



HT RMxx1 Family

HT CM401

Interface Protocol Reader ↔ Host

Table of Contents

1	Introduction	5
1.1	Abbreviations.....	6
1.2	Definitions	7
2	Provided Host Software Modules	8
2.1	Introduction.....	8
2.2	Using the Provided Host Software Modules	9
3	Communication Reader-Host.....	10
3.1	Introduction.....	10
3.2	Ordinary Protocol	11
3.3	Extended Protocol	11
3.4	Transfer Timeout Intervals	12
3.5	Command Set	13
3.6	Status Byte	15
3.7	Command Description for Operating Mode	16
3.7.1	HT1_Get_Snr.....	16
3.7.2	HT1_Get_Snr_Adv	17
3.7.3	HT1_GetSnrSelectHalt.....	18
3.7.4	HT1_Select_Snr	19
3.7.5	HT1_SelectSnrDirect	20
3.7.6	HT1_Select_Last.....	21
3.7.7	HT1_Halt_Selected	22
3.7.8	HT1_Read_Page	23
3.7.9	HT1_Read_Block.....	24
3.7.10	HT1_Write_Page.....	25
3.7.11	HT1_Write_Block.....	26
3.7.12	HT1_Tag_Authent	27
3.7.13	HT1_Mutual_Authent	28
3.7.14	HTS_GetSnrReset.....	29
3.7.15	HTS_SelectSnrReset	30
3.7.16	HTS_TagAuthent	31
3.7.17	HTS_TagAuthent_PW	31
3.7.18	HT2_Get_Snr.....	32
3.7.19	HT2_Get_Snr_Reset	33
3.7.20	HT2_Halt_Selected	34
3.7.21	HT2_Read_Page	35
3.7.22	HT2_Read_Page_Inv	36
3.7.23	HT2_Write_Page.....	37
3.7.24	HT2_Read_PublicB.....	38
3.7.25	HT2_Read_Miro	39
3.7.26	HT2_Poll_Tags	40
3.7.27	RWD_Poll_Kb_Tags.....	42

3.7.28	RWD_Get_Version	44
3.7.29	RWD_Reset_Sys.....	45
3.7.30	RWD_HF_Reset	46
3.7.31	RWD_Stop_Cmd	47
3.7.32	RWD_SetBaudrate.....	47
3.7.33	RWD_Read_Input.....	48
3.7.34	RWD_Set_Output	49
3.7.35	RWD_Config_Ports	50
3.7.36	RWD_Read_Ports.....	51
3.7.37	RWD_Write_Ports	52
3.7.38	RWD_EE_Read	53
3.7.39	RWD_EE_Write	53
3.7.40	RWD_Set_Prox_Trm_Time	54
3.7.41	RWD_SetModuleAdr.....	55
3.7.42	RWD_Set_HF_Mode.....	56
3.7.43	RWD_FFT_Command.....	57
3.7.44	RWD_Read_BCD	58
3.7.45	RWD_Set_BCD	59
3.7.46	RWD_Set_BCD_Offset.....	60
3.7.47	RWD_Read_LR_Status.....	60
3.7.48	RWD_Set_Power_Down.....	61
3.7.49	Vegas_Read_All_Page	62
3.7.50	Vegas_Get_Dsp_Version	63
3.7.51	RWD_Key_Init_Mode.....	64
3.8	Command Description for KeyInit Mode.....	65
3.8.1	KI_Reset	65
3.8.2	KI_Write_SerNum	66
3.8.3	KI_Read_EE_Data.....	67
3.8.4	KI_Write_EE_Data	68
3.8.5	KI_Read_Control	69
3.8.6	KI_Write_Control	70
3.8.7	KI_Read_Secret2	71
3.8.8	KI_Write_Secret2	72
3.8.9	KI_Read_Control2	73
3.8.10	KI_Write_Control2	74
3.9	Examples to Access HITAG 1/S Transponders.....	75
3.9.1	Long Range: Anticollision Cycle.....	75
3.9.2	Proximity/Long Range: READ PLAIN	76
3.9.3	Proximity/Long Range: WRITE PLAIN	76
3.9.4	Proximity/Long Range: READ CRYPTO	77
3.9.5	Proximity/Long Range: WRITE CRYPTO	77
3.10	Examples to Access HITAG 2 Transponders.....	78
3.10.1	Proximity/Long Range: READ	78
3.10.2	Proximity/Long Range: WRITE.....	78
3.11	Examples to Access HITAG S Transponders.....	79
3.11.1	Hitag S State Diagram.....	79
3.11.2	Long Range: Anticollision Cycle.....	80
3.11.3	Long Range: Anticollision Cycle.....	81
3.11.4	Proximity/Long Range: READ PLAIN	82
3.11.5	Proximity/Long Range: WRITE PLAIN	82

3.11.6	Proximity/Long Range: READ CRYPTO	83
3.11.7	Proximity/Long Range: WRITE CRYPTO	83
4	Transponders	84
4.1	HITAG 1 Transponders	84
4.1.1	Memory Organization.....	84
4.1.2	Anticollision	85
4.1.3	Operation-Modes and Configuration.....	86
4.1.4	Configuration of Delivered HITAG 1 Transponders.....	89
4.2	HITAG 2 Transponders	90
4.2.1	Memory Organization.....	90
4.2.2	Operation-Modes and Configuration.....	91
4.2.3	Configuration of Delivered HITAG 2 Transponders.....	93
4.3	PIT (PCF793x) Transponders	94
4.3.1	Memory Organization.....	94
5	Personalization.....	95
5.1	Introduction	95
5.2	Personalization Concept	95
5.3	Personalization of HITAG 1 Transponders	100
5.3.1	Definition of Keys and Logdata	100
5.3.2	Changing Keys and Logdata	101
5.4	Personalization of HITAG 2 Transponders	103
5.4.1	Definition of Passwords and Keys.....	103
5.4.2	Changing Passwords and Keys.....	104
6	Security Considerations	105
6.1	Data Reliability.....	105
6.1.1	Data Stream between Read/Write Device and Transponder.....	105
6.1.2	Checking User Data.....	105
6.2	Data Privacy	106

1 Introduction

This description refers to the interface between a host (e.g. PC) and a contactless 125 kHz read/write device based on the HITAG Communication Controller, as there is e.g. the HT RM 1xx/2xx (Hitag Proximity Reader Module based on HTRC110), the HT CM400/1 (HITAG Core Module), HT RM440/401 family (HITAG Proximity Reader Module) and HT RM80x/90x family (HITAG Long Range Reader Module).

For easy and quick development of application specific host software Frosch Electronics provides a Library, Dll- and Header-Files.

Following transponders of the 125 kHz family are supported:

- HITAG 1
- HITAG 2
- HITAG S

Additional Features:

- High security by using cryptography, mutual authentication and password verification
- Addressing multiple (up to 255) read/write devices on a RS485-Bus
- Programmable port pins: 4 outputs; 2 inputs;
optional (requiring a special hardware because signals are not available on pin connectors of Philips Core Module): 8 pins either in-/output configurable or for connection to a keyboard-matrix up to 12 keys
- 85 bytes of user-defined data can be stored in an EEPROM of the read/write device

1.1 Abbreviations

Please find in the following a list of the abbreviations used in this document.

addr	Address
BCC	Block Check Character
BYTE_T	Byte (unsigned character)
char	Character
CRC	Cyclic Redundancy Check
DSP	Digital Signal Processor
DWORD_T	Double Word (unsigned)
FFT	Fast Fourier Transformation
HF	High Frequency
LSB	Least Significant Byte
MSB	Most Significant Byte
nmb	Number
OTP	One Time Programmable
pagenr	Page Number
RF	Radio Frequency
ro	Read Only
r/w	Read/Write
RWD	Read/Write Device
snr	Serial Number
TAG (tag)	Transponder
wo	Write Only

1.2 Definitions

Data sheet status	
Objective specification	This data sheet contains target or goal specifications for product development.
Preliminary specification	This data sheet contains preliminary data; supplementary data may be published later.
Product specification	This data sheet contains final product specifications.
Limiting values	
Limiting values given are in accordance with the Absolute Maximum Rating System (IEC 134). 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 section of the specification is not implied. Exposure to limiting values for extended periods may affect device reliability.	
Application information	
Where application information is given, it is advisory and does not form part of the specification.	

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 customers using or selling these products for use in such applications do so on their own risk and agree to fully indemnify Philips for any damages resulting from such improper use or sale.

2 Provided Host Software Modules

2.1 Introduction

On the Floppy Disk/CD-Rom added to this description you will find the following tools:

Library Files:

Hitag1.LIB	Hitag1 Functions
Hitag2.LIB	Hita2 Functions
HitagS.LIB	HitagS Functions
HitagKeyInit.LIB	HitagSecretData Functions
HitagRWD.LIB	Hitag Reader Functions
HitagVegas.LIB	Hitag Vegas Functions
HitagSerialPort.LIB	Serial Functions

Header Files:

Hitag1.H	HITAG 1/S function declarations
Hitag2.H	HITAG 2 function declarations
HitagS.H	HITAG S function declarations
HitagKeyInit.H	RWD Secret Data function declarations
HitagRWD.H	RWD general function declarations
HitagVegas.H	Vegas function declarations
HitagSerialPort.H	Serial Functions

Each Header File provides function declarations with detailed information about the use of commands.

Depending on the used command function (e.g. proloc_GetSnr) you have to include the corresponding Header File(s) in your application specific source file.

DLL Files:

Hitag1.DLL	HITAG 1/S Libriary
Hitag2.DLL	HITAG 2 Libriary
HitagS.DLL	HITAG S Libriary
HitagKeyInit.DLL	RWD Libriary
HitagRWD.DLL	RWD Libriary
HitagVegas.DLL	Vegas Libriary
HitagSerialPort.DLL	Serial FunctionsLibrary

We recommend to use our Source Files and Header Files and make your specific adaptations.

2.2 Using the Provided Host Software Modules

Communication via the serial interface between the host and the read/write device is handled by the serial Port DLL (SerialPort.dll).

- To configure the serial Port (e.g. Com2 at 57600 kBaud) on the Host System use the SerialPortProperties Structure before calling the SP_Init Command
strcpy(SerialPortProperties.sPort, "Com2"); //Select Port Com2
SerialPortProperties.BaudRate = 57600; //Set BaudRate to 57600
Default: Com1, 9600
- To open the serial port on the host system use function (Header File SerialPort.H)
SP_Init()
- To close the serial port on the host system use function (Header File PROLIB6.H)
SP_Exit()
- The change of the BCC calculation (when entering or leaving KeyInitMode) is done automatically

The names of the commands described in the following have to be prefixed with e.g. „Hitag1_“ to get the corresponding names in the C-Library (e.g. function *Hitag1_GetSnr()* for command *GetSnr*) for Ordinary Protocol.

All Header Files contain short examples to illustrate the usage of each command.

3 Communication Reader-Host

3.1 Introduction

The host (e.g. PC) communicates with the contactless 125 kHz read/write device via a serial interface using a baud rate of 9600 baud after PowerOn of the reader, this value can be changed by serial command, the new value is not stored in EEPROM.

Data transfer details are: 1 start bit, 8 data bits, 1 stop bit and no parity bit, the Least Significant Bit is sent first.

Each communication sequence consists of a block of bytes sent by the host, and a block of bytes answered by the reader.

All bytes are transmitted transparently, i.e. you can use any character between 0x00 and 0xFF.

Block Length:

Block Length is the sum of all transferred bytes including Block Length but excluding BCC.

Block Title:

The Command Byte if sent from host to reader.

The Status Byte if sent from reader to host.

Data:

Data bytes are only transmitted if data is transferred.

BCC:

The BCC (Block Check Character) is calculated by bytes 1 to n-1 (n=number of bytes of the whole communication sequence).

A different BCC calculation in Operating Mode (mode of the reader for using standard commands) and in KeyInit Mode (mode of the reader device for using personalization commands) helps to avoid the overwriting of secret data accidentally.

BCC calculation in Operating Mode of the reader:

The BCC is computed by EXOR-operation of all block data bytes including Block Length.

EXOR for 1 Bit:

A	B	EXOR
0	0	0
0	1	1
1	0	1
1	1	0

Example for command *GetSnr*:

Byte 1: Block Length	0000 0010	0x02
Byte 2: Command Byte	0100 0111	0x47
Byte 3: BCC	0100 0101	0x45

BCC calculation in KeyInit Mode of the reader:

The BCC is computed by adding all block data bytes including Block Length. The least significant eight bits are used as BCC.

3.2 Ordinary Protocol

If only a single read/write device with a node address equal to zero is connected to the host (e.g. on a RS232 serial line) the Ordinary Protocol is used to address this reader.

Format of the Ordinary Protocol (*HOST*→*READER* and *READER*→*HOST*):

Byte	1	2	3	4	n
Function	Block Length	Block Title	data	data	BCC

3.3 Extended Protocol

If more than one read/write devices with node addresses different from zero are connected to the host (e.g. on a RS485 serial line) the Extended Protocol is used to address a single reader.

Format of the Extended Protocol (*HOST*→*READER* and *READER*→*HOST*):

Byte	1	2	3	4	n-1	n
Function	Block Length + 0x80	Block Title	data	data	Node Address	BCC

Differences to Ordinary Protocol: Bit 7 of Block Length is set, and the Node Address is inserted just before BCC.

If a reader's node address is different from zero, the reader enters net-mode. In this mode the reader expects all commands from the host to be sent in Extended Protocol including the right Node Address (except *SetModuleAdr*). If the host transmits a string that does not meet these conditions, the command is ignored, and there will be no answer from the reader (whereas a reader being not in net-mode - with node address equal to zero - would at least answer with a SERIAL ERROR message).

The command *SetModuleAdr* is used to assign a unique node address to a device whose serial number is known. This command should be sent in Ordinary Protocol. If the right serial number was sent, there will be an answer from the read/write device. This answer is sent in Ordinary Protocol if the former node address of the reader was zero, otherwise the answer is sent in Extended Protocol.

For communication in Extended Protocol use commands with 'Proloc_M'-prefix. For further information see Header File PROLBMU6.h.

3.4 Transfer Timeout Intervals

Character Delay:

Character Delay is the maximum time permitted to elapse between sending two consecutive characters of a block.

$$\text{Character Delay} \leq 150 \text{ ms}$$

Block Delay:

Block Delay is only necessary if an error has occurred in the serial communication. To allow for re-synchronization in that case of malfunction there must be a minimum interval - defined as Block Delay - until sending the next block.

$$\text{Block Delay} \geq 160 \text{ ms}$$

3.5 Command Set

The Command Byte is part of the block sent from the host.

Command Bytes used in Abic or HTRC110 based (A), a Proximity (P) and/or Long Range (L) Reader:

Operating Mode:

COMMAND BYTE		COMMAND NAME	READER	TRANSPONDERS			
				HITAG 1/S	HITAG 2	MIRO	HitagS
'A'	0x41	HT1 Mutual Authent		P/L			
'B'	0x42	HT1 Read Block		P/L/A			P/L/A
'D'	0x44	RWD Set Power Down	L/A				
'E'	0x45	RWD EE Read	P/L				
'F'	0x46	RWD FFT Command	L				
'F'	0x46	RWD Set BCD	L				
'G'	0x47	HT1 Get Snr		P/L/A			P/L/A
'H'	0x48	HT1 Halt Selected		P/L/A			P/L/A
'I'	0x49	RWD Read Input	P/L				
'K'	0x4B	RWD Key Init Mode	P/L/A				
'L'	0x4C	RWD Set HF Mode	P/L				
'M'	0x4D	HT2 Read Miro			P/L	P/L	
'O'	0x4F	RWD Set Output	P/L				
'P'	0x50	HT1 Read Page		P/L/A			P/L/A
'R'	0x52	RWD Reset Sys	P/L				
'S'	0x53	HT1 Select Snr		P/L/A			P/L/A
'S'	0x53	HT1 Select Last		P/L/A			P/L/A
'V'	0x56	RWD Get Version	P/L/A				
'a'	0x61	HT1 Tag Authent		P/L			
'b'	0x62	HT1 Write Block		P/L/A			P/L/A
'c'	0x63	RWD Config Ports	P/L				
'e'	0x65	RWD EE Write	P/L				
'f'	0x66	RWD Read BCD	L				
'h'	0x68	RWD HF Reset	P/L/A				
'i'	0x69	RWD Read Ports	P/L				
'l'	0x6C	HT1 Poll Tags		P/L	P/L	P/L	
'o'	0x6F	RWD Write Ports	P/L				
'p'	0x70	HT1 Write Page		P/L/A			P/L/A
'r'	0x72	RWD Read LR Status	L				
'y'	0x79	HT1 Get Snr Select Halt		P/L/A			P/L/A
'y'	0x79	HTS Get Snr Reset		P/L/A			P/L/A
'z'	0x7A	HT1 Select Snr Direct		P/L/A			P/L/A
'z'	0x7A	HTS Select Snr Reset		P/L/A			P/L/A
	0x80	HT2 Get Snr			P/L/A		
	0x80	HT2 Get Snr Reset			P/L/A		
	0x81	HT2 Halt Selected			P/L/A		
	0x82	HT2 Read Page			P/L/A		
	0x83	HT2 Read Page Inv			P/L		
	0x84	HT2 Write Page			P/L/A		
	0x90	HT2 Poll Kb Tags	P/L	P/L	P/L		
	0x91	RWD SetModuleAdr	P/L				
	0x9E	HT2 Read PublicB			P/L/A		P/L/A
	0x9F	HT2 Read TTF			P/L/A		P/L/A

	0xA1	RWD_Set_Prox_Trm_Time	P/L				
	0xA2	HT1_Get_Snr_Adv		P/L/A			
	0xA4	RWD_Set_BCD_Offset	L				
	0xA6	RWD_Stop_Command	P/L				
	0xA7	RWD_SetBaudrate	P/L				
	0xA8	HTS_Tag_Authent					P/L/A
	0xA9	HTS_Tag_Authent_PW	V3.21				P/L/A
'x'	0x78	ReadAbicPage	A				
'X'	0x58	AbicCommand	A				

Additional commands for a special project requiring a special Reader-Hardware and -Software:

COMMAND BYTE		COMMAND NAME
	0x98	ReadAllPage
'v'	0x76	GetDspVersion

KeyInit Mode:

The KeyInit Mode is a mode of all HITAG Readers for using a set of personalization commands.

