Round Sequence 18
7.5 Waiting Time after WRITE / DESTROY 18
8 STATE DIAGRAM 19
8.1 General Description of States 20

9 COMMAND SET 21
9.1 BEGIN ROUND Command 21
9.1.1 Structure 21
9.1.2 I•CODE EPC Smart Label Reaction 21
9.2 Close Slot Sequence 22
9.2.1 Structure 22
9.2.2 I•CODE EPC Smart Label Reaction 22
9.3 FIX SLOT Command 22
9.3.1 Structure 22
9.3.2 I•CODE EPC Smart Label Reaction 22
9.4 WRITE Command 23
9.4.1 Structure 23
9.4.2 I•CODE EPC Smart Label Reaction 23
9.5 DESTROY Command 24
9.5.1 Structure 24
9.5.2 I•CODE EPC Smart Label Reaction 24
10 DATA INTEGRITY/CALCULATION OF CRC 25
10.1 Data Transmission: RWD to I•CODE EPC Smart Label 25
10.2 Data Transmission: I•CODE EPC Smart Label to RWD 25
10.3 Example Routines for CRC 8 and CRC 16 Checksum Calculation 26
10.3.1 CRC8_EPC.C 26
10.3.2 CRC16_EPC.C 29
11 I•CODE EPC INLET/SMART LABEL CHARACTERISATION AND TEST 32
12 ABBREVIATIONS 33
13 DATA SHEET STATUS 34
14 DEFINITIONS 34
15 DISCLAIMERS 34
16 REVISION HISTORY 35

1 FEATURES

Integrated Circuit for Contactless Radio Frequency Identification Smart Label.

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

1.1 I•CODE EPC 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)
- 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 EPC smart labels per second)
- Label DESTROY command with 24-bit Destroy Code protection

1.2 Memory Features

- 136 bits, organised in 17 blocks of 1 byte each
- Data retention of 5 years

1.3 Security Features

- Lock mechanism for each user memory block (write protection)

1.4 Supported Standards

- EPCglobal Inc., 13.56 MHz ISM Band Class 1 Radio Frequency Identification Tag Interface Specification

2 DESCRIPTION

The I•CODE EPC 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 Philips Semiconductors there are various manufacturers available for both the smart labels as well as for the Read/Write devices (RWD).

2.1 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 EPC 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 EPC 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.

2.2 Anticollision

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

2.3 Customer Application Support and Training

Within the dedicated CAS team at the BL Identification.

Please Contact:

info.bli@philips.com

Accompanying Data Sheets and Application Notes:

http://www.semiconductors.com/markets/identification/datasheets/index.html#icode

3 BLOCK DIAGRAM

The I•CODE EPC 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 EPC smart label, and modulates the electromagnetic field for data transmission from the I•CODE EPC smart label to the RWD.

Data are stored in a memory with a capacity of 136 bits which is organised in 17 blocks consisting of 1 byte each (1 block = 8 bits).

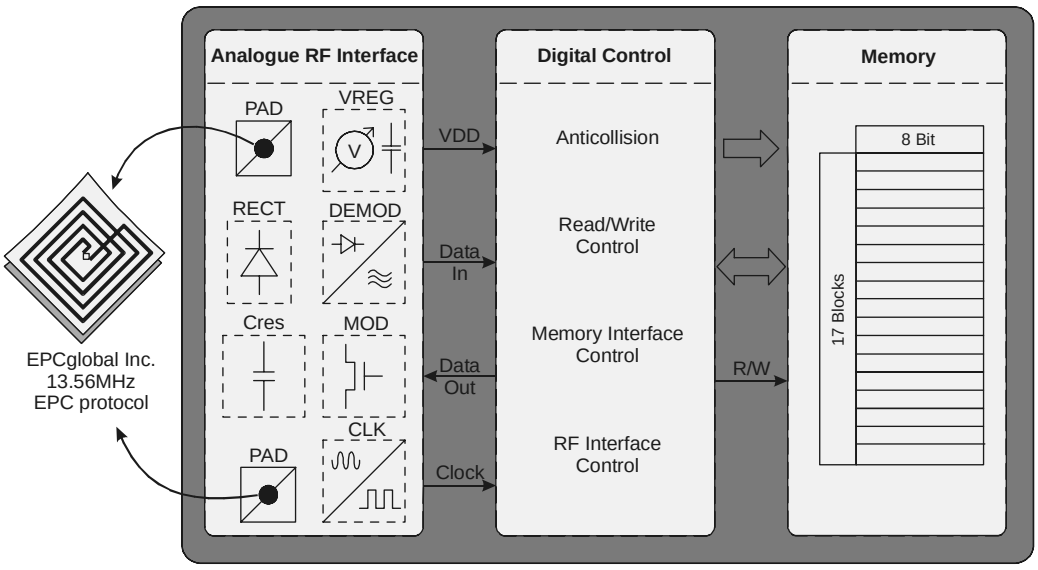


Fig.1 Block Diagram.