COMMAND BYTE		COMMAND NAME	READER
'C'	0x43	KI_Read_Control	P/L
'R'	0x52	KI_Reset (Switch to Operating Mode)	P/L
'V'	0x56	KI_Read_Secret2	P/L
'W'	0x57	KI_Write_Secret2	P/L/A
'X'	0x58	KI_Read_EE_Data	P/L
'Y'	0x59	KI_Write_EE_Data	P/L/A
'c'	0x63	KI_Write_Control	P/L
's'	0x73	KI_Write_SerNum	P/L/A
	0x90	KI_Read_Control2	P/L
	0x91	KI_Write_Control2	P/L
	0x92	KI_Read_ControlS	P/L
	0x93	KI_Write_ControlS	P/L

3.6 Status Byte

The read/write device returns a Status Byte indicating an error if different from 0.

The following Error Codes are defined:

VALUE	ERROR NAME	DESCRIPTION
0	no error	
-1	SERIAL ERROR	Error at the serial interface.
-3	NOTAG	There was no answer of a transponder detected by the read/write device.
-4	TIMEOUT	There is not enough energy available to write to the transponder.
-5	INCORRECT PASSWORD RWD	The HITAG 2 password of the read/write device is invalid.
-6	INCORRECT PASSWORD TAG	The HITAG 2 password of the transponder is invalid.
-7	AUTHENTICATION ERROR	An error occurred during the authentication process.
-8	ACKNOWLEDGEMENT ERROR	The acknowledgement was not received correctly.
-9	CRYPTOBLOCK NOT INIT	A cryptographic command was transmitted without authentication between the read/write device and transponder.
-10	EEPROM ERROR	EEPROM (of the read/write device) acknowledgement error.
-11	EEPROM WRONG OLD DATA	On comparison old and new data prove inconsistent.
-12	EEPROM WRITE PROTECTED	You attempted to write to the read/write device EEPROM, although writing was not allowed.
-13	EEPROM READ PROTECTED	You attempted to read from the read/write device EEPROM, although reading was not allowed.
-14	PIT DATA OVERFLOW	New PIT-Data were received by the host before the host-program read the old PIT-Data (error generated by C-Library during command <i>ReadPit</i>).
-15	CRC ERROR	Wrong CRC from a HITAG 1/S transponder in Advanced Protocol Mode.
-16	AC ERROR	Error in the anticollisiopn cycle
-17	SEL ERROR	Error during internal Select in the <i>Get_SNr_Reset</i> Command
-18	HALT ERROR	Error during internal Halt Selected in the <i>Get_SNr_Reset</i> Command
-20	ANTENNA OVERLOAD	Long Range Reader: Broken or badly detuned antenna (error only after command <i>ReadLRStatus</i>).

3.7 Command Description for Operating Mode

The Operating Mode is a mode of the reader for using a set of standard commands as described in the following.

In this mode the BCC is computed by EXOR-operation of all block data bytes including Block Length.

The command *KeyInitMode* is used to enter the KeyInit Mode (mode of the read/write device for using personalization commands), and a different set of commands becomes available.

3.7.1 HT1_Get_Snr

This command provides the serial number of a HITAG 1/S transponder in Standard Protocol Mode.

For further information on the Standard Protocol Mode see chapter „Transponders“.

C-Function: void HT1_Get_Snr (DWORD_T *snr, BYTE_T *more);

Header-File: Hitag1.H

Serial protocol:

HOST - READ/WRITE DEVICE

0x02	'G'	0x45
------	-----	------

READ/WRITE DEVICE - HOST

		7	0	-----	31	24	
n+2	Status	SNR-LSB	-----	SNR-MSB	more	BCC	

n = 0 if an error occurred (error code in Status).

n = 5 if data were read from a transponder (Status = 0).

more: Proximity Reader: *more* is always 0.

Long Range Reader: *more* equal to one indicates that there is at least one additional transponder in the reading area of the read/write device.

Status: 0 ... no error
 -1 ... SERIAL ERROR
 -3 ... NOTAG

3.7.2 HT1_Get_Snr_Adv

This command provides the serial number of a HITAG 1/S transponder and sets the transponder into Advanced Protocol Mode (command is not available for HITAG 1 transponders based on ASIC HT1 ICS30 01x ; only available for HITAG 1 transponders based on ASIC HT1 ICS30 02x).

For further information on the Advanced Protocol Mode see chapter „Transponders“.

C-Function: void HT1_Get_Snr_Adv (DWORD_T *snr, BYTE_T *more);

Header-File: Hitag1.H

Serial protocol:

HOST - READ/WRITE DEVICE

0x02	0xA2	0x45
------	------	------

READ/WRITE DEVICE - HOST

		7	0	-----	31	24	
n+2	Status	SNR-LSB	-----	SNR-MSB	more	BCC	

n = 0 if an error occurred (error code in Status).

n = 5 if data were read from a transponder (Status = 0).

more: Proximity Reader: *more* is always 0.

Long Range Reader: *more* equal to one indicates that there is at least one additional transponder in the reading area of the read/write device.

Status: 0 ... no error
 -1 ... SERIAL ERROR
 -3 ... NOTAG

3.7.3 HT1_GetSnrSelectHalt

This command provides the serial number of a HITAG 1/S transponder and sets the transponder into Advanced Protocol Mode (command is not available for HITAG 1 transponders based on ASIC HT1 ICS30 01x ; only available for HITAG 1 transponders based on ASIC HT1 ICS30 02x, this command does not work with Hitag S transponders configured in TTF or Authentication mode).

For further information on the Advanced Protocol Mode see chapter „Transponders“.

C-Function: void HT1_GetSnrSelectHalt (DWORD_T *snr, BYTE_T *more);

Header-File: Hitag1.H

Serial protocol:

HOST - READ/WRITE DEVICE

0x02	'y'	0x45
------	-----	------

READ/WRITE DEVICE - HOST

		7	0	-----	31	24	
n+2	Status	SNR-LSB	-----	SNR-MSB	more	BCC	

n = 0 if an error occurred (error code in Status).

n = 5 if data were read from a transponder (Status = 0).

more: Proximity Reader: *more* is always 0.

Long Range Reader: *more* equal to one indicates that there is at least one additional transponder in the reading area of the read/write device.

Status: 0 ... no error
 -1 ... SERIAL ERROR
 -3 ... NOTAG

3.7.4 HT1_Select_Snr

This command selects the HITAG 1/S transponder with a specified serial number. The content of the transponder's Configuration Page is returned.

If there is no such transponder in the field, a NOTAG error message is displayed.

ATTENTION: The serial number has to be the same as received with the preceding GetSnr.

C-Function: void HT1_Select_Snr (DWORD_T snr, DWORD_T *otp);

Header-File: Hitag1.H

Serial protocol:

HOST - READ/WRITE DEVICE

		7	0	-----	31	24
0x06	'S'	SNR-LSB	-----	SNR-MSB	BCC	

READ/WRITE DEVICE - HOST

		7	0	-----	31	24
n+2	Status	OTP-LSB	-----	OTP-MSB	BCC	

OTP: Configuration Page of HITAG 1/S

n = 0 if an error occurred (error code in Status).

n = 4 if data were read from a transponder (Status = 0).

Status: 0 ... no error
 -1 ... SERIAL ERROR
 -3 ... NOTAG

3.7.5 HT1_SelectSnrDirect

This command performs a HF-reset, performs a GetSnr Command without using the dedected serial number and selects the HITAG 1/S transponder with a specified serial number. The content of the transponder's Configuration Page is returned.

If there is no such transponder in the field, a NOTAG error message is displayed.

ATTENTION: The serial number does not have to be the same as received with the preceding GetSnr.

C-Function: void HT1_SelectSnrDirect (DWORD_T snr, DWORD_T *otp);

Header-File: Hitag1.H

Serial protocol:

HOST - READ/WRITE DEVICE

		7	0	-----	31	24
0x06	'z'	SNR-LSB	-----	SNR-MSB	BCC	

READ/WRITE DEVICE - HOST

		7	0	-----	31	24
n+2	Status	OTP-LSB	-----	OTP-MSB	BCC	

OTP: Configuration Page of HITAG 1/S

n = 0 if an error occurred (error code in Status).

n = 4 if data were read from a transponder (Status = 0).

Status:

- 0 ... no error
- 1 ... SERIAL ERROR
- 3 ... NOTAG

3.7.6 HT1_Select_Last

Selects the HITAG 1/S transponder with the serial number that was read executing the last, error-free *GetSnr* command.

This command is an abbreviated version of *SelectSnr* as no serial number has to be transmitted via the serial interface and the content of the Configuration Page is not returned.

C-Function: void HT1_Select_Last (void);

Header-File: Hitag1.H

Serial protocol:

HOST - READ/WRITE DEVICE

0x02	'S'	0x51
------	-----	------

READ/WRITE DEVICE - HOST

0x02	Status	BCC
------	--------	-----

Status: 0 ... no error
 -1 ... SERIAL ERROR
 -3 ... NOTAG

3.7.7 HT1_Halt_Selected

Puts the selected HITAG 1/S transponder into Halt Mode, which means that this transponder remains silent until it leaves and reenters the RF field. This command does not work with Hitag S transponders configured in TTF or Authentication mode.

You can reset a transponder previously turned off by *HaltSelected* using the command *HFRreset* or putting it out of RF field.

C-Function: void HT1_Halt_Selected (void);

Header-File: Hitag1.H

Serial protocol:

HOST - READ/WRITE DEVICE

0x02	'H'	0x4A
------	-----	------

READ/WRITE DEVICE - HOST

0x02	Status	BCC
------	--------	-----

Status: 0 ... no error
 -1 ... SERIAL ERROR
 -8 ... ACKNOWLEDGEMENT ERROR

3.7.8 HT1_Read_Page

Reads a page (4 bytes) of a selected HITAG 1/S transponder.

If no transponder is selected, a NOTAG message will be generated even if there is a transponder in the communication field of the antenna.

Using the byte *-crypto-* you define whether you want to work in Plain or in Crypto Mode. Access to the secret area is only possible in Crypto Mode after a mutual authentication.

If *-crypto-* equals 1 (Crypto Mode) and you did not run an authentication procedure before, Status will be set to -9.

C-Function: void HT1_Read_Page (BYTE_T crypto, BYTE_T pagenr, char *data);

Header-File: Hitag1.H

Serial protocol:

HOST - READ/WRITE DEVICE

0x04	'P'	crypto	pagenr	BCC
------	-----	--------	--------	-----

crypto: 0x00 ... Plain Mode
 0x01 ... Crypto Mode
pagenr: page number

READ/WRITE DEVICE - HOST

n+2	Status	data[0]	data[3]	BCC
-----	--------	---------	-------	---------	-----

n = 0 if an error occurred (error code in Status).

n = 4 if data were read from a transponder (Status = 0).

Status: 0 ... no error
 -1 ... SERIAL ERROR
 -3 ... NOTAG
 -9 ... CRYPTOBLOCK NOT INIT

3.7.9 HT1_Read_Block

Reads a block (16 bytes) of the selected HITAG 1/S transponder.

If no transponder is selected, a NOTAG message will be generated even if there is a transponder in the communication field of the antenna.

The start address is specified by *-pagenr-*. Data is read from the start address until the end of the corresponding block. Thus a data length of 4, 8, 12 or 16 bytes is possible.

Use byte *-crypto-* to define whether you want to work in Plain or in Crypto Mode.

Access to the secret area is only possible in Crypto Mode after a mutual authentication.

If *-crypto-* equals 1 (Crypto Mode) and you did not run an authentication procedure before, Status will be set to -9.

C-Function: void HT1_Read_Block (BYTE_T crypto, BYTE_T pagenr, char *data);

Header-File: Hitag1.H

Serial protocol:

HOST - READ/WRITE DEVICE

0x04	'B'	crypto	pagenr	BCC
------	-----	--------	--------	-----

crypto: 0x00 ... Plain Mode
 0x01 ... Crypto Mode

pagenr: page number (for start address)

READ/WRITE DEVICE - HOST

n+2	Status	data[0]	data[n-1]	BCC
-----	--------	---------	-------	-----------	-----

n = 0 if an error occurred (error code in Status).

n = 4, 8, 12, 16 depending on the page address if data were read from a transponder (Status = 0).

Status: 0 ... no error
 -1 ... SERIAL ERROR
 -3 ... NOTAG
 -9 ... CRYPTOBLOCK NOT INIT

3.7.10 HT1_Write_Page

Writes a page (4 bytes) to the selected HITAG 1/S transponder.

If no transponder is selected, a NOTAG message will be generated even if there is a transponder in the communication field of the antenna.

Use byte *-crypto-* to define whether you want to work in Plain or in Crypto Mode.

Access to the secret area is only possible in Crypto Mode after a mutual authentication.

If *-crypto-* equals 1 (Crypto Mode) and you did not run an authentication process before, Status will be set to -9.

ATTENTION: To substantially increase the data reliability we strictly recommend to read the previously written data (read after write).

C-Function: void HT1_Write_Page (BYTE_T crypto, BYTE_T pagenr, char *data);

Header-File: Hitag1.H

Serial protocol:

HOST - READ/WRITE DEVICE

0x08	'p'	crypto	Pagenr	data[0]	data[3]	BCC
------	-----	--------	--------	---------	-------	---------	-----

crypto: 0x00 ... Plain Mode
 0x01 ... Crypto Mode

pagenr: page number

READ/WRITE DEVICE - HOST

0x02	Status	BCC
------	--------	-----

Status: 0 ... no error
 -1 ... SERIAL ERROR
 -3 ... NOTAG
 -4 ... TIMEOUT
 -9 ... CRYPTOBLOCK NOT INIT

3.7.11 HT1_Write_Block

Writes a block (16 bytes) to the selected HITAG 1/S transponder.

If no transponder is selected, a NOTAG message will be generated even if there is a transponder in the communication field of the antenna.

The start address is specified by *-pagenr-*. Data is written from the start address until the end of the corresponding block. Thus a data length of 4, 8, 12 or 16 bytes is possible.

Use byte *-crypto-* to define whether you want to work in Plain or in Crypto Mode.

Access to the secret area is only possible in Crypto Mode after a mutual authentication.

If *-crypto-* equals 1 (Crypto Mode) and you did not run an authentication procedure before, Status will be set to -9.

ATTENTION: To substantially increase the data reliability we strictly recommend to read the previously written data (read after write).

C-Function: void HT1_Write_Block (BYTE_T crypto, BYTE_T pagenr, char *data);

Header-File: Hitag1.H

Serial protocol:

HOST - READ/WRITE DEVICE

n+4	'b'	crypto	pagenr	data[0]	data[n-1]	BCC
-----	-----	--------	--------	---------	-------	-----------	-----

crypto: 0x00 ... Plain Mode
 0x01 ... Crypto Mode

pagenr: page number (for start address)

READ/WRITE DEVICE - HOST

0x02	Status	BCC
------	--------	-----

Status: 0 ... no error
 -1 ... SERIAL ERROR
 -3 ... NOTAG
 -4 ... TIMEOUT
 -9 ... CRYPTOBLOCK NOT INIT

3.7.12 HT1_Tag_Authent

Carries out the single authentication procedure for HITAG 1 transponders (authentication of the transponder). The authentication procedure is aborted after sending the transponder logdata.

Using *-keyinfo-* you can choose between Key/Logdata Set A and B.

This command can be used - e.g. - to check if Keys and Logdata in the transponder and the read/write device are the same. ("Check, if the transponder is member of the same 'family' as the read/write device").

ATTENTION: You cannot use any Crypto commands after *TagAuthent*.

After this abbreviated authentication procedure the transponder can only be accessed using *GetSnr* or the command *HFReset*.

C-Function: void HT1_Tag_Authent (BYTE_T keyinfo);

Header-File: Hitag1.H

Serial protocol:

HOST - READ/WRITE DEVICE

0x03	'a'	keyinfo	BCC
------	-----	---------	-----

keyinfo: 0x00 ... Key/Logdata Set A
 0x01 ... Key/Logdata Set B

READ/WRITE DEVICE - HOST

0x02	Status	BCC
------	--------	-----

Status: 0 ... no error
 -1 ... SERIAL ERROR
 -7 ... AUTHENTICATION ERROR

3.7.13 HT1_Mutual_Authent

Carries out the full authentication procedure between the transponder and the read/write device. After this mutual authentication you are allowed to edit areas which can only be accessed in Crypto Mode.

Using *-keyinfo-* you can choose between Key/Logdata Set A and B.

Use a Plain command (that is still encrypted), *HFReset* or *GetSnr* (resets the already selected transponder) to exit this mode.

C-Function: void HT1_Mutual_Authent (BYTE_T keyinfo);

Header-File: Hitag1.H

Serial protocol:

HOST - READ/WRITE DEVICE

0x03	'A'	keyinfo	BCC
------	-----	---------	-----

keyinfo: 0x00 ... Key/Logdata Set A
 0x01 ... Key/Logdata Set B

READ/WRITE DEVICE - HOST

0x02	Status	BCC
------	--------	-----

Status: 0 ... no error
 -1 ... SERIAL ERROR
 -7 ... AUTHENTICATION ERROR

3.7.14 HTS_GetSnrReset

This command provides the serial number of a HITAG 1/S transponder. If bit4 of *mode* is set the new command set of the HitagS transponder is used, the whole anticollision is performed and one transponder selected and halted (with the new HitagS command Select_Quiet) reader internally.

C-Function: void HTS_GetSnrReset (BYTE mode, DWORD_T *snr, BYTE_T *more);

Header-File: HitagS.H

Serial protocol:

HOST - READ/WRITE DEVICE

0x03	'y'	mode	BCC
------	-----	------	-----

READ/WRITE DEVICE - HOST

		7	0	-----	31	24	
n+2	Status	SNR-LSB	-----	SNR-MSB	more	BCC	

n = 0 if an error occurred (error code in Status).

n = 5 if data were read from a transponder (Status = 0).

mode: bit0: 0... without HF Reset
1... with HF Reset

bit2: 0... Hitag standard protocoll
1... Hitag advance protocoll

bit4: 0... Diffent Tranponder types expected
1... HitagS Transponder expected

more: Proximity Reader: *more* is always 0.
Long Range Reader: *more* equal to one indicates that there is at least one additional transponder in the reading area of the read/write device.

Status: 0 ... no error
-1 ... SERIAL ERROR
-3 ... NOTAG
-16 ... AC ERROR
-17 ... SEL ERROR
-18 ... HALT ERROR

3.7.15 HTS_SelectSnrReset

This command performs a HF-reset depending on *mode*, a GetSnr Command without using the dedected serial number and selects the HITAG S transponder with the specified serial number. The content of the transponder's Configuration Page is returned.

If there is no such transponder in the field, a NOTAG error message is displayed.

ATTENTION: The serial number does not have to be the same as received with the preceding GetSnr.

C-Function: void HTS_SelectSnrReset (BYTE mode, DWORD_T snr, DWORD_T *otp);

Header-File: HitagS.H

Serial protocol:

HOST - READ/WRITE DEVICE

		7	0	-----	31	24	
0x07	'z'	SNR-LSB	-----	SNR-MSB	mode	BCC	

READ/WRITE DEVICE - HOST

		7	0	-----	31	24	
n+2	Status	OTP-LSB	-----	OTP-MSB	BCC		

OTP: Configuration Page of HITAG 1/S

n = 0 if an error occurred (error code in Status).

n = 4 if data were read from a transponder (Status = 0).

mode: bit0: 0... without HF Reset
1... with HF Reset
bit2: 0... Hitag standard protocoll
1... Hitag advance protocoll

Status: 0 ... no error
-1 ... SERIAL ERROR
-3 ... NOTAG

3.7.16 HTS_TagAuthent

Carries out the full authentication procedure between the transponder and the read/write device. After this mutual authentication you are allowed to access a HitagS transponder configured in Authentication Mode. After Authentication the data are transferred plain.

C-Function: void HTS_TagAuthent ();

Header-File: HitagS.H

Serial protocol:

HOST - READ/WRITE DEVICE

0x02	0xA8	BCC
------	------	-----

READ/WRITE DEVICE - HOST

0x02	Status	BCC
------	--------	-----

Status: 0 ... no error
 -1 ... SERIAL ERROR
 -7 ... AUTHENTICATION ERROR

3.7.17 HTS_TagAuthent_PW

Carries out the full authentication procedure between the transponder and the read/write device. After this mutual authentication you are allowed to access a HitagS transponder configured in Authentication Mode. After Authentication the data are transferred plain.

C-Function: void HTS_TagAuthent ();

Header-File: HitagS.H

Serial protocol:

HOST - READ/WRITE DEVICE

0x02	0xA9	BCC
------	------	-----

READ/WRITE DEVICE - HOST

0x06	Status	n.d.	PWH	PWL1	PWL0	BCC
------	--------	------	-----	------	------	-----

Status: 0 ... no error
 -1 ... SERIAL ERROR
 -7 ... AUTHENTICATION ERROR

3.7.18 HT2_Get_Snr

This command is applied to a HITAG 2 transponder being in Password or Crypto Mode. The command selects the transponder and provides its serial number and Configuration Byte *-config-*.

If the byte *-Status-* shows „no error“ the transponder is selected and ready for read or write accesses.

The byte *-mode-* selects one of two possible modes: Password or Crypto.

C-Function: void HT2_Get_Snr (BYTE_T mode, DWORD_T *snr, BYTE_T *config);

Header-File: Hitag2.H

Serial protocol:

HOST - READ/WRITE DEVICE

0x03	0x80	mode	BCC
------	------	------	-----

mode: 0x00 ... Password
 0x01 ... Crypto

READ/WRITE DEVICE - HOST

		7	0	-----	31	24	
n+2	Status	SNR-LSB	-----	SNR-MSB	config	BCC	

config: Configuration Byte of HITAG 2

n = 0 if an error occurred (error code in Status).

n = 5 if data were read from a transponder (Status = 0).

Status: 0 ... no error
 -1 ... SERIAL ERROR
 -3 ... NOTAG
 -5 ... INCORRECT PASSWORD RWD
 -6 ... INCORRECT PASSWORD TAG
 -7 ... AUTHENTICATION ERROR

3.7.19 HT2_Get_Snr_Reset

This command is applied to a HITAG 2 transponder which is currently not in Password or Crypto Mode but in one of the Public Modes. The command selects the transponder and provides its serial number and Configuration Byte.

If the byte *-Status-* shows „no error“ the transponder is selected and ready for read or write accesses.

The byte *-mode-* decides whether the selection process for the transponder is done corresponding to the Password Mode or the Crypto Mode.

C-Function: void HT2_Get_Snr_Reset (BYTE_T mode, DWORD_T *snr,
BYTE_T *config);

Header-File: Hitag2.H

Serial protocol:

HOST - READ/WRITE DEVICE

0x03	0x80	mode	'M'	BCC
------	------	------	-----	-----

mode: 0x00 ... Password
 0x01 ... Crypto

READ/WRITE DEVICE - HOST

		7	0	-----	31	24	
n+2	Status	SNR-LSB	-----	SNR-MSB	config	BCC	

config: Configuration Byte of HITAG 2
n = 0 if an error occurred (error code in Status).
n = 5 if data were read from a transponder (Status = 0).

Status: 0 ... no error
 -1 ... SERIAL ERROR
 -3 ... NOTAG
 -5 ... INCORRECT PASSWORD RWD
 -6 ... INCORRECT PASSWORD TAG
 -7 ... AUTHENTICATION ERROR

3.7.20 HT2_Halt_Selected

Puts the selected HITAG 2 transponder into Halt Mode, which means that this transponder remains silent until it leaves the RF field.

You can reset a transponder previously turned off by *HaltSelected_LT* using the command *HFRreset* or putting it out of RF field.

C-Function: void HT2_Halt_Selected (void);

Header-File: Hitag2.H

Serial protocol:

HOST - READ/WRITE DEVICE

0x02	0x81	0x83
------	------	------

READ/WRITE DEVICE - HOST

0x02	Status	BCC
------	--------	-----

Status: 0 ... no error
 -1 ... SERIAL ERROR
 -8 ... ACKNOWLEDGEMENT ERROR

3.7.21 HT2_Read_Page

Reads a page (4 bytes) of a selected HITAG 2 transponder.

If no transponder is selected, a NOTAG message will be generated.

This command should be used together with *ReadPageInv_LT* to compare plain data with the bit-inverted data to gain maximum data reliability.

C-Function: void HT2_Read_Page (BYTE_T pagenr, char *data);

Header-File: Hitag2.H

Serial protocol:

HOST - READ/WRITE DEVICE

0x03	0x82	pagenr	BCC
------	------	--------	-----

pagenr: page number

READ/WRITE DEVICE - HOST

n+2	Status	data[0]	data[3]	BCC
-----	--------	---------	-------	---------	-----

n = 0 if an error occurred (error code in Status).

n = 4 if data were read from a transponder (Status = 0).

Status: 0 ... no error
 -1 ... SERIAL ERROR
 -3 ... NOTAG

3.7.22 HT2_Read_Page_Inv

Reads a bit-inverted page (4 bytes) of a selected HITAG 2 transponder.
If no transponder is selected, a NOTAG message will be generated.

This command should be used together with *ReadPage_LT* to compare plain data with the bit-inverted data to gain maximum data reliability.

C-Function: void HT2_Read_Page_Inv (BYTE_T pagenr, char *data);

Header-File: Hitag2.H

Serial protocol:

HOST - READ/WRITE DEVICE

0x03	0x83	pagenr	BCC
------	------	--------	-----

pagenr: page number

READ/WRITE DEVICE - HOST

n+2	Status	data[0]	data[3]	BCC
-----	--------	---------	-------	---------	-----

n = 0 if an error occurred (error code in Status).

n = 4 if data were read from a transponder (Status = 0).

Status: 0 ... no error
 -1 ... SERIAL ERROR
 -3 ... NOTAG

3.7.23 HT2_Write_Page

Writes a page (4 bytes) onto the selected HITAG 2 transponder.
If no transponder is selected, a NOTAG message will be generated.

ATTENTION: To check if *HT2_Write_Page* was successful it is important that the immediately following command is a *HT2_Read_Page*. If *HT2_Read_Page* does not return „no error“ and the right data, you have to repeat *HT2_Write_Page*.

C-Function: void HT2_Write_Page (BYTE_T pagenr, char *data);

Header-File: Hitag2.H

Serial protocol:

HOST - READ/WRITE DEVICE

0x07	0x84	pagenr	data[0]	data[3]	BCC
------	------	--------	---------	-------	---------	-----

pagenr: page number

READ/WRITE DEVICE - HOST

0x02	Status	BCC
------	--------	-----

Status: 0 ... no error
 -1 ... SERIAL ERROR
 -3 ... NOTAG
 -4 ... TIMEOUT

3.7.24 HT2_Read_PublicB

This command sets the read/write device to Permanent Reading Mode for HITAG 2 transponders being in Public Mode B.

The read/write device attempts continuously to synchronize on and read a HITAG 2 transponder in Public Mode B. If it succeeds and all checks report positive results, the device sends the 16 data bytes (a 128-bit-stream that has to be prepared afterwards for subsequent treatment) via the serial interface. After that the read/write device returns to Normal Mode. The software running on the host has to decode the read data depending on the chosen data protocol.

To put the read/write device back to normal mode, a *StopCommand* should be sent. Do not use a *Reset*, since *Reset* can cause undesirable side effects (resetting output pins).

Since the tag sends its 128-bit data continuously, the user must store its data on the tag in a way which allows for synchronization.

C-Function: void HT2_Read_PublicB (BYTE_T *data);

Header-File: Hitag2.H

Serial protocol:

HOST - READ/WRITE DEVICE

0x02	0x9E	BCC
------	------	-----

READ/WRITE DEVICE - HOST

0x12	Status	data[0]	-----	data[15]	BCC
------	--------	---------	-------	----------	-----

Status: 0 ... no error
 -1 ... SERIAL ERROR

3.7.25 HT2_Read_Miro

Sets the read/write device to Permanent Reading Mode for MIRO compatible transponders.

In this mode you can read either HITAG 2 transponders in Public Mode A or MIRO transponders.

The unique serial number of a MIRO transponder consists of 5 bytes.

The read/write device attempts continuously to synchronize on and read a MIRO transponder. If it succeeds and all checks report positive results, the device sends the 5 data bytes via the serial interface. After that the read/write device returns to Normal Mode.

To put the read/write device back to normal mode, a *StopCommand* should be sent. Do not use a *Reset*, since *Reset* can cause undesirable side effects (resetting output pins).

MIRO-compatible data protocol for using HITAG 2 transponders in Public Mode A (data is stored on Pages 4 and 5 of a HITAG 2 transponder):

9 bit header (= '1')	9 bit
10 * 4 bit ID data + 10 * 1 bit even parity	50 bit
4 bits even parity for columns (of ID data nibbles)	4 bit
last bit (= '0')	1 bit
<hr/>	
total	64 bit

C-Function: void HT2_Read_Miro (char *data);

Header-File: Hitag2.H

Serial protocol:

HOST - READ/WRITE DEVICE

0x02	'M'	0x4F
------	-----	------

READ/WRITE DEVICE - HOST

0x07	Status	data[0]	-----	data[4]	BCC
------	--------	---------	-------	---------	-----

Status: 0 ... no error
 -1 ... SERIAL ERROR

3.7.26 HT2_Poll_Tags

This command sets the read/write device to Permanent Reading Mode for specified types of transponders.

The read/write device attempts continuously to synchronize on and read specified types of transponders. If it succeeds and all checks report positive results, the device sends data for transponder identification via the serial interface. After that the read/write device returns to Normal Mode.

Using the byte *-mode-* you select the types of transponders for poll-operation.

To avoid conflicts it is important to set only one bit at a time for following transponder types:

- PIT or HITAG 2 PublicC or HITAG 2 PublicB
- HITAG 1 Standard Protocol Mode or HITAG 1 Advanced Protocol Mode

If bit 3 (HITAG 2 Password Mode) or bit 4 (HITAG 2 Crypto Mode) is selected, we recommend to activate the „Check PW TAG“ option in Control_LT (located in the EEPROM of the read/write device ... see Chapter „Personalization“) to reduce the possibility to erroneously identify other types of (especially read-only) transponders as HITAG 2 Password or HITAG 2 Crypto.

C-Function: void HT2_Poll_Tags (BYTE_T mode, char *data);

Header-File: Hitag2.H

Serial protocol:

HOST - READ/WRITE DEVICE

0x03	'I'	mode	BCC
------	-----	------	-----

mode:

7	6	5	4	3	2	1	0
Poll HITAG 1 Advanced Protocol Mode	Poll HITAG 2 Public Mode B	Poll HITAG S Advanced Protocol Mode	Poll HITAG 2 Crypto Mode	Poll HITAG 2 Password Mode	Poll HITAG S Crypto Mode	Poll Miro / HITAG 2 Public Mode A	Poll HITAG 1 Standard Protocol Mode

READ/WRITE DEVICE - HOST

If polling for HITAG 1 Standard Protocol Mode was successful:

7	0	-----	31	24			
0x08	Status	0x01	SNR-LSB	-----	SNR-MSB	more	BCC

more: Proximity Reader: *more* is always 0.

Long Range Reader: *more* equal to one indicates that there is at least one additional transponder in the reading area of the read/write device.

If polling for HITAG 1 Advanced Protocol Mode was successful:

7 0 ----- 31 24

0x08	Status	0x80	SNR-LSB	-----	SNR-MSB	more	BCC
------	--------	------	---------	-------	---------	------	-----

more: Proximity Reader: *more* is always 0.

Long Range Reader: *more* equal to one indicates that there is at least one additional transponder in the reading area of the read/write device.

If polling for Miro / HITAG 2 Public Mode A was successful:

0x08	Status	0x02	data[0]	-----	data[4]	BCC
------	--------	------	---------	-------	---------	-----

If polling for HitagS Crypto was successful:

7 0 ----- 31 24

0x08	Status	0x80	SNR-LSB	-----	SNR-MSB	more	BCC
------	--------	------	---------	-------	---------	------	-----

If polling for HITAG 2 Password Mode was successful:

7 0 ----- 31 24

0x08	Status	0x08	SNR-LSB	-----	SNR-MSB	config	BCC
------	--------	------	---------	-------	---------	--------	-----

If polling for HITAG 2 Crypto Mode was successful:

7 0 ----- 31 24

0x08	Status	0x10	SNR-LSB	-----	SNR-MSB	config	BCC
------	--------	------	---------	-------	---------	--------	-----

If polling for HitagS Advanced Protocol Mode was successful:

7 0 ----- 31 24

0x08	Status	0x80	SNR-LSB	-----	SNR-MSB	more	BCC
------	--------	------	---------	-------	---------	------	-----

If polling for HITAG 2 Public Mode B was successful:

0x13	Status	0x40	data[0]	-----	data[15]	BCC
------	--------	------	---------	-------	----------	-----

Status: 0 ... no error
 -1 ... SERIAL ERROR

3.7.27 RWD_Poll_Kb_Tags

This command polls once for a transponder in the RF-field of the read/write device antenna and reads the keyboard-buffer and the digital inputs IN1 and IN2. The read/write device does not enter the Permanent Reading Mode.