4 MEMORY ORGANIZATION

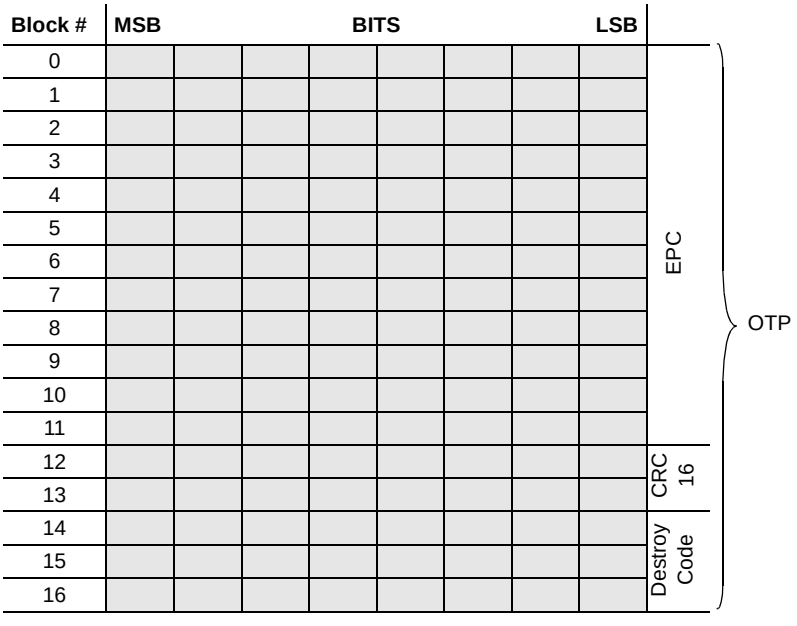


Fig.2 Memory Organisation.

The memory has a capacity of 136 bits and is organised in 17 blocks, consisting of 1 byte each. All MSBytes of the different fields (EPC, CRC 16, Destroy Code) are located at the **lowest** block addresses of the corresponding memory blocks.

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

- The electronic product code (EPC) can be of 96 bits in extent. The values of EPC are defined by the EPCglobal Inc. (please refer to the EPCglobal Inc. Technical Report"The Electronic Product Code – a naming scheme for physical objects").
- A 16-bit Cyclic Redundancy Check (CRC 16).
- A 24-bit Destroy Code.

4.1 Configuration of delivered ICs

The I•CODE EPC Smart Label ICs are delivered with following configuration by Philips Semiconductors:

- The content of all memory blocks is not defined and One Time Programmable (OTP). The CRC 16 is not valid.

5 FUNCTIONAL DESCRIPTION

5.1 Basic System Configuration

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

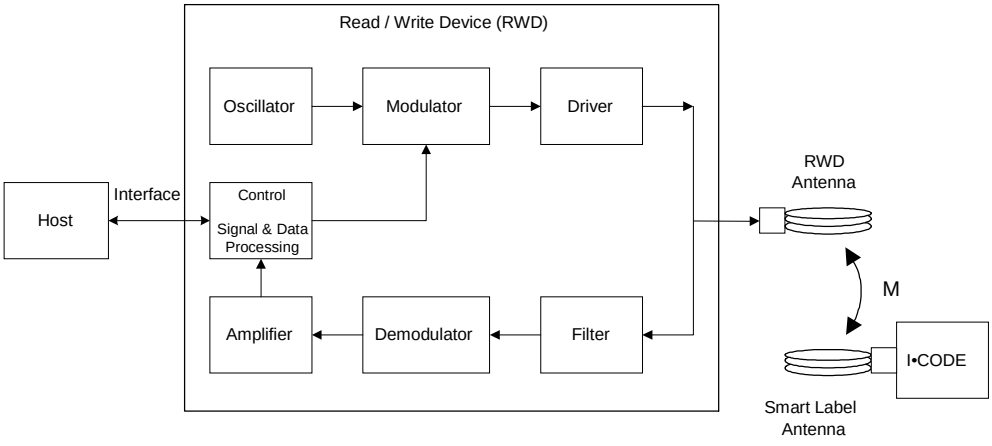


Fig.3 System Configuration.