Port 0 of the HITAG Communication Controller is used to connect a keyboard-matrix.

See document „HT RC100 HITAG™ Communication Controller“ for information about hardware connections and key-decoding.

ATTENTION: To use the keyboard-decoding function a special hardware with connected Port 0 signals is required (Port 0 signals are not available on pin connectors of Philips Core Modules).

C-Function: void RWD_Poll_KB_Tags (BYTE_T mode, char *data);

Header-File: HitagRWD.H

Serial protocol:

HOST - READ/WRITE DEVICE

0x03	0x90	mode	BCC
------	------	------	-----

mode:

- 0x00 ... poll keyboard-matrix and inputs
- 0x80 ... poll serial number of HITAG 1 transponder in Standard Protocol Mode, keyboard-matrix and inputs
- 0x81 ... poll serial number of HITAG 2 transponder in Crypto Mode, keyboard-matrix and inputs
- 0x82 ... poll serial number of HITAG 2 transponder in Password Mode, keyboard matrix and inputs
- 0x83 ... poll serial number of HITAG 1 transponder in Advanced Protocol Mode, keyboard-matrix and inputs

READ/WRITE DEVICE - HOST

n+2	Info	optional 4 bytes keyboard-buffer	optional 4 bytes serial number	BCC
-----	------	----------------------------------	--------------------------------	-----

n = 0 no transponder in RF-field

n = 4 keyboard-buffer not empty or transponder in RF-field

n = 8 keyboard-buffer not empty and transponder in RF-field

info:

- bit0 ... state of input IN1
- bit1 ... state of input IN2
- bit6 ... when set protocol contains keyboard-buffer
- bit7 ... when set protocol contains serial number

Keyboard-Buffer (appended to protocol when keyboard-buffer is not empty):

Keyb[0]	Keyb[1]	Keyb[2]	Keyb[3]
---------	---------	---------	---------

Keyb[0] Bits 4-7 oldest key-code

Keyb[0] Bits 0-3

⋮

⋮

Keyb[3] Bits 4-7 second newest key-code

Keyb[3] Bits 0-3 newest key-code

Serial number (appended to the protocol if a transponder of requested type was found):

7	0	-----	31	24
SNR-LSB		-----	SNR-MSB	

3.7.28 RWD_Get_Version

This command retrieves the serial number of the read/write device, the version number of the HITAG Communication Controller software and its date of creation.

C-Function: void RWD_Get_Version (char *data);

Header-File: HitagRWD.H

Serial protocol:

HOST - READ/WRITE DEVICE

0x02	'V'	BCC
------	-----	-----

READ/WRITE DEVICE - HOST

0x1D	Status	data[0]	-----	data[26]	BCC
------	--------	---------	-------	----------	-----

data[0] ... data[7]: Version (format: Vx.yy.zz)
data[8] ... data[15]: Date (format: dd-mm-yy)
data[16] ... data[26]: Serial number (11 characters)

Status: 0 ... no error
 -1 ... SERIAL ERROR

3.7.29 RWD_Reset_Sys

This command resets basic functions of the read/write device. All port-pins of the HITAG Communication Controller are reset to an initial state (output pins are set to '0', input pins are set to '1').

You should not interrupt the Permanent Reading Mode (activated after *ReadMiro*, *ReadPit*, ...) or the permanent writing mode (e.g. activated after *WritePit*) of the read/write device by invoking this command, since *Reset* can cause undesirable side effects (resetting output pins). Use *StopCommand* instead.

C-Function: void RWD_Reset_Sys (void);

Header-File: HitagRWD.H

Serial protocol:

HOST - READ/WRITE DEVICE

0x02	'R'	0x50
------	-----	------

READ/WRITE DEVICE - HOST

0x02	Status	BCC
------	--------	-----

Status: 0 ... no error
 -1 ... SERIAL ERROR

3.7.30 RWD_HF_Reset

This function turns off the RF-part of the read/write device for a certain time (about 100 ms in a Proximity Reader, about 40 ms in a Long Range Reader).

This means that all HITAG transponders are reset and transponders that were in Halt Mode will respond again.

C-Function: void RWD_HF_Reset (void);

Header-File: HitagRWD.H

Serial protocol:

HOST - READ/WRITE DEVICE

0x02	'h'	0x6A
------	-----	------

READ/WRITE DEVICE - HOST

0x02	Status	BCC
------	--------	-----

Status: 0 ... no error
 -1 ... SERIAL ERROR

3.7.31 RWD_Stop_Cmd

The command *RWD_Stop_Cmd* interrupts the Permanent Reading Mode (activated after *ReadMiro*, *ReadPit*, ...) or the permanent writing mode (e.g. activated after *WritePit*) of the read/write device.

You should not use the command *RWD_Reset_Sys* instead, since *RWD_Reset_Sys* can cause undesirable side effects (resetting output pins).

C-Function: void RWD_Stop_Cmd (void);

Header-File: HitagRWD.H

Serial protocol:

HOST - READ/WRITE DEVICE

0x02	0xA6	BCC
------	------	-----

READ/WRITE DEVICE - HOST

0x02	Status	BCC
------	--------	-----

Status: 0 ... no error
 -1 ... SERIAL ERROR

3.7.32 RWD_SetBaudrate

The command *RWD_SetBaudRate* changes the baud rate from the default value 9600 to the new baud rate defined in "mode".

C-Function: void RWD_SetBaudrate (void);

Header-File: HitagRWD.H

Serial protocol:

HOST - READ/WRITE DEVICE

0x03	0xA7	mode	BCC
------	------	------	-----

READ/WRITE DEVICE - HOST

0x02	Status	BCC
------	--------	-----

mode: 0... 4800 baud Not in use
 1... 9600 baud
 2... 14400 baud
 3... 19200 baud
 4... 38400 baud
 5... 57600 baud
 6... 115200 baud Causes problems in some systems

Status: 0 ... no error
 -1 ... SERIAL ERROR

3.7.33 RWD_Read_Input

You can read input-ports of the HITAG Communication Controller by using the command *ReadInput*.

ATTENTION:

- **Pins are internally pulled up!**
- **Using Philips Long Range Readers the state of input In1 is inverted (input is buffered by an inverting schmitt trigger input driver).**

There are certain restrictions concerning the applied hardware:

In1: available for Proximity and Long Range Readers

In2: available only for Proximity Readers

C-Function: void RWD_Read_Input (BYTE_T *input);

Header-File: HitagRWD.H

Serial protocol:

HOST - READ/WRITE DEVICE

0x02	T	0x4B
------	---	------

READ/WRITE DEVICE - HOST

0x03	Status	input	BCC
------	--------	-------	-----

input:

7	6	5	4	3	2	1	0
x	x	x	x	x	x	In2	In1

Proximity Reader (In1,In2): 0 ... reset (0 V) 1 ... set (5 V)

Long Range Reader (In1): 0 ... set (5 V) 1 ... reset (0 V)

Status: 0 ... no error
-1 ... SERIAL ERROR

3.7.34 RWD_Set_Output

You can set (5 V) or reset (0 V) output-ports of the HITAG Communication Controller by *SetOutput*.

ATTENTION: Using Philips Long Range Readers the state of output Out1 is inverted (output is buffered by an inverting CMOS driver).

There are certain restrictions concerning the applied hardware:

Out1 (P2.0): available for Proximity and Long Range Readers

Out2 (P2.1): available only for Proximity Readers

Out3 (P1.4): available only with a special hardware including connection of this pin (signal is not available on pin connectors of Philips Core Module)

Out4 (P2.7): available only with a special hardware including connection of this pin (signal is not available on pin connectors of Philips Core Module)

C-Function: void RWD_Set_Output (BYTE_T output);

Header-File: HitagRWD.H

Serial protocol:

HOST - READ/WRITE DEVICE

0x03	'O'	output	BCC
------	-----	--------	-----

output:

7	6	5	4	3	2	1	0
x	x	x	x	(Out4)	(Out3)	Out2	Out1

Proximity Reader (Out1,Out2): 0 ... reset (0 V) 1 ... set (5 V)

Long Range Reader (Out1): 0 ... set (5 V) 1 ... reset (0 V)

READ/WRITE DEVICE - HOST

0x02	Status	BCC
------	--------	-----

Status: 0 ... no error

-1 ... SERIAL ERROR

3.7.35 RWD_Config_Ports

This command writes a new Port 0 Configuration-Byte into the EEPROM of the read/write device.

The Port 0 Configuration-Byte (*-config-*) defines, whether a Port-0-pin of the HITAG Communication Controller has to be handled as an input or as an output.

Initial value stored in the EEPROM of a delivered read/write device:

config = 0x00

ConfigPorts automatically initializes the status of input-configured pins to '1' (5V). The status of output-configured pins is left unchanged.

ATTENTION: To use commands referring to Port 0 you need a special hardware with connected Port 0 signals is required (Port 0 signals are not available on pin connectors of Philips Core Modules).

Power-Up or Reset of the read/write device:

- The Port 0 Configuration-Byte is not lost (because stored in EEPROM).
- Input-configured pins are initialized to HIGH (5 V), output-configured to LOW (0 V) by the read/write device operating system.

C-Function: void RWD_Config_Ports (BYTE_T config);

Header-File: HitagRWD.H

Serial protocol:

HOST - READ/WRITE DEVICE

0x03	'c'	config	BCC
------	-----	--------	-----

config:

7	6	5	4	3	2	1	0
P0.7	P0.6	P0.5	P0.4	P0.3	P0.2	P0.1	P0.0

0 ... configure as input

1 ... configure as output

READ/WRITE DEVICE - HOST

0x02	Status	BCC
------	--------	-----

Status: 0 ... no error
 -1 ... SERIAL ERROR

3.7.36 RWD_Read_Ports

This command reads the status of those port Pins of the HITAG Communication Controller (Port 0) that are configured as inputs.

ATTENTION: To use commands referring to Port 0 you need a special hardware with connected Port 0 signals is required (Port 0 signals are not available on pin connectors of Philips Core Modules).

Bit-positions of output-configured pins are read as '0'.

C-Function: void RWD_Read_Ports (BYTE_T *input);

Header-File: HitagRWD.H

Serial protocol:

HOST - READ/WRITE DEVICE

0x02	'i'	BCC
------	-----	-----

READ/WRITE DEVICE - HOST

0x03	Status	input	BCC
------	--------	-------	-----

input:

7	6	5	4	3	2	1	0
P0.7	P0.6	P0.5	P0.4	P0.3	P0.2	P0.1	P0.0

0 ... reset (0 V)

1 ... set (5 V)

Status: 0 ... no error
 -1 ... SERIAL ERROR

3.7.37 RWD_Write_Ports

This command changes the status of those port pins of the HITAG Communication Controller (Port 0) that are configured as outputs.

ATTENTION: To use commands referring to Port 0 you need a special hardware with connected Port 0 signals is required (Port 0 signals are not available on pin connectors of Philips Core Modules).

Write accesses to input-configured pins always result in writing '1' (5 V) to the pins.

C-Function: void RWD_Write_Ports (BYTE_T mode, BYTE_T output);

Header-File: HitagRWD.H

Serial protocol:

HOST - READ/WRITE DEVICE

0x04	'o'	output	Mode	BCC
------	-----	--------	------	-----

output:

7	6	5	4	3	2	1	0
P0.7	P0.6	P0.5	P0.4	P0.3	P0.2	P0.1	P0.0

- mode:*
- 0 ... The bits in *-output-* are directly written to the output-configured port pins.
 - 1 ... The current status of the output-configured port pins is AND-combined with the bits in *-output-*. The result is written to the output-configured port pins.
 - 2 ... The current status of the output-configured port pins is OR-combined with the bits in *-output-*. The result is written to the output-configured port pins.
 - 3 ... The current status of the output-configured port pins is EXOR-combined with the bits in *-output-*. The result is written to the output-configured port pins.

READ/WRITE DEVICE - HOST

0x02	Status	BCC
------	--------	-----

- Status:
- 0 ... no error
 - 1 ... SERIAL ERROR

3.7.38 RWD_EE_Read

Reads - starting with the chosen address - up to 16 data bytes from the user memory in the EEPROM of the HITAG read/write device. If you reach the limit of the address area the command is finished.

C-Function: void RWD_EE_Read (BYTE_T addr, BYTE_T bytenmb, char *data);

Header-File: HitagRWD.H

Serial protocol:

HOST - READ/WRITE DEVICE

0x04	'E'	addr	bytenmb	BCC
------	-----	------	---------	-----

addr: EEPROM user address ($0 \leq addr \leq 84$)

bytenmb: number of bytes to read ($1 \leq bytenmb \leq 16$)

READ/WRITE DEVICE - HOST

n + 2	status	data[0]	data[n-1]	BCC
-------	--------	---------	-------	-----------	-----

Status: 0 ... no error
 -1 ... SERIAL ERROR
 -10 ... EEPROM ERROR

3.7.39 RWD_EE_Write

Writes - starting with the chosen address - up to 16 data bytes into the user memory of the EEPROM of the read/write device. If you reach the limit of the address area the command is finished.

C-Function: void RWD_EE_Write (BYTE_T addr, BYTE_T bytenmb, char *data);

Header-File: HitagRWD.H

Serial protocol:

HOST - READ/WRITE DEVICE

bytenmb+4	'e'	addr	bytenmb	data[0]	data[bytenmb-1]	BCC
-----------	-----	------	---------	---------	-------	-----------------	-----

addr: EEPROM user address ($0 \leq addr \leq 84$)

bytenmb: number of bytes to write ($1 \leq bytenmb \leq 16$)

READ/WRITE DEVICE - HOST

0x02	status	BCC
------	--------	-----

Status: 0 ... no error
 -1 ... SERIAL ERROR
 -10 ... EEPROM ERROR

3.7.40 RWD_Set_Prox_Trm_Time

This Proximity Reader command writes new RF-bit-times t_0 , t_1 , t_p into the EEPROM of the read/write device.

ATTENTION: It is not necessary to use this command when working with Philips Proximity Readers because EEPROM is already initialized to following standard values:

$t_0 = 0xA9$	(176 μ s)	Duration of a '0'-bit including t_p
$t_1 = 0x81$	(224 μ s)	Duration of a '1'-bit including t_p
$t_p = 0xEC$	(48 μ s)	Duration of a Modulation Gap

ATTENTION: The values for t_0 , t_1 , t_p do not represent the RF-bit-times in μ s. They have to be computed. If you provide T_0 , T_1 , T_P in μ s you can compute t_0 , t_1 and t_p using following code sequence:

```
/* TP >= 43  $\mu$ s; T0 >= T_P + 40  $\mu$ s; T1 >= T_P + 40  $\mu$ s; */
t_0=(unsigned char)(32768-((T0-T_P-24)/1.2));
t_1=(unsigned char)(32768-((T1-T_P-24)/1.2));
t_p=(unsigned char)(32768-((TP-24)/1.2));
```

C-Function: void RWD_Set_Prox_Trm_Time (BYTE_T t_0 , BYTE_T t_1 , BYTE_T t_p);

Header-File: HitagRWD.H

Serial protocol:

HOST - READ/WRITE DEVICE

0x05	0xA1	t_0	t_1	t_p	BCC
------	------	-------	-------	-------	-----

READ/WRITE DEVICE - HOST

0x02	Status	BCC
------	--------	-----

Status: 0 ... no error
 -1 ... SERIAL ERROR

3.7.41 RWD_SetModuleAdr

The command *RWD_SetModuleAdr* is used to assign a unique node-address to a device whose serial number is known. The new node-address is written into the EEPROM of the read/write device.

Initial value stored in the EEPROM of a delivered read/write device:

addr = 0x00

RWD_SetModuleAdr should be sent in Ordinary Protocol. If the right serial number was sent, the read/write device answers with Ordinary Protocol if its former node-address was zero, otherwise it answers in Extended Protocol.

If the serial number does not match, the command is ignored, and there will be no answer from the reader.

You can read the serial number of the read/write device by using the command *GetVersion*.

C-Function: void RWD_SetModuleAdr (BYTE_T addr, char *snr);

Header-File: HitagRWD.H

Serial protocol:

HOST - READ/WRITE DEVICE

0x0E	0x91	Snr[0]	Snr[10]	addr	BCC
------	------	--------	-------	---------	------	-----

addr: 0x00 ... for communication with a single reader using the Ordinary Protocol.
 >0x00 ... for communication with multiple readers (e.g. in RS485 net)
 using the Extended protocol. Each reader gets a specific address.