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

5.2 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 EPC smart label antenna and induces a voltage there. The voltage drives a current and the I•CODE EPC 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 EPC Smart Label IC is designed for low power consumption.

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

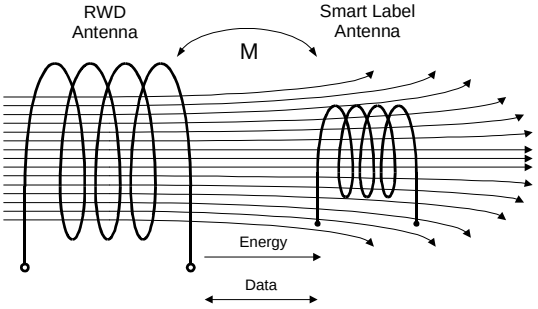


Fig.4 Energy Transmission.

5.3 Data Transmission: I•CODE EPC Smart Label to RWD

For data transmission from the I•CODE EPC smart label to the RWD, the implemented method is called 'load modulation'. Here the I•CODE EPC 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.

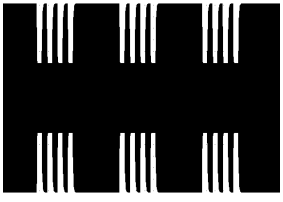


Fig.5 Smart label antenna voltage during load modulation.

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.

5.3.1 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 µs), as shown in following Figure 6.

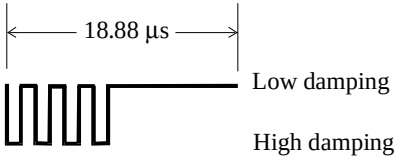


Fig.6 Modulation for logic 0.

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.

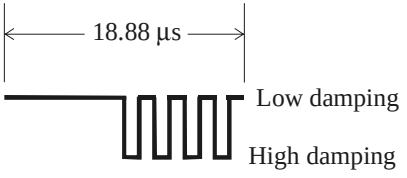


Fig.7 Modulation for logic 1.

5.3.2 BAUD RATE

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

5.3.3 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.

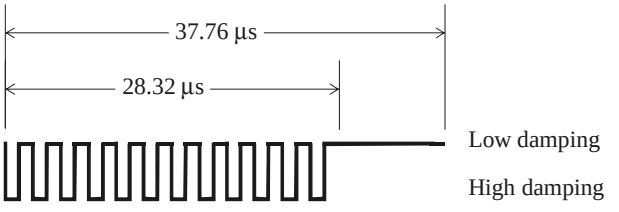


Fig.8 Reply Start of Frame (RSOF).

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).

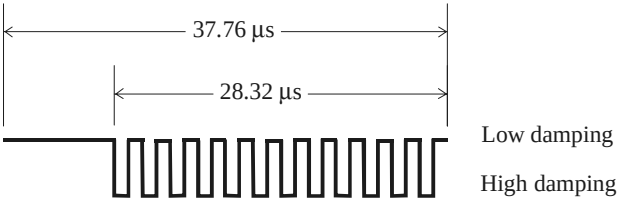


Fig.9 Reply End of Frame (REOF).

5.4 Data Transmission: RWD to I•CODE EPC Smart Label

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

5.4.1 Unitary Pulse

All of the signalling from the RWD to the I•CODE EPC 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)

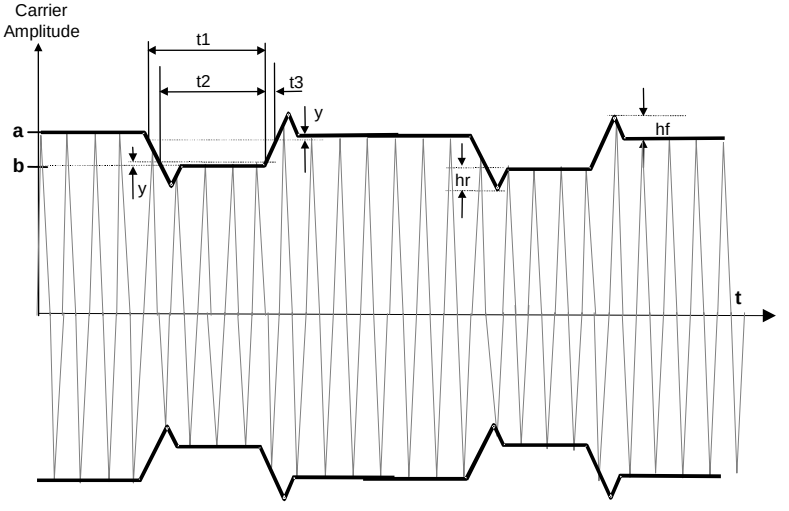


Fig.10 Unitary Pulse.