READ/WRITE DEVICE - HOST

(only if serial number matches!)

0x02	Status	BCC
------	--------	-----

Status: 0 ... no error
 -1 ... SERIAL ERROR

3.7.42 RWD_Set_HF_Mode

This command sets the read/write device into Proximity or Long Range Mode.

In standard applications (e.g. using standard reader hardware from Philips) *RWD_Set_HF_Mode* is not used because the HITAG Communication Controller automatically sets the right mode after power-up.

Examples:

- *SetHFMode* setting the mode to Long Range sets a Proximity Reader in a powerdown-state with reduced power consumption.
- *SetHFMode* can be used in a system with a Proximity RF-part and a Long Range RF-part to select one part at a time.

C-Function: void RWD_Set_HF_Mode (BYTE_T mode);

Header-File: HitagRWD.H

Serial protocol:

HOST - READ/WRITE DEVICE

0x03	'L'	mode	BCC
------	-----	------	-----

mode: 0x00 ... Proximity Mode
 0x01 ... Long Range Mode

READ/WRITE DEVICE - HOST

0x02	Status	BCC
------	--------	-----

Status: 0 ... no error
 -1 ... SERIAL ERROR

3.7.43 RWD_FFT_Command

This Long Range Reader command starts the FFT (Fast Fourier Transformation) of the DSP (Digital Signal Processor) with the current BitClockDelay (BCD) value in the EEPROM of the read/write device.

This function suppresses up to two harmonic electromagnetic disturbers in the RF Band of the receiver (105 kHz - 145 kHz), e.g. from computers or monitors. Use this function as often as new RF background noise arises near the Long Range antenna.

ATTENTION: As the answer to this command appears before the FFT is ready (duration of FFT is approximately 110 ms), the host program has to wait at least 50 ms until sending the next transponder command.

C-Function: void RWD_FFT_Command (void);

Header-File: HitagRWD.H

Serial protocol:

HOST - READ/WRITE DEVICE

0x02	'F'	0x44
------	-----	------

READ/WRITE DEVICE - HOST

0x02	Status	BCC
------	--------	-----

Status: 0 ... no error
 -1 ... SERIAL ERROR
 -8 ... ACKNOWLEDGEMENT ERROR
 The DSP did not send a correct acknowledge.
 The error leads to a reset of the read/write device.

3.7.44 RWD_Read_BCD

This Long Range Reader command reads the BCD (BitClockDelay) value from the EEPROM of the read/write device. This value adjusts the timing of the read/write device in accordance to the connected antenna.

C-Function: void RWD_Read_BCD (BYTE_T *bitclockdata);

Header-File: HitagRWD.H

Serial protocol:

HOST - READ/WRITE DEVICE

0x02	'f'	BCC
------	-----	-----

READ/WRITE DEVICE - HOST

0x03	Status	bitclockdata	BCC
------	--------	--------------	-----

bitclockdata:

7	6	5	4	3	2	1	0
bitclockdelay Bit 3	bitclockdelay Bit 2	bitclockdelay Bit 1	bitclockdelay Bit 0	0	0	0	0

bitclockdelay: 0 ... 15_{dec} possible

Status: 0 ... no error
 -1 ... SERIAL ERROR

3.7.45 RWD_Set_BCD

This Long Range Reader command effects that a new BCD (BitClockDelay) value is passed to the DSP (Digital Signal Processor) and written into the EEPROM of the read/write device. This value adjusts the timing of the read/write device in accordance to the connected antenna.

A new adjustment may be necessary whenever a new type of antenna is connected to the read/write device.

Bits 4-7 of *-bitclockdata-* represent the BCD value. If Bit 3 of *-bitclockdata-* is set to 0, a FFT (Fast Fourier Transformation) of the DSP with the new BCD value is started in addition.

Standard value stored in the EEPROM:

bitclockdata = 0x90

C-Function: void RWD_Set_BCD (BYTE_T bitclockdata);

Header-File: HitagRWD.H

Serial protocol:

HOST - READ/WRITE DEVICE

0x03	'F'	Bitclockdata	BCC
------	-----	--------------	-----

bitclockdata:

7	6	5	4	3	2	1	0
bitclockdelay Bit 3	bitclockdelay Bit 2	bitclockdelay Bit 1	bitclockdelay Bit 0	mode	0	0	0

bitclockdelay: 0 ... 15_{dec} possible

mode: 0 ... start a FFT

1 ... no FFT

READ/WRITE DEVICE - HOST

0x02	Status	BCC
------	--------	-----

Status: 0 ... no error

-1 ... SERIAL ERROR

-8 ... ACKNOWLEDGEMENT ERROR

The DSP did not send a correct acknowledge.

The error leads to a reset of the read/write device.

3.7.46 RWD_Set_BCD_Offset

This Long Range Reader command writes a new BCD (BitClockDelay) -Offset value into the EEPROM of the read/write device. This value adjusts the difference of the timing between HITAG 1 and HITAG 2 transponders.

ATTENTION: It is not necessary to use this command when working with Philips Long Range Readers because EEPROM is already initialized to following standard value:

```
bcd_offset = 5
```

C-Function: void RWD_Set_BCD_Offset (BYTE_T bcd_offset);

Header-File: HitagRWD.H

Serial protocol:

HOST - READ/WRITE DEVICE

0x03	0xA4	bcd_offset	BCC
------	------	------------	-----

READ/WRITE DEVICE - HOST

0x02	Status	BCC
------	--------	-----

Status: 0 ... no error
 -1 ... SERIAL ERROR

3.7.47 RWD_Read_LR_Status

This Long Range Reader command reads the antenna overload bit. In case of broken or badly detuned antenna the overload bit is high.

C-Function: void RWD_Read_LR_Status (void);

Header-File: HitagRWD.H

Serial protocol:

HOST - READ/WRITE DEVICE

0x02	'r'	0x70
------	-----	------

READ/WRITE DEVICE - HOST

0x02	Status	BCC
------	--------	-----

Status: 0 ... no error
 -1 ... SERIAL ERROR
 -20 ... ANTENNA OVERLOAD

3.7.48 RWD_Set_Power_Down

This command turns the Long Range Reader into Standby Mode.

The byte *-mode-* is set to zero for Standby Mode. To activate the amplifier again this byte must be set to one.

By default the read/write device is in Active Mode.

C-Function: void RWD_Set_Power_Down (BYTE_T mode);

Header-File: HitagRWD.H

Serial protocol:

HOST - READ/WRITE DEVICE

0x03	'D'	mode	BCC
------	-----	------	-----

mode: 0x00 ... Standby Mode
 0x01 ... Active Mode

READ/WRITE DEVICE - HOST

0x02	Status	BCC
------	--------	-----

Status: 0 ... no error
 -1 ... SERIAL ERROR

3.7.49 Vegas_Read_All_Page

Reads one page of all HITAG 1 transponders in the active antenna field.

ATTENTION: This command was developed for a special project requiring a special Reader-Hardware and -Software.

C-Function: void Vegas_Read_All_Page (BYTE_T mode, BYTE_T pagenr, char *data, WORD_T *data_len);

Header-File: HitagVegas.H

CAUTION: The size of the buffer for read data has to be dimensioned big enough by the user. For further information see Header File PROLVEG6.h.

Serial protocol:

HOST - READ/WRITE DEVICE

0x04	0x98	mode	pagenr	BCC
------	------	------	--------	-----

mode: Bit0=0 ... use KEY A for Authentication
 Bit0=1 ... use KEY B for Authentication
 Bit1=0 ... Plain (without Authentication)
 Bit1=1 ... Crypto (with Authentication)
 Bits2-7 must be zero

pagenr: page number

READ/WRITE DEVICE - HOST

0x06	Status	data[0]	data[3]	BCC
------	--------	---------	-------	---------	-----

:
:
:

0x02	Status	BCC
------	--------	-----

An answer string includes one page of each data carrier.

(n+1) strings are transmitted (n ... number of data carriers). The last string contains the last error-condition.

Status: 0 ... no error
 -1 ... SERIAL ERROR
 -3 ... NOTAG
 -7 ... AUTHENTICATION ERROR
 -8 ... ACKNOWLEDGEMENT ERROR
 -9 ... CRYPTOBLOCK NOT INIT

3.7.50 Vegas_Get_Dsp_Version

This command retrieves the version number of the DSP-software.

ATTENTION: This command was developed for a special project requiring a special Reader-Hardware and -Software.

C-Function: void Vegas_Get_Dsp_Version (char *data);

Header-File: HitagVegas.H

Serial protocol:

HOST - READ/WRITE DEVICE

0x02	'v'	BCC
------	-----	-----

READ/WRITE DEVICE - HOST

0x12	Status	data[0]	data[15]	BCC
------	--------	---------	-------	----------	-----

data[0] ... data[15]: Version
 bytes with even index: ASCII
 bytes with odd index: 0

Status: 0 ... no error
 -1 ... SERIAL ERROR

3.7.51 RWD_Key_Init_Mode

To be able to personalize the read/write device it is necessary to enter a special mode, the KeyInit Mode.

The password (is different from Keys or Logdata) ensures that none but authorized persons are able to enter the KeyInitMode.

ATTENTION: After the successful execution of this command (answer sent with Operating Mode BCC calculation) the read/write device enters the KeyInit Mode and BCC calculation changes.

The read/write device changes BCC calculation automatically. On the host system the user is responsible for the new BCC calculation. The C-Library provides the function `proloc_SetBCCMode ()`.

C-Function: `void RWD_Key_Init_Mode (DWORD_T password);`

Header-File: `HitagRWD . H`

Serial protocol:

HOST - READ/WRITE DEVICE

		7	0	15	8	23	16	31	24
0x06	'K'	PW0	PW1	PW2	PW3	BCC			
		/<---				Password		-->/	

READ/WRITE DEVICE - HOST

0x02	Status	BCC
------	--------	-----

Status:

- 0 ... no error
- 1 ... SERIAL ERROR
- 11 ... EEPROM WRONG OLD DATA

The password was incorrect. The read/write device remains in Operating Mode.

3.8 Command Description for KeyInit Mode

The KeyInit Mode is a mode of the reader for using a set of personalization commands as described in the following (See also Chapter „Personalization“).

In this mode the BCC is computed by adding all block data bytes including Block Length. The least significant eight bits are used as BCC.

The command *RWD_Key_Init_Mode* is used to get from Operating Mode to KeyInit Mode. Exit of KeyInit Mode is done by the command *RWD_Reset* or by a failing *KI_Write_EE_Data*, *KI_Write_Secret2* or *KI_Write_Control2*.

3.8.1 KI_Reset

This command switches the read/write device back to the Operating Mode.

ATTENTION:

- After the successful execution of this command (answer with KeyInit Mode BCC calculation) the read/write device enters the Operating Mode and BCC calculation changes.
- In Operating Mode the same command *Reset* (different BCC calculation) has a different functionality.

The read/write device changes BCC calculation automatically. On the host system the user is responsible for the new BCC calculation. The C-Library provides the function `proloc_SetBCCMode ()`.

C-Function: `void KI_Reset (void);`

Header-File: `HitagKeyInit. H`

Serial protocol:

HOST - READ/WRITE DEVICE

0x02	'R'	0x54
------	-----	------

READ/WRITE DEVICE - HOST

0x02	Status	BCC
------	--------	-----

Status: 0 ... no error
 -1 ... SERIAL ERROR

3.8.2 KI_Write_SerNum

Writes a 11 byte serial number into the EEPROM of the read/write device.

ATTENTION: The serial number in Philips Core Module is already fixed and write-protected at delivery.

C-Function: void KI_Write_SerNum (char *snr);

Header-File: HitagKeyInit. H

Serial protocol:

HOST - READ/WRITE DEVICE

0x0D	's'	Snr[0]	Snr[10]	BCC
------	-----	--------	-------	---------	-----

READ/WRITE DEVICE - HOST

0x02	Status	BCC
------	--------	-----

Status: 0 ... no error
 - 1 ... SERIAL ERROR (in this case the error can also mean, that the read/write device is not in the KeyInit Mode!)
 -12 ... EEPROM WRITE PROTECTED

If any error occurs KeyInit Mode is exited immediately.

3.8.3 KI_Read_EE_Data

This command reads personalization data (4 data bytes) from the EEPROM of the read/write device.

Access rights are verified automatically by the read/write device before this command is executed. If a Read command is not permitted, Status is set to -13 (*EEPROM READ PROTECTED*).

C-Function: void KI_Read_EE_Data (BYTE_T num, DWORD_T *data);

Header-File: HitagKeyInit. H

Serial protocol:

HOST - READ/WRITE DEVICE

0x03	'X'	num	BCC
------	-----	-----	-----

num: defines which personalization data is to be read
 0x00 ... Password
 0x01 ... Key A
 0x02 ... Key B
 0x03 ... Logdata 0A
 0x04 ... Logdata 0B
 0x05 ... Logdata 1A
 0x06 ... Logdata 1B

READ/WRITE DEVICE - HOST

		7	0		31	24	
0x06	Status	data[0]	data[3]	BCC		

Status: 0 ... no error
 - 1 ... SERIAL ERROR (in this case the error can also mean, that the read/write device is not in the KeyInit Mode!)
 -13 ... EEPROM READ PROTECTED
 (The read/write device remains in KeyInit Mode.)

3.8.4 KI_Write_EE_Data

This command writes new personalization data (4 data bytes) into the EEPROM of the read/write device.

This command requires the old data to be transmitted as well, which means that data can only be changed if the user knows the old written data.

Access rights and conformity between the sent old data and the stored data are verified before command execution.

C-Function: void KI_Write_EE_Data (BYTE_T num, DWORD_T od,
DWORD_T nd);

Header-File: HitagKeyInit. H

Serial protocol:

HOST - READ/WRITE DEVICE

			7	0	-----	31	24	7	0	-----	31	24	
0x0B	'Y'	num	OD[0]	-----	OD[3]	ND[0]	-----	ND[3]	BCC				

num: defines which personalization data is to be written
 0x00 ... Password
 0x01 ... Key A
 0x02 ... Key B
 0x03 ... Logdata 0A
 0x04 ... Logdata 0B
 0x05 ... Logdata 1A
 0x06 ... Logdata 1B

OD[0] ... OD[3]: Old data
ND[0] ... ND[3]: New data to be written

READ/WRITE DEVICE - HOST

0x02	Status	BCC
------	--------	-----

Status: 0 ... no error
 - 1 ... SERIAL ERROR (in this case the error can also mean, that the read/write device is not in the KeyInit Mode!)
 -11 ... EEPROM WRONG OLD DATA
 -12 ... EEPROM WRITE PROTECTED

If any error occurs KeyInit Mode is exited immediately.

3.8.5 KI_Read_Control

With this command you can read the two control bytes Control_RW and Control_WO from the EEPROM of a read/write device.

C-Function: void KI_Read_Control (BYTE_T *data);

Header-File: HitagKeyInit. H

Serial protocol:

HOST - READ/WRITE DEVICE

0x02	'C'	BCC
------	-----	-----

READ/WRITE DEVICE - HOST

0x04	Status	data[0]	data[1]	BCC
------	--------	---------	---------	-----

data[0]: Control_RW ... see Chapter „Personalization“

data[1]: Control_WO ... see Chapter „Personalization“

Status: 0 ... no error

- 1 ... SERIAL ERROR (in this case the error can also mean, that the read/write device is not in the KeyInit Mode!)

3.8.6 KI_Write_Control

This command writes a new value to the control bytes Control_RW and Control_WO into the EEPROM of the read/write device.

Initial values stored in the EEPROM of a delivered read/write device:

Control_RW = 0x7F

Control_WO = 0xFF

ATTENTION: Once a bit in Control_RW or Control_WO has been set to '0' it is impossible to change it back to one. We strongly recommend to read Chapter „Personalization“ carefully before using this command.

C-Function: void KI_Write_Control (BYTE_T control_rw, BYTE_T control_wo);

Header-File: HitagKeyInit. H

Serial protocol:

HOST - READ/WRITE DEVICE

0x04	'c'	Control_RW	Control_WO	BCC
------	-----	------------	------------	-----

Control_RW: see Chapter „Personalization“

Control_WO: see Chapter „Personalization“

READ/WRITE DEVICE - HOST

0x02	Status	BCC
------	--------	-----

Status:

- 0 ... no error
- 1 ... SERIAL ERROR (in this case the error can also mean, that the read/write device is not in the KeyInit Mode!)

3.8.7 KI_Read_Secret2

This command reads HITAG 2 personalization data (4 data bytes) from the EEPROM of the read/write device.

Access rights are verified automatically by the read/write device before this command is executed. If a Read command is not permitted, Status is set to -13 (*EEPROM READ PROTECTED*).

C-Function: void KI_Read_Secret2 (BYTE_T num, DWORD_T *data);

Header-File: HitagKeyInit. H

Serial protocol:

HOST - READ/WRITE DEVICE

0x03	'V'	num	BCC
------	-----	-----	-----

num: defines which information is to be read
 0x00 ... KEY LOW
 0x01 ... KEY HIGH
 0x02 ... Password TAG
 0x03 ... Password RWD
 0x00 ... KEY LOW
 0x01 ... KEY HIGH

READ/WRITE DEVICE - HOST

		7	0		31	24	
0x06	Status	data[0]	data[3]	BCC		

Status:

- 0 ... no error
- 1 ... SERIAL ERROR (in this case the error can also mean, that the read/write device is not in the KeyInit Mode!)
- 13 ... EEPROM read protected.
(The read/write device remains in KeyInit Mode.)

3.8.8 KI_Write_Secret2

This command writes new HITAG 2 personalization data (4 data bytes) into the EEPROM of the read/write device.

This command requires the old data to be transmitted as well, which means that data can only be changed if the user knows the old written data.

Access rights and conformity between the sent old data and the stored data are verified before command execution.

C-Function: void KI_Write_Secret2 (BYTE_T num, DWORD_T od, DWORD_T nd);

Header-File: HitagKeyInit. H

Serial protocol:

HOST - READ/WRITE DEVICE

			7	0	-----	31	24	7	0	-----	31	24	
0x0B	'W'	num	OD[0]	-----	OD[3]	ND[0]	-----	ND[3]	BCC				

num: defines which information is to be written

0x00 ... KEY LOW

0x01 ... KEY HIGH

0x02 ... Password TAG

0x03 ... Password RWD

OD[0] ... OD[3]: Old data

ND[0] ... ND[3]: New data to be written

READ/WRITE DEVICE - HOST

0x02	Status	BCC
------	--------	-----

Status: 0 ... no error

- 1 ... SERIAL ERROR (in this case the error can also mean, that the read/write device is not in the KeyInit Mode!)

- 11 ... EEPROM WRONG OLD DATA

- 12 ... EEPROM WRITE PROTECTED.

If any error occurs, KeyInit Mode is exited immediately.

3.8.9 KI_Read_Control2

With this command you read the control byte Control_LT from the EEPROM of the read/write device. Control_LT is related to the HITAG 2 transponder.

C-Function: void KI_Read_Control2 (BYTE_T *data);

Header-File: HitagKeyInit. H

Serial protocol:

HOST - READ/WRITE DEVICE

0x02	0x90	BCC
------	------	-----

READ/WRITE DEVICE - HOST

0x03	Status	data[0]	BCC
------	--------	---------	-----

data[0]: Control_LT ... see Chapter „Personalization“

Status: 0 ... no error

- 1 ... SERIAL ERROR (in this case the error can also mean, that the read/write device is not in the KeyInit Mode!)

3.8.10 KI_Write_Control2

This command writes a new value to the control byte Control_LT into the EEPROM of the read/write device. Control_LT is related to the HITAG 2 transponder.

Initial value stored in the EEPROM of a delivered read/write device:

Control_LT = 0xFF

ATTENTION: Once a bit in Control_LT has been set to '0' it is impossible to change it back to one. We strongly recommend to read Chapter „Personalization“ carefully before using this command.

C-Function: void KI_Write_Control2 (BYTE_T control_lt);

Header-File: HitagKeyInit. H

Serial protocol:

HOST - READ/WRITE DEVICE

0x03	0x91	Control_LT	BCC
------	------	------------	-----

Control_LT: see Chapter „Personalization“

READ/WRITE DEVICE - HOST

0x02	Status	BCC
------	--------	-----

Status:

- 0 ... no error
- 1 ... SERIAL ERROR (in this case the error can also mean, that the read/write device is not in the KeyInit Mode!)
- 12 ... EEPROM WRITE PROTECTED

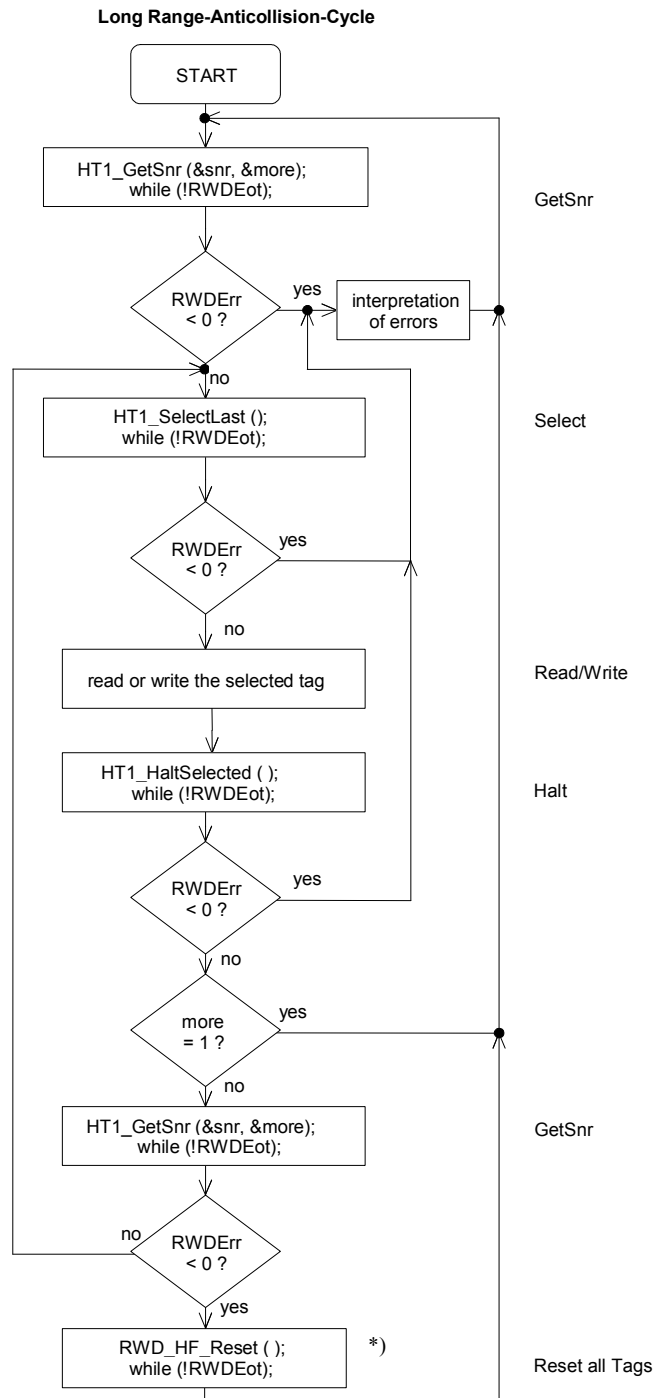
If any error occurs KeyInit Mode is exited immediately.

3.9 Examples to Access HITAG 1/S Transponders

In the following please find examples of read/write cycles both for plain and encrypted access in order to illustrate the command sequence. This sequence does not work with Hitag S transponders configured in TTF and/or Authentication Mode.

3.9.1 Long Range: Anticollision Cycle

In case of several transponders in the reading area of the read/write device the GetSnr command indicates this by the *more* byte. To select one of these transponders for following read or write operations an anticollision cycle must be executed as described in the following flow chart.



*) In case you want to access the same tags for several times.

3.9.2 Proximity/Long Range: READ PLAIN

GetSnr	Reads the serial number of a transponder in the communication field of the antenna. Use C-Function <i>proloc_GetSnr</i> or <i>proloc_GetSnr_Adv</i> .
SelectSnr	Selects (prepares) the transponder for a following read process. Use C-Function <i>proloc_SelectSnr</i> or <i>proloc_SelectLast</i> .
Read (Plain)	Reads a transponder. Use C-Function <i>proloc_ReadPage</i> or <i>proloc_ReadBlock</i> .
HaltSelected	Mutes the just treated transponder.

3.9.3 Proximity/Long Range: WRITE PLAIN

GetSnr	
SelectSnr	
Write (Plain)	Writes data to a transponder. Use C-Function <i>proloc_WritePage</i> or <i>proloc_WriteBlock</i> .
Read (Plain)	To substantially increase the data reliability we strictly recommend to read the previously written data (read after write). Use C-Function <i>proloc_ReadPage</i> or <i>proloc_ReadBlock</i> .
HaltSelected	Mutes the just treated transponder.

3.9.4 Proximity/Long Range: READ CRYPTO

GetSnr

SelectSnr

MutualAuthent Carries out the mutual authentication of the transponder and the read/write device.

Read (Crypto) Use C-Function *proloc_ReadPage* or *proloc_ReadBlock*.

HaltSelected

3.9.5 Proximity/Long Range: WRITE CRYPTO

GetSnr

SelectSnr

MutualAuthent Carries out the mutual authentication of the transponder and the read/write device.

Write (Crypto) Use C-Function *proloc_WritePage* or *proloc_WriteBlock*.

Read (Crypto) To substantially increase the data reliability we strictly recommend to read the previously written data (read after write).
Use C-Function *proloc_ReadPage* or *proloc_ReadBlock*.

HaltSelected

3.10 Examples to Access HITAG 2 Transponders

3.10.1 Proximity/Long Range: READ

HT2_GetSnr Reads the serial number of a transponder in the communication field of the antenna. There is a parameter in the C-Function *proloc_GetSnr_LT* to specify whether you want to access a transponder in Password Mode or in Crypto Mode. If the response of the read/write device includes „no error“ the transponder is selected and ready for read or write accesses.

HT2_ReadPage Reads a transponder.
Use C-Function *proloc_ReadPage_LT*.

HT2_HaltSelected Mutes the just treated transponder.

3.10.2 Proximity/Long Range: WRITE

HT2_GetSnr Reads the serial number of a transponder in the communication field of the antenna. There is a parameter in the C-Function *proloc_GetSnr_LT* to specify whether you want to access a transponder in Password Mode or in Crypto Mode. If the response of the read/write device includes „no error“ the transponder is selected and ready for read or write accesses.

HT2_WritePage Writes to the transponder.
Use C-Function *proloc_WritePage_LT*.

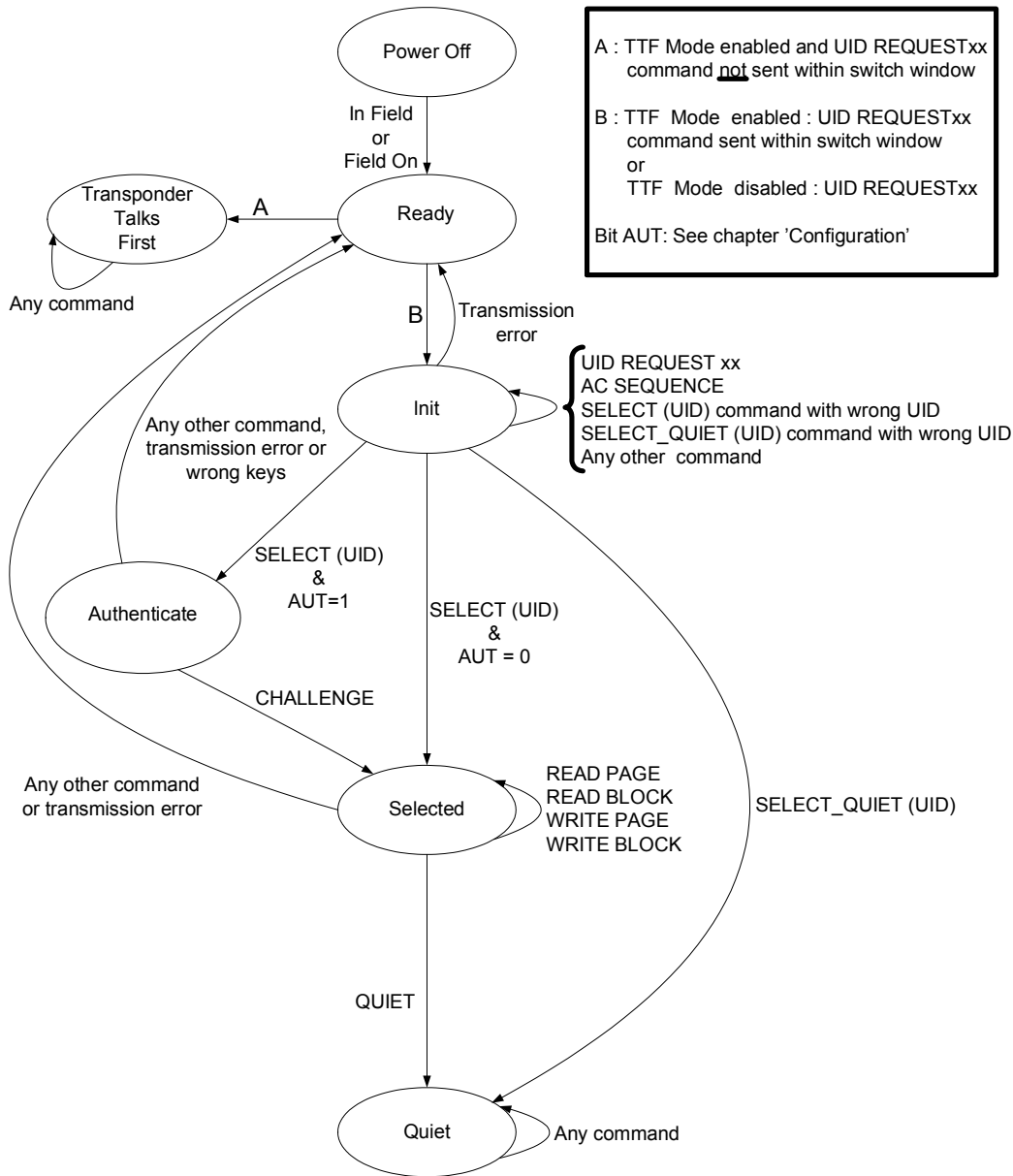
HT2_ReadPage Reads from the transponder to verify if the write process was successful. Use C-Function *proloc_ReadPage_LT*.
If the response of the read/write device (to this first command after *WritePage_LT*) includes an error-condition, start from the beginning of the Write-Sequence again!

HT2_HaltSelected Mutes the just treated transponder.

3.11 Examples to Access HITAG S Transponders

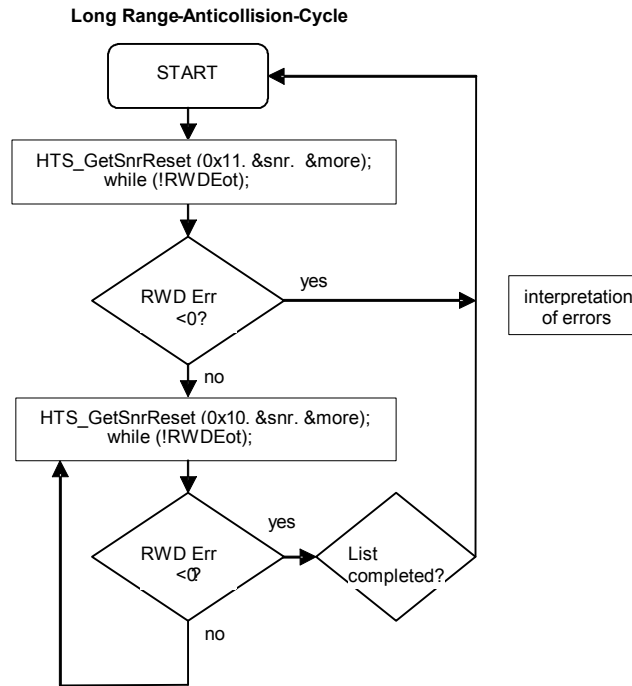
In the following please find the state diagram of an HITAG S Transponder and examples of read/write cycles both for plain and encrypted access in order to illustrate the command sequence on reader side.

3.11.1 Hitag S State Diagram



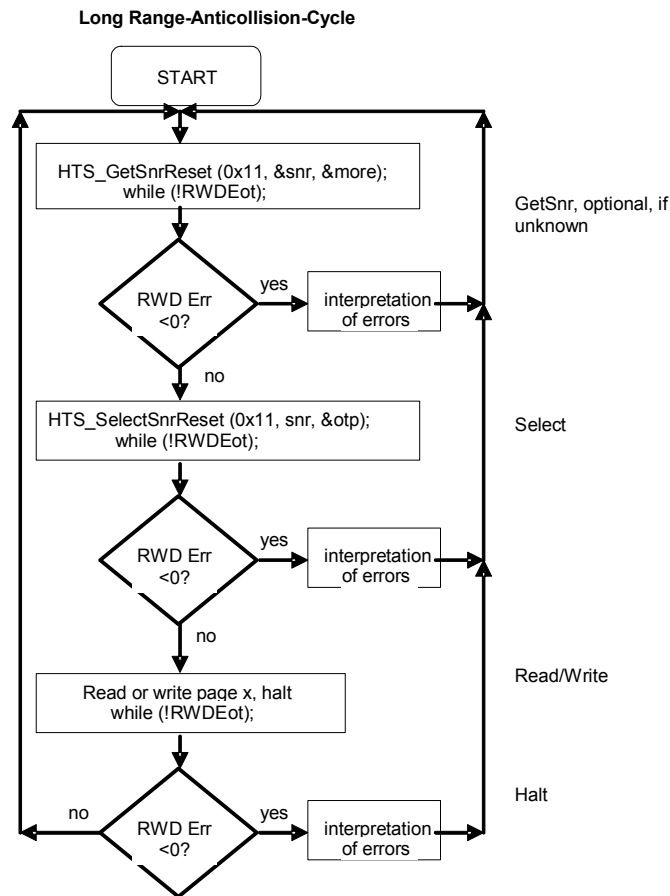
3.11.2 Long Range: Anticollision Cycle

In some applications it is necessary to get all Serial Numbers in the field in the shortest time possible. In this case we recommend to use the command HTS_GetSnrReset. This Command performs one UID Request, a Select and Halt or a Select_Quit and returns a Serial Number and information wether there are more transponder in the field or not (*more*).