Table 1 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

5.4.2 Baud Rate

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

5.4.3 SET OF SYMBOLS

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

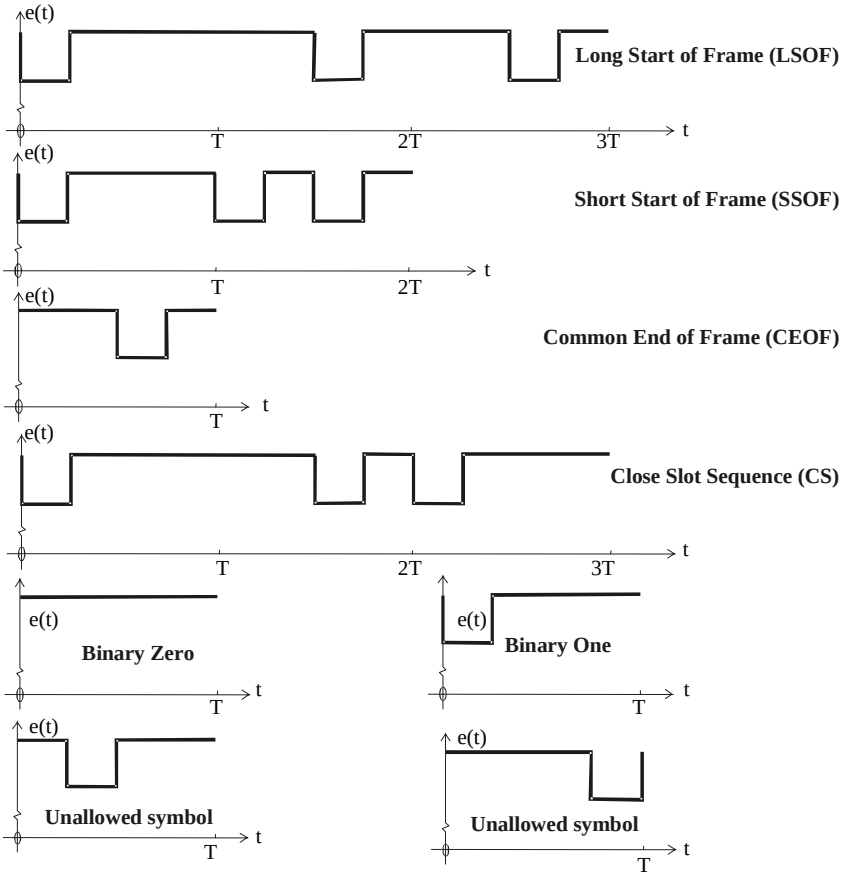


Fig.11 Set of Symbols.

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 Chapter 9.

5.4.4 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).

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

5.5 Collision Detection

5.5.1 CONCEPT

The used collision detection methodology depends upon the facts that:

- the data content of all replying I•CODE EPC smart labels will be different.
- the bit boundaries in the reply signals from all simultaneously replying I•CODE EPC 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 EPC smart labels will exhibit in the receiver of the RWD the interference pattern as illustrated in the Figure 12 below.

5.5.2 ILLUSTRATION

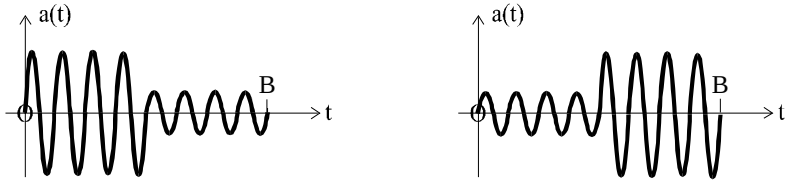


Fig.12 Illustrations of reply collision.

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).

6 PROTOCOL

6.1 General Description

The I•CODE EPC 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.

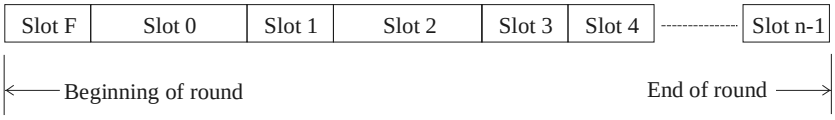


Fig.13 Illustration of slots in a Reply Round.