3.11.3 Long Range: Anticollision Cycle

In case of several transponders in the reading area of the read/write device the GetSnr command indicates this by the *more* byte. To select one of these transponders for following read or write operations an anticollision cycle must be executed as described in the following flow chart.



*) In case you want to access the same tags for several times.

3.11.4 Proximity/Long Range: READ PLAIN

GetSnr	Reads the serial number of a transponder in the communication field of the antenna. Use C-Function <i>proloc_GetSnr</i> or <i>proloc_GetSnr_Adv</i> .
SelectSnr	Selects (prepares) the transponder for a following read process. Use C-Function <i>proloc_SelectSnr</i> or <i>proloc_SelectLast</i> .
Read (Plain)	Reads a transponder. Use C-Function <i>proloc_ReadPage</i> or <i>proloc_ReadBlock</i> .
HaltSelected	Mutes the just treated transponder.

3.11.5 Proximity/Long Range: WRITE PLAIN

GetSnr	
SelectSnr	
Write (Plain)	Writes data to a transponder. Use C-Function <i>proloc_WritePage</i> or <i>proloc_WriteBlock</i> .
Read (Plain)	To substantially increase the data reliability we strictly recommend to read the previously written data (read after write). Use C-Function <i>proloc_ReadPage</i> or <i>proloc_ReadBlock</i> .
HaltSelected	Mutes the just treated transponder.

3.11.6 Proximity/Long Range: READ CRYPTO

GetSnr

SelectSnr

MutualAuthent Carries out the mutual authentication of the transponder and the read/write device.

Read (Crypto) Use C-Function *proloc_ReadPage* or *proloc_ReadBlock*.

HaltSelected

3.11.7 Proximity/Long Range: WRITE CRYPTO

GetSnr

SelectSnr

MutualAuthent Carries out the mutual authentication of the transponder and the read/write device.

Write (Crypto) Use C-Function *proloc_WritePage* or *proloc_WriteBlock*.

Read (Crypto) To substantially increase the data reliability we strictly recommend to read the previously written data (read after write).
Use C-Function *proloc_ReadPage* or *proloc_ReadBlock*.

HaltSelected

4 Transponders

4.1 HITAG 1 Transponders

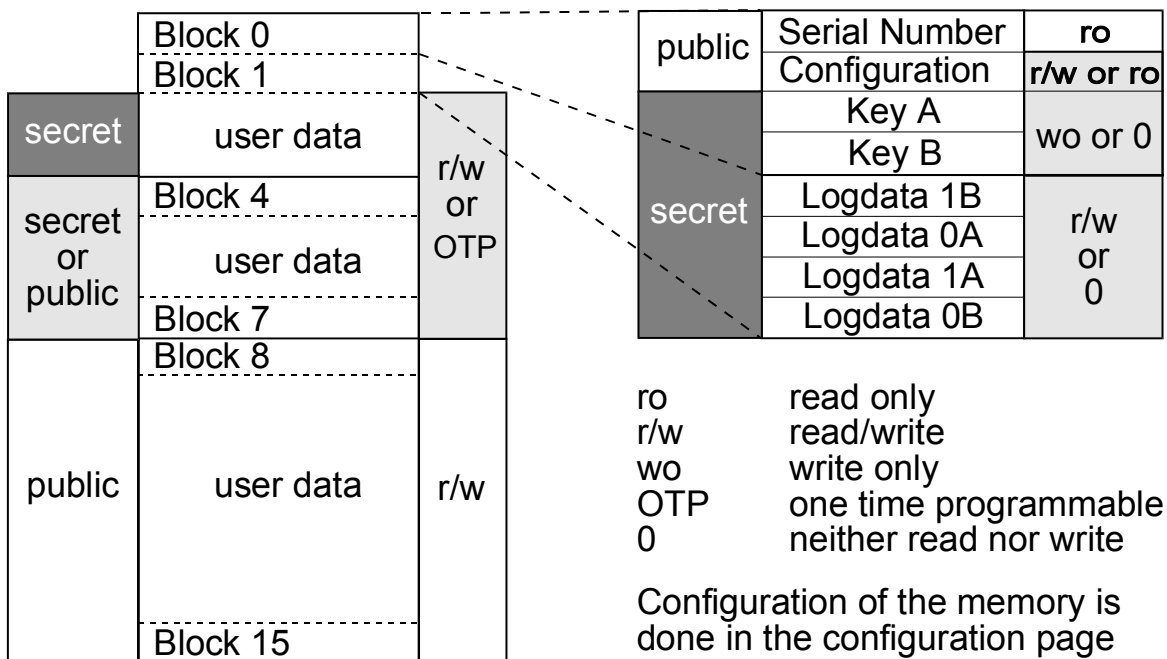
4.1.1 Memory Organization

The 2 kBit EEPROM memory on the transponder is divided into 16 blocks.

Every block consists of 4 pages. A page is the smallest access unit.

Every page consists of 4 bytes (at 8 bits).

Block access is only available for Blocks 2-15, page access is available for Pages 0-63.



Areas (or settings) with light dark background may be configured by the OEM client using the Configuration Page (Page 1).

Memory locations marked with "secret" can only be accessed after a mutual authentication. An enciphered data communication is used in that area.

Memory locations marked with "public" can be accessed without mutual authentication, no encryption is used.

Block 0 includes the unique serial number (programmed during the production process), the Configuration Page (configuration of the memory area) and the keys, Block 1 includes the logdata.

Blocks 4 to 7 can be used either as secret or public areas (configurable), and Blocks 2 to 7 either as read / write or read only areas (configurable). You can also modify keys and logdata and prevent them from being accessed.

Finally the Configuration Page itself can be set to read only.

It is extremely important to be particularly careful when using the Configuration Page (it only can be set to read only once!), keys and logdata as you can lose access to the secret area on the transponder if you make a mistake.

ATTENTION:

Changing of the Configuration Page (Page 1), Keys and Logdata must be done in secure environment. The transponder must not be moved out of the communication field of the antenna during programming! We recommend to put the transponder close to the antenna (zero-distance) and not to remove it during programming.

4.1.2 Anticollision

Anticollision Mode in Long Range applications including HITAG 1/S transponders permits you to process several transponders simultaneously. Theoretically up to 2^{32} transponders can be processed simultaneously. In practice this number is limited, because of the mutual influence of the transponders - they detune each other, if there are too many too close to each other.

In Proximity applications only one transponder is handled even if there are several transponders within the communication field of the antenna. In this case either no communication takes place or the "stronger" or closer transponder takes over.

By muting a selected transponder (HALT Mode) another transponder that is to be found in the communication field of the antenna can be recognized.

4.1.3 Operation-Modes and Configuration

4.1.3.1 Modes of Operation

The HITAG 1 can be operated in following 2 modes that **cannot** be configured using the Configuration Page, but via host-commands.

Standard Protocol Mode:

This mode is activated using the command *GetSnr*.

Advanced Protocol Mode:

This mode is activated using the command *GetSnr_Adv*.

Advanced Protocol Mode is not available for HITAG 1 transponders based on ASIC HT1 ICS30 01x (only available for HITAG 1 transponders based on ASIC HT1 ICS30 02x).

Advanced Protocol Mode uses, above all, an additional Cyclic Redundancy Check (CRC) for read operations.

4.1.3.2 Configuration

The Configuration Page consists of 4 Configuration Bytes, the first two bytes are used for configuration, the other two bytes can be used freely.

The bitmaps in Configuration Bytes 0 and 1 determine the configuration of the memory, i.e. they define which area is secret or public, r/w, ro, wo or neither read nor write.

You can allocate and write the Configuration Page until it is locked (Bit 4 of Configuration Byte 1 is set to '0').

After that these bytes are read only bytes and the configuration of the transponder memory cannot be changed any more.

ATTENTION: Once set to read only the Configuration Page cannot be changed back to r/w again (transponder is hardware protected)!

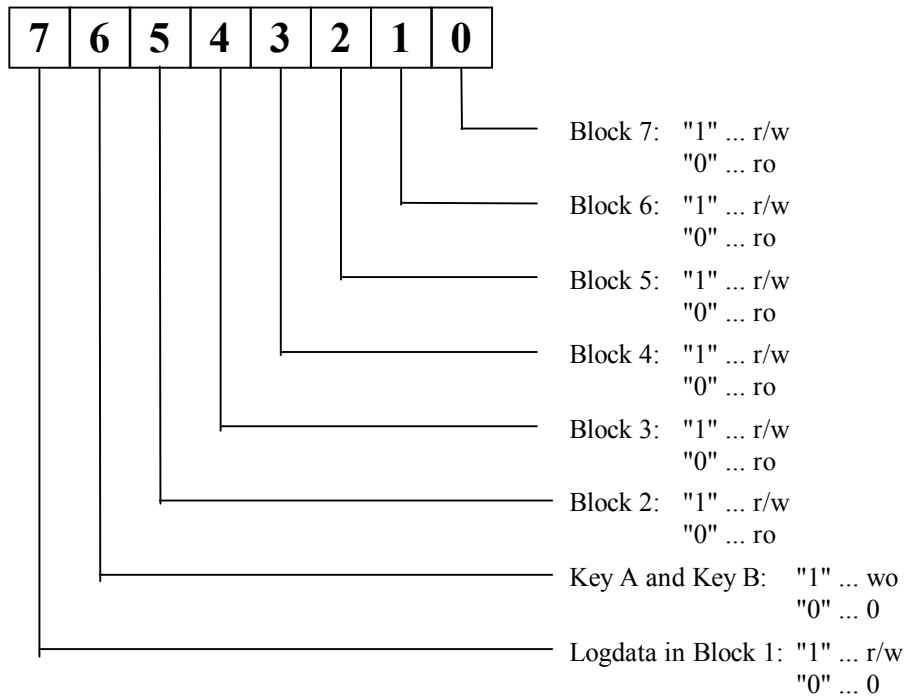
If you change the configuration, you have to place the transponder directly on the antenna or hold it directly to it (0-distance)! In order to avoid any errors do not move the transponder during this write process and be sure that you are in a safe environment without electrical noise.

Configuration Bytes 2 and 3: These two bytes, too, are set to read only by the OEM Lock Bit (Configuration Byte 1 / Bit 4 = "0"). Considering that fact you can use these two bytes freely. They will not affect memory configuration. For example, the OEM client can put his own OEM serial number here.

Explanations of abbreviations used:

r/w	read and write
ro	read only
wo	write only
0	neither read nor write

Configuration Byte 0:



Configuration Byte 0 / Bit 7:

Bit= '1': Logdata can be read and written to.

Bit= '0': Logdata cannot be accessed.

This bit can be set or reset until Bit 4 of Configuration Byte 1 is set to '0'.

For further information on Logdata and Keys see chapter „Personalization“.

Configuration Byte 0 / Bit 6:

Bit= '1': Keys can only be written to.

Bit= '0': Keys cannot be accessed.

This bit can be set or reset until Bit 4 of Configuration Byte 1 is set to '0'.

For further information on Logdata and Keys see chapter „Personalization“.

Configuration Byte 0 / Bits 0 ... 5:

If one of these Configuration Bits reads '1', the corresponding block of the transponder can be read and written.

If the bit is set to '0', the corresponding block can only be read.

Within one block the configuration is always identical, that means either all 4 pages are read/write or all of them are read only.

These bits can be set or reset until Bit 4 of Configuration Byte 1 is set to '0'.

4.1.4 Configuration of Delivered HITAG 1 Transponders

HITAG 1 transponders are delivered with the following configuration by Philips:

Unique Serial Number:

Serial Number:	Read Only	-	fixed
----------------	-----------	---	-------

Configuration Byte 0:

Logdata:	'1' = r/w	-	can be changed
Key A, Key B:	'1' = wo	-	can be changed
Blocks 2 - 7:	'1' = r/w	-	can be changed

Configuration Byte 1:

OEM Lock Bit:	'1' = Configuration Page is r/w	-	can be changed
Blocks 4 - 7:	'1' = public	-	can be changed

Value for Transport Keys, Transport Logdata:

0x00000000

RECOMMENDATION:

Before delivering transponders to end users, the Configuration Page should be set to read only (Configuration Byte 1/Bit 4 = '0').

4.2 HITAG 2 Transponders

4.2.1 Memory Organization

The memory of the transponder consists of 256 bits EEPROM and is organized in 8 pages with 32 bits each.

Depending on the operation-mode the EEPROM is organized differently.

Crypto Mode:

Page	Content
0	Serial Number
1	32 bit "KEY LOW"
2	16 bit "KEY HIGH"
3	8 bit Config., 24 Bit Password TAG
4	read/write page
5	read/write page
6	read/write page
7	read/write page

Password Mode:

Page	Content
0	Serial Number
1	Password RWD
2	reserved
3	8 bit Config., 24 bit Password TAG
4	read/write page
5	read/write page
6	read/write page
7	read/write page

4.2.2 Operation-Modes and Configuration

With the Configuration Byte the operation-mode and the access rights to the memory can be selected. During Power-Up of the transponder the Configuration Byte is read from the transponder's EEPROM.

If you change keys, passwords or configuration, you have to place the transponder directly on the antenna or hold it directly to it (0-distance)! In order to avoid any errors do not move the transponder during this write process and be sure that you are in a safe environment without electrical noise.

4.2.2.1 Modes of Operation

The HITAG 2 can be operated in several modes.

Crypto Mode:

Mode for writing or reading the transponder with encrypted data transmission.

Password Mode:

Mode for writing or reading the transponder with plain data transmission.

Public Mode A (Manchester):

The 64 bits of the user Pages 4 and 5 are cyclically transmitted to the base station.

Public Mode B (Biphase):

Read-only mode according to ISO standards 11784 and 11785 for animal identification.

The 128 bits of the user Pages 4 to 7 are cyclically transmitted to the base station.

See chapter „Communication Reader-Host“ (command description for ReadPublicB_LT) for an example of allocating Pages 4 to 7 for animal identification.

Public Mode C (Biphase):

Read-only mode emulating the read operation of the PCF793X (with a slightly different Program Mode Check).

In the Public Mode C the 128 bits of the user Pages 4 to 7 are cyclically transmitted to the basestation.

4.2.2.2 Configuration

The Configuration Byte is represented by the first 8 bits of Page 3 of the transponder memory.

Configuration Byte:

7	6	5	4	3	2	1	0
0: Manchester Code 1: Biphase Code							
Bit 2		Bit 1		Version		Coding	Coding in HITAG 2-Operation
0		0		Public Mode B		biphase	depending on bit 0
0		1		Public Mode A		manchester	depending on bit 0
1		0		Public Mode C		biphase	depending on bit 0
1		1		HITAG 2		depending on bit 0	depending on bit 0
0: password mode 1: crypto mode							
0: PAGE 6 and 7 read/write 1: PAGE 6 and 7 read only							
0: PAGE 4 and 5 read/write 1: PAGE 4 and 5 read only							
THE SETTING OF THIS BIT IS OTP ! 0: PAGE 3 read/write 1: PAGE 3 read only; Configuration Byte and Password TAG fixed							
THE SETTING OF THIS BIT IS OTP ! 0: PAGE 1 and 2 read/write 1: PAGE 1 no read/no write PAGE 2 read only (when transponder is in password mode) PAGE 2 no read/no write (when transponder is in crypto mode)							

Configuration Byte / Bit 6:

Bit= '0': Page 3 is read/write.

Bit= '1': Page 3 can only be read. This process is irreversible !

ATTENTION: Do not set Bit 6 of the Configuration Byte to '1' before having written the final data into Page 3 (including the Configuration Byte and Password TAG) of the transponder.

Configuration Byte / Bit 7:

Bit= '0': Pages 1 and 2 are read/write.

Bit= '1': Pages 1 and 2 are locked against writing. This process is irreversible !

ATTENTION: Do not set Bit 7 of the Configuration Byte to '1' before having written the final data into Pages 1 and 2 of the transponder.