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 EPC smart labels, which were waiting in the READY state, to enter the SLOTTED READ state (please refer to Chapter 8). Those I•CODE EPC smart labels calculate for themselves a proposed reply slot. Each I•CODE EPC smart label is equipped with a slot counter. The slot counters of all I•CODE EPC 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 EPC 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 5.5, and makes the detection of collisions, when they occur, highly probable.

Another of these is the checking of the 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 10.2.

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 9.3. 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 7.1.

6.2 I•CODE EPC Smart Label Selection

The I•CODE EPC Smart Label IC supports a selection feature in which groups of I•CODE EPC smart labels may be selected e.g. by header, domain manager, object class, or serial number. 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 EPC smart labels in the READY state, the Selection Mask is checked against the EPC data stored in the I•CODE EPC smart labels on a bit by bit basis, Most Significant Bit (MSB) first. In case of a match the related I•CODE EPC smart label is selected and moves from the READY state to the SLOTTED READ state.

I•CODE EPC 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 EPC smart labels, as a result of the BEGIN ROUND command not having a Selection Mask parameter, all I•CODE EPC smart labels in a READY state are automatically selected and move into the SLOTTED READ state.

6.3 Truncated Reply

As shown in Figure 14 a selected I•CODE EPC 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 EPC (memory blocks 0 to 11).

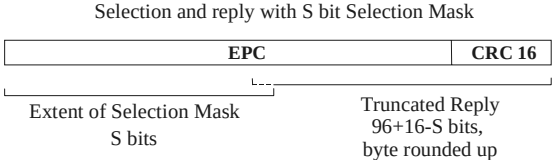


Fig.14 Illustration of selection and reply.

The Truncated Reply starts with the MSB at the preceding byte border of the truncated EPC data and stops with the LSB of the CRC 16.

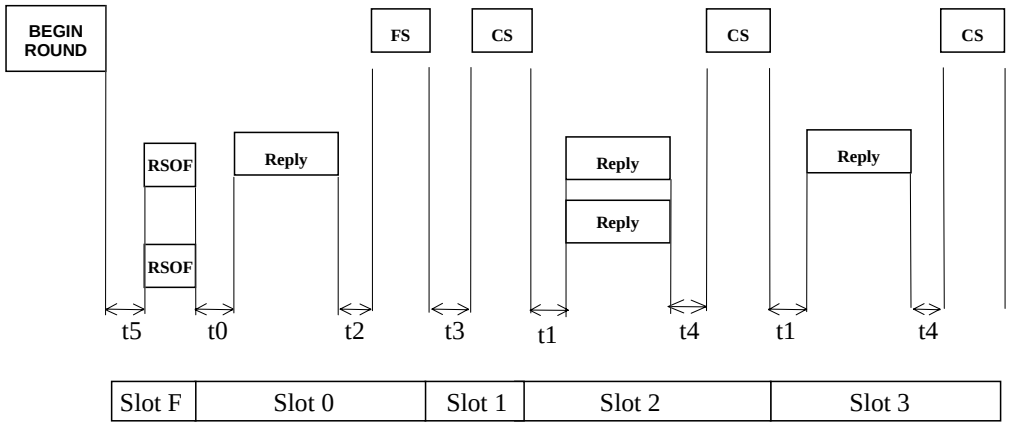
In case S = 96 bits only the CRC 16 is replied.

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.

7 PROTOCOL TIMING

7.1 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

Fig.15 Reply Round Sequence.

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

t1: Time between the logical end of any slot except Slot F and the reply of I•CODE EPC 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 EPC 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 EPC 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 EPC 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 EPC smart label was correctly received.

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

Table 2 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

Note

1. 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.

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 $32/f_c$

7.2 Start-up Time

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

7.3 Reset Time

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

7.4 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.

7.5 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 ms before sending a subsequent command or switching off the RF field.

8 STATE DIAGRAM

The following figure shows the State Diagram of the I•CODE EPC Smart Label IC.

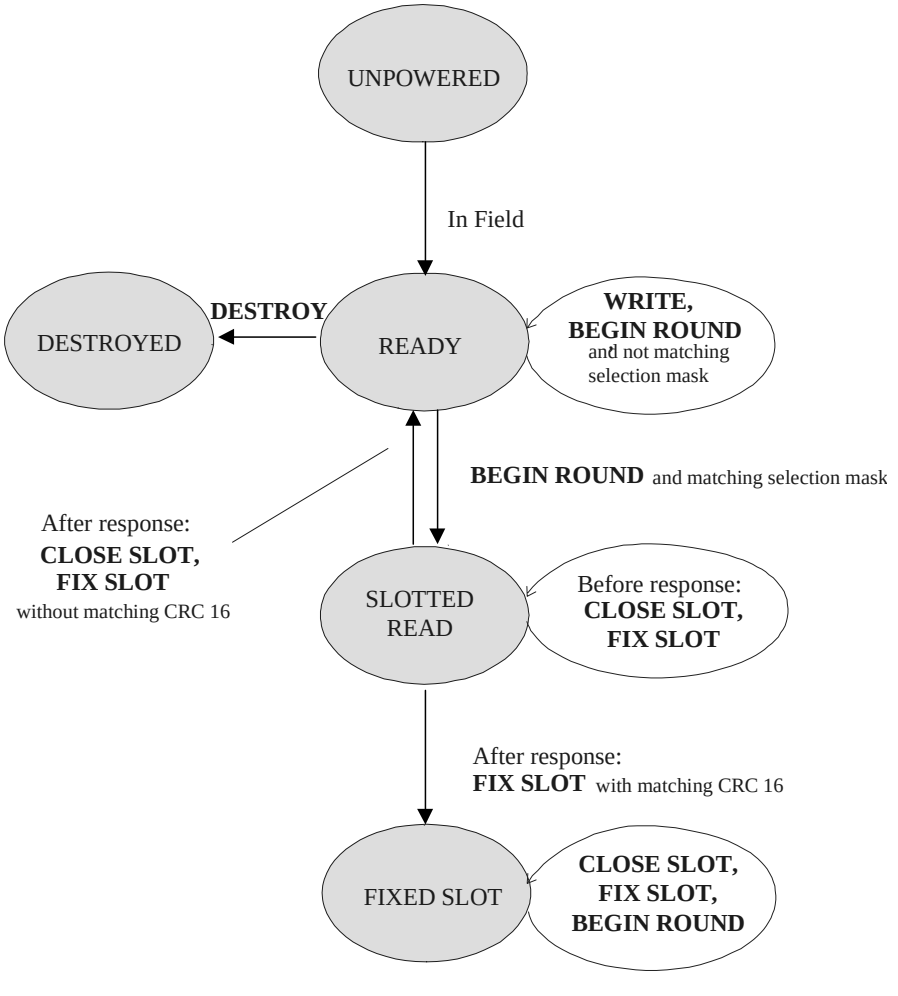


Fig.16 State Diagram.

8.1 General Description of States

UNPOWERED

The powering magnetic field of the RWD is switched off or the I•CODE EPC 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.

READY

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

SLOTTED READ

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

FIXED SLOT

The I•CODE EPC 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.

DESTROYED

The I•CODE EPC 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.

9 COMMAND SET

All bytes are transmitted MSB first.

9.1 BEGIN ROUND Command

9.1.1 STRUCTURE

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

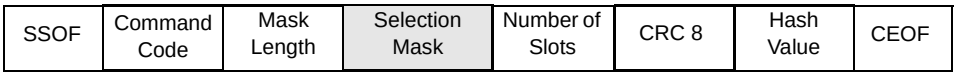


Fig.17 BEGIN ROUND command.

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

- The **Command Code** has length 8 bits and value 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.
- 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 EPC smart label EPC data in the selection process. If all bits of the Selection Mask match, I•CODE EPC 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 10.1.

The **Hash Value** has 8 bits and is used by the I•CODE EPC smart label to generate from the EPC data the random slot positions.

9.1.2 I•CODE EPC SMART LABEL REACTION

All I•CODE EPC 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 EPC smart label reaches that count, the I•CODE EPC smart label will reply as described in Section 6.3.

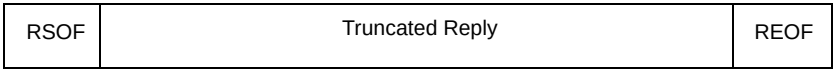


Fig.18 BEGIN ROUND reply.

9.2 Close Slot Sequence

9.2.1 STRUCTURE

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

9.2.2 I•CODE EPC SMART LABEL REACTION

Upon receiving a Close Slot Sequence, an I•CODE EPC 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 EPC 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 EPC smart label on the Close Slot Sequence.

9.3 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 5.5, it can be concluded that a successful collection of data has occurred, and a FIX SLOT command may be issued.

9.3.1 STRUCTURE

The FIX SLOT command is targeted at an I•CODE EPC smart label, and normally that is an I•CODE EPC 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 EPC smart label being targeted, and a Common end of frame, as shown in Figure 19 below.

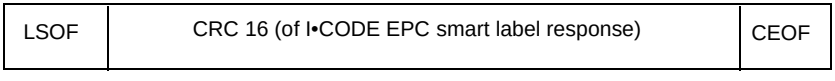


Fig.19 FIX SLOT command.

9.3.2 I•CODE EPC SMART LABEL REACTION

Only the I•CODE EPC 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 EPC 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 EPC smart labels will ignore this command except that such I•CODE EPC smart labels in the SLOTTED READ state will increment their count of slot number.

There is no reply from the I•CODE EPC smart label on the FIX SLOT command.

9.4 WRITE Command

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

9.4.1 STRUCTURE

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

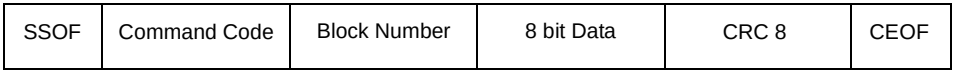


Fig.20 WRITE command.

Writing may occur to the EPC, the CRC 16 and the Destroy Code.

The parameters of the Write command are defined as follows:

- The **Command Code** has 8 bits and the value of 01 hex.
- The **Block Number** parameter takes the values:
  - 00 hex (MSByte of EPC), 01 hex, … 0B hex (LSByte of EPC);
  - 0C hex (MSByte of CRC 16), 0D hex (LSByte of CRC 16);
  - 0E hex … 10 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 10.1.

As the EPC, the CRC 16 and the Destroy Code are only one time programmable, writing needs to take place in a secure environment.

9.4.2 I•CODE EPC SMART LABEL REACTION

There is no reply from the I•CODE EPC smart label on the WRITE command.

9.5 DESTROY Command

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

9.5.1 STRUCTURE

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

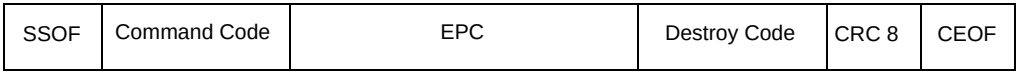


Fig.21 DESTROY command.

The parameters of the DESTROY command are defined as follows:

- The **Command Code** has 8 bits and the value of 02 hex.
- The **EPC** parameter includes the entire 96 bits of the EPC.
- 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 EPC and the Destroy Code as described in Section 10.1.

9.5.2 I•CODE EPC SMART LABEL REACTION

There is no reply from the I•CODE EPC smart label on the DESTROY command.

10 DATA INTEGRITY/CALCULATION OF CRC

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

10.1 Data Transmission: RWD to I•CODE EPC Smart Label

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

Table 3 CRC 8 definition

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

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

10.2 Data Transmission: I•CODE EPC Smart Label to RWD

The I•CODE EPC 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 EPC smart label.

Table 4 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 96 bits of the EPC 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 EPC smart label and attached to the message for transmission is the One's Complement of the CRC 16 calculated as defined in Table 4.

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.

10.3 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.

10.3.1 CRC8_EPC.C

```
/* crc8_EPC.c */

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

#define PRESET     0xFF
#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_EPC <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_EPC Example Input File

```
02
00
01
02
03
04
05
06
07
08
FF
FE
FD
00
00
00
51
```

CRC8_EPC Example Output

```
                Preset  ==>  CRC8-Register: FF hex
 1. Input-Byte: 02 hex  ==>  CRC8-Register: FE hex
 2. Input-Byte: 00 hex  ==>  CRC8-Register: D9 hex
 3. Input-Byte: 01 hex  ==>  CRC8-Register: 10 hex
 4. Input-Byte: 02 hex  ==>  CRC8-Register: F7 hex
 5. Input-Byte: 03 hex  ==>  CRC8-Register: 0B hex
 6. Input-Byte: 04 hex  ==>  CRC8-Register: BB hex
 7. Input-Byte: 05 hex  ==>  CRC8-Register: CA hex
 8. Input-Byte: 06 hex  ==>  CRC8-Register: A9 hex
 9. Input-Byte: 07 hex  ==>  CRC8-Register: 07 hex
10. Input-Byte: 08 hex  ==>  CRC8-Register: BB hex
11. Input-Byte: FF hex  ==>  CRC8-Register: 67 hex
12. Input-Byte: FE hex  ==>  CRC8-Register: 1E hex
13. Input-Byte: FD hex  ==>  CRC8-Register: 95 hex
14. Input-Byte: 00 hex  ==>  CRC8-Register: 82 hex
15. Input-Byte: 00 hex  ==>  CRC8-Register: 1C hex
16. Input-Byte: 00 hex  ==>  CRC8-Register: 51 hex
17. Input-Byte: 51 hex  ==>  CRC8-Register: 00 hex

Press any key to continue ...
```

10.3.2 CRC16_EPC.C

```
/* crc16_EPC.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_EPC <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_EPC Example Input File

```
01
02
03
04
05
06
07
08
```

```
09
10
11
12
64
32
```

CRC16_EPC 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

11 I•CODE EPC INLET/SMART LABEL CHARACTERISATION AND TEST

The parameters recommended to be characterised for the I•CODE EPC Inlet/Smart Label are:

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

12 ABBREVIATIONS

Abbreviation	Definition
AIDC	Auto-ID Center
ASK	Amplitude Shift Keying
CEOF	Common End of Frame
CRC	Cyclic Redundancy Check
CS	Close Slot
EOF	End of Frame
EPC	Electronic Product Code
FS	Fix Slot
ISM	Industrial, Scientific, Medical
LSB	Least Significant Bit
LSByte	Least Significant Byte
LSOF	Long Start of Frame
m	Modulation Index
MIT	Massachusetts Institute of Technology
MSB	Most Significant Bit
MSByte	Most Significant Byte
OTP	One Time Programmable
REOF	Reply End of Frame
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

13 DATA SHEET STATUS

LEVEL	DATA SHEET STATUS[1]	PRODUCT STATUS[2][3]	DEFINITION
I	Objective data	Development	This data sheet contains data from the objective specification for product development. Philips Semiconductors reserves the right to change the specification in any manner without notice.
II	Preliminary data	Qualification	This data sheet contains data from the preliminary specification. Supplementary data will be published at a later date. Philips Semiconductors reserves the right to change the specification without notice, in order to improve the design and supply the best possible product.
III	Product data	Production	This data sheet contains data from the product specification. Philips Semiconductors reserves the right to make changes at any time in order to improve the design, manufacturing and supply. Relevant changes will be communicated via a Customer Product/Process Change Notification (CPCN).

Notes

1. Please consult the most recently issued data sheet before initiating or completing a design.
2. The product status of the device(s) described in this data sheet may have changed since this data sheet was published. The latest information is available on the Internet at URL http://www.semiconductors.philips.com.
3. For data sheets describing multiple type numbers, the highest-level product status determines the data sheet status.

14 DEFINITIONS

Short-form specification — The data in a short-form specification is extracted from a full data sheet with the same type number and title. For detailed information see the relevant data sheet or data handbook.

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

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

15 DISCLAIMERS

Life support applications — These products are not designed for use in life support appliances, devices, or systems where malfunction of these products can reasonably be expected to result in personal injury. Philips Semiconductors' customers using or selling these products for use in such applications do so at their own risk and agree to fully indemnify Philips Semiconductors for any damages resulting from such application.

Right to make changes — Philips Semiconductors reserves the right to make changes in the products - including circuits, standard cells, and/or software - described or contained herein in order to improve design and/or performance. When the product is in full production (status 'Production'), relevant changes will be communicated via a Customer Product/Process Change Notification (CPCN). Philips Semiconductors assumes no responsibility or liability for the use of any of these products, conveys no licence or title under any patent, copyright, or mask work right to these products, and makes no representations or warranties that these products are free from patent, copyright, or mask work right infringement, unless otherwise specified.

16 REVISION HISTORY

Table 5 SL2 ICS10 Revision History

REVISION	DATE	CPCN	PAGE	DESCRIPTION
3.0	**2003 Dec**	-		**Status now -> Product Specification.**
1.1	**2003 Dec**	-		**Reworked version.**
1.0	**2003 Feb**	-		**Initial version.**

Philips Semiconductors – a worldwide company

Contact information

For additional information please visit **http://www.semiconductors.philips.com**. Fax: **+31 40 27 24825**
For sales offices addresses send e-mail to: **sales.addresses@www.semiconductors.philips.com**.

SCA74

Let's make things better.

Philips Semiconductors

PHILIPS