Standard values for the Configuration Byte:

Password Mode:	0x06
Crypto Mode:	0x0E
Public Mode A:	0x02
Public Mode B:	0x00
Public Mode C:	0x04

4.2.3 Configuration of Delivered HITAG 2 Transponders

HITAG 2 transponders are delivered with the following configuration by Philips:

Unique Serial Number:

Serial Number:	Read Only	-	fixed
----------------	-----------	---	-------

Configuration Byte:

0x06:	Password Mode (Manchester Code)	-	can be changed
	Page 6 and 7 r/w	-	can be changed
	Page 4 and 5 r/w	-	can be changed
	Page 3 r/w	-	can be changed
	Page 1 and 2 r/w	-	can be changed

Values for Transport Passwords, Transport Keys:

Password RWD:	0x4D494B52	(= "MIKR")
Password TAG:	0xAA4854	
Key Low:	0x4D494B52	(= "MIKR")
Key High:	0x4F4E	(= "ON")

RECOMMENDATION:

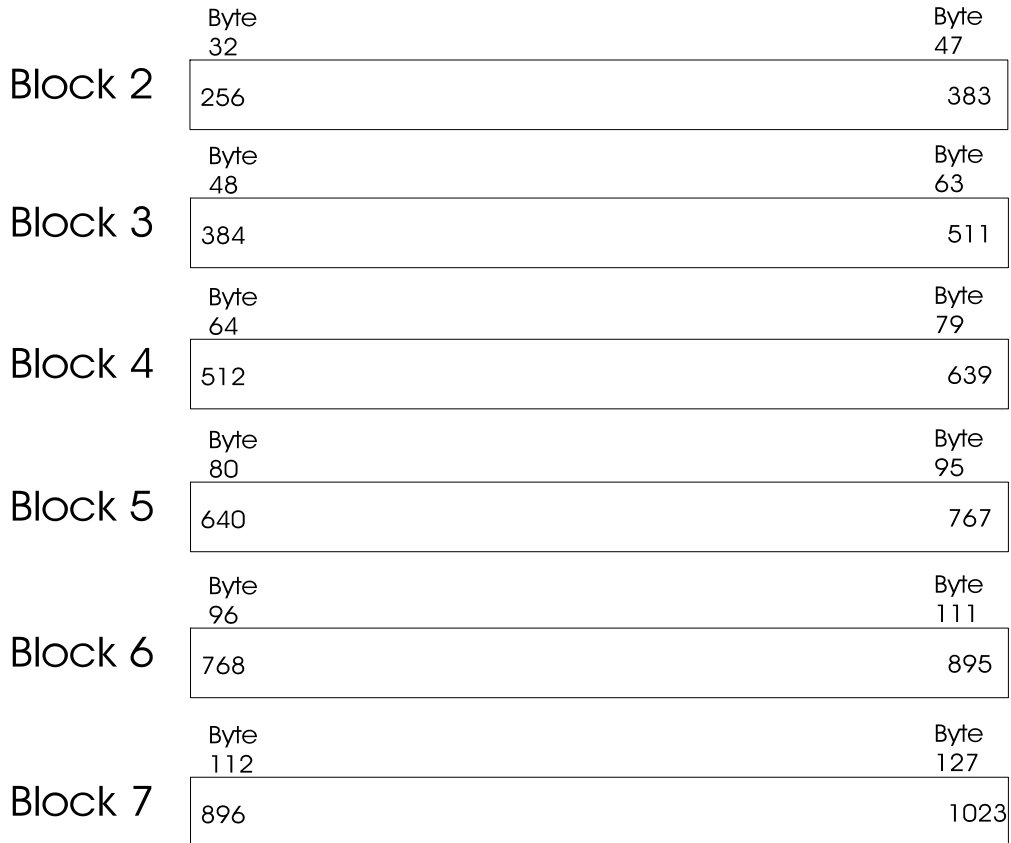
Before delivering transponders to end users, Pages 1 to 3 should be locked (set Configuration Byte / Bit 6 to '1' for Page 3 and set Configuration Byte / Bit 7 to '1' for Pages 1 and 2).

4.3 PIT (PCF793x) Transponders

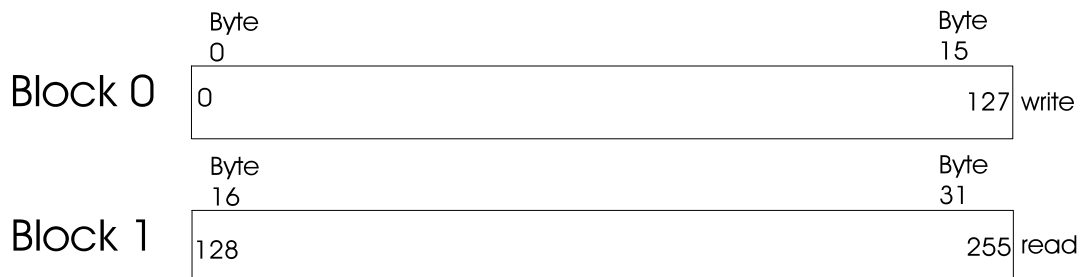
4.3.1 Memory Organization

The EEPROM provides a memory capacity of 128 bytes. It is organized in 8 blocks, each block consisting of 16 bytes. This capacity is split into 6 blocks (=96 bytes) for reading/writing of user data and into 2 blocks (=32 bytes) for the control of the memory access.

The user memory partitioning is shown below.



Blocks 0 and 1 store information for read/write access control. The intention of these blocks is to provide some flexibility for different applications in terms of data security and access to relevant information.



ATTENTION:

PIT transponders can only be accessed using the proximity read/write device !

5 Personalization

5.1 Introduction

In order to profit from the full functionality of the HITAG transponders, the read/write device has to support the transponder's cryptographic feature.

This requires the use of some secret data (keys, logdata). The process of loading these data into the read/write device is called personalization. The same personalization procedure has to be carried out on your transponders.

5.2 Personalization Concept

To enable utmost security and flexibility Philips worked out a personalization concept that shall be shortly described in the following:

The first stage is a test that is done by the producer respectively Philips. Here the **unique serial number** is fixed and defined **Transport Keys, Transport Logdata and Transport Passwords** are pre-programmed.

In the next stage the customers program their own keys and passwords (to ensure that only persons who got the authorisation from the customer are able to access secret data of the transponders) and configure the memory of the transponders. We recommend to lock sensitive areas, that means for example to prevent the possibility to change keys and passwords for the user.

In the last stage the user just reads from and writes to the memory of the transponders.

Note: If you change these Transport Keys and Transport Logdata (and we strictly recommend to do so if you want to store security - sensitive data) in the course of system integration, you have to be extremely careful. Make sure you are in a safe environment while writing secret data to the transponder or the read/write device. This prevents possible listening in to the communication between HOST and read/write device.

All the security relevant data in the read/write device can be protected from read or write accesses using special serial commands.

Security relevant data for HITAG 1 transponders:

- Key information A and B
- Logdata 0A, 0B
- Logdata 1A, 1B

Security relevant data for HITAG 2 transponders:

- Key information
- Password TAG
- Password RWD

The mechanism to protect security relevant data in the read/write device has 3 levels:

Level 0: All security relevant data can be read and written.

Level 1: The data cannot be read any more. If you want to change an entry, you have to know the old value. Otherwise writing access will be denied.

Level 2: The internal data are locked and can neither be read nor written. At this level it is impossible for the user to change the stored data.

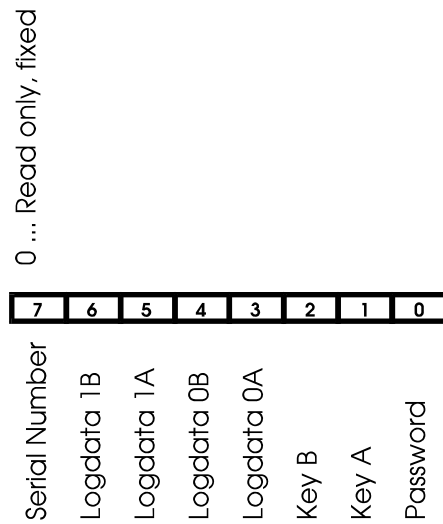
You cannot reset levels, e.g. from level 2 to level 1. Once a security level has been chosen it becomes irreversible.

The functionality of this mechanism is based on the control bytes Control_RW (only for HITAG 1), Control_WO (only for HITAG 1) and Control_LT (only for HITAG 2). All control bytes are located in EEPROM of the read/write device.

On delivery of a read/write device all bits of Control_RW, Control_WO and Control_LT are set to 1, except Bit 7 of CONTROL_RW (serial number).

Control_RW (only HITAG 1):

Control_RW (located in EEPROM of the read/write device) controls read accesses to the following data:



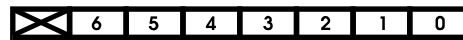
ATTENTION:

You cannot change bits that have once been set to 0 !

BIT NUMBER	BIT NAME	BIT-VALUE = 1	BIT-VALUE = 0
7	Serial Number	-	read allowed
6	Logdata 1B	read allowed	read prohibited
5	Logdata 1A	read allowed	read prohibited
4	Logdata 1A	read allowed	read prohibited
3	Logdata 1A	read allowed	read prohibited
2	Key A	read allowed	read prohibited
1	Key B	read allowed	read prohibited
0	Password	read allowed	read prohibited

Control_WO (only HITAG 1):

Control_WO (located in EEPROM of the read/write device) controls write accesses to the following data:



Logdata 1B	Logdata 1A	Logdata 0B	Logdata 0A	Key B	Key A	Password
------------	------------	------------	------------	-------	-------	----------

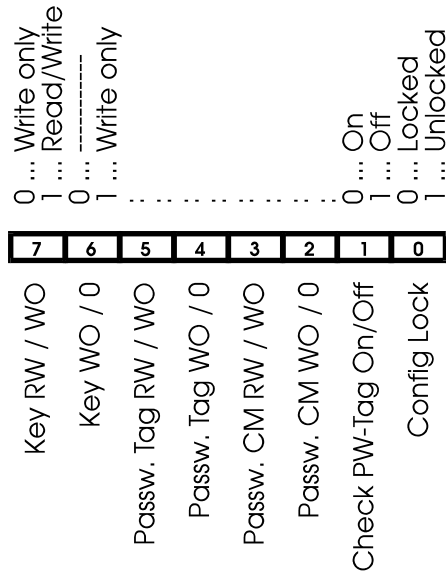
ATTENTION:

You cannot change bits that have once been set to 0 !

BIT NUMBER	BIT NAME	BIT-VALUE = 1	BIT-VALUE = 0
7	-	-	-
6	Logdata1B	write allowed	read/write prohibited
5	Logdata1A	write allowed	read/write prohibited
4	Logdata1A	write allowed	read/write prohibited
3	Logdata1A	write allowed	read/write prohibited
2	Key A	write allowed	read/write prohibited
1	Key B	write allowed	read/write prohibited
0	Password	write allowed	read/write prohibited

Control_LT (only HITAG 2):

Control_LT (located in EEPROM of the read/write device) controls write accesses to the following data:



ATTENTION:

You cannot change bits that have once been set to 0 !

BIT NUMBER	BIT NAME	BIT-VALUE = 1	BIT-VALUE = 0
7	Key RW/WO	read allowed	read prohibited
6	Key WO/0	write allowed	read/write prohibited
5	Passw. TAG RW/WO	read allowed	read prohibited
4	Passw. TAG WO/0	write allowed	read/write prohibited
3	Passw. RWD RW/WO	read allowed	read prohibited
2	Passw. RWD WO/0	write allowed	read/write prohibited
1	Check PW TAG *)	Off	On
0	Config Lock	read/write of Control LT allowed	only read of Control LT allowed

*) Note for Bit 1 of Control_LT:

If the HITAG 2 transponder is in Password or Crypto Mode and a GetSnr_LT command is processed, the incoming Password TAG of the transponder is checked whether it matches with the Password TAG of the reader. If it doesn't, the read/write device transmits the error-message INCORRECT PASSWORD TAG to the host.

We recommend to activate „Check PW TAG“ (set bit 1 to zero) in order to increase the security for GetSnr_LT and PollTags commands.

5.3.2 Changing Keys and Logdata

You do not have to change keys and logdata in order to operate a system with the read/write device because access to the secret area of the transponder is possible with the Transport Keys and Transport Logdata. Nevertheless we strictly recommend to change these data to be sure no other person (and nobody of Philips) than that who got the authorization from you are able to access the secret area of the transponder.

If you change keys and logdata, you have to place the transponder directly on the antenna or hold it directly to it (0-distance)! In order to avoid any errors do not move the transponder during this write process and be sure that you are in a safe environment without electrical noise.

5.3.2.1 Changing Keys

Please, note the order of the steps!

1. Access the transponder (using the Transport Keys).
2. Change one key (e.g.: Key A) on the transponder, i.e., overwrite the corresponding page on the transponder (in this case Page 2) with the new key.
3. Change Key A on the read/write device to the new value.

Caution: On the transponder the key can only be written, which means that you cannot call up the entry! Moreover, you need to know the old value if you want to change the key on the read/write device!

Only after carrying out correctly steps 1 through to 3 (execute a read-access test with the changed key to check it!) may the second key be changed following the steps described above. Conveniently you change both keys to the same value!

5.3.2.2 Incorrect Procedures Changing Keys

- You change both keys on the read/write device and then try to access the transponder. This is not possible because there is no identity between any of the keys on the transponder and the read/write device.
- You change only one key (e.g.: Key A) on the read/write device; the second key (in this example B) remains the Transport Key. Then you try again to access the transponder. This can be possible, only if your system works with both keys and checks one after the other, because one key (here it is Key B) on the transponder and the read/write device is still identical.

The same scenario applies if you first change one or both of the keys on the transponder but leave the keys on the read/write device unchanged (transport keys).

ATTENTION:

If neither Key A nor Key B of the transponder and the read/write device are identical, you cannot access the secret area on that transponder! Access to the plain area of the transponder (e.g. serial number) is possible in any case.

5.3.2.3 Changing Logdata

To change logdata use the same procedure as described for changing keys. Be careful to change them by pairs (on the read/write device and on the transponder):

1. Change, for example, Logdata 0A on the transponder (by overwriting Page 5).
2. Change Logdata 0A on the read/write device to the new value.
3. Change Logdata 1A on the transponder (by overwriting Page 6).
4. Change Logdata 1A on the read/write device to the new value.

Again, you need to know the old values before they can be changed on the read/write device.

For changing the logdata of a big number of tags we recommend to doing it in the same way as described in the former chapter „Changing Keys“ in the note.

When you change a key, this does not mean that you also have to change the corresponding logdata and the other way round.

5.4 Personalization of HITAG 2 Transponders

5.4.1 Definition of Passwords and Keys

Keys are cryptographic codes, which determine data encryption during data transfer between read/write device and transponder. They are used to select a HITAG 2 transponder in Crypto Mode. The 16 bit KEY HIGH and 32 bit KEY LOW form one 48 bit key which has to be identical on both the transponder and the read/write device.

Passwords are needed to select a HITAG 2 transponder in Password Mode. There is one pair of passwords (Password TAG, Password RWD) which has to be identical both on the transponder and the read/write device.

Password TAG: Password that the transponder sends to the read/write device and which may be verified by the latter (depending of the configuration of the read/write device).

Password RWD: Password that the read/write device sends to the transponder and which is checked for identity by the latter.

The passwords and keys are predefined by Philips by means of defined Transport Passwords and a Transport Key. They can be written to, which means that they can be changed.

ATTENTION: Passwords and Keys only can be changed if their current values are known!

It is important that the following values are in accordance with each other, i.e. the respective data on the read/write device and on the transponder have to be identical pairs.

HITAG 2 in Password mode:

on the read/write device		on the transponder
Password RWD	↔	Password RWD

as an option (depending on bit 1 of CONTROL_LT):

Password TAG	↔	Password TAG
--------------	---	--------------

HITAG 2 in Crypto mode:

on the read/write device		on the transponder
KEY LOW	↔	KEY LOW
KEY HIGH	↔	KEY HIGH

as an option (depending on bit 1 of CONTROL_LT):

Password TAG	↔	Password TAG
--------------	---	--------------

5.4.2 Changing Passwords and Keys

You do not have to change passwords and keys in order to operate a system with the read/write device because access to the secret area of the transponder is possible with the Transport Passwords and Transport Keys. Nevertheless, we strictly recommend to change these data to be sure no other person (and nobody of Philips) than that who got the authorization from you are able to access the secret area of the transponder.

If you change passwords and keys, you have to place the transponder directly on the antenna or hold it directly to it (0-distance)! In order to avoid any errors do not move the transponder during this write process and be sure that you are in a safe environment without electrical noise.

5.4.2.1 Changing Password RWD and Keys

Please, note the order of the steps!

1. Select the transponder.
2. If the transponder is in Password Mode, you only have to overwrite Page 1 (Password RWD).
If the transponder is in Crypto Mode, you have to overwrite Page 1 (KEY LOW) and 2 upper bytes of Page 3 (KEY HIGH).
It is recommended to execute a read-access test to check the changed values.
3. Change the corresponding data on the read/write device to the new values.

Caution: You need to know the old value if you want to change the passwords or keys on the read/write device!

5.4.2.2 Incorrect Procedures Changing Password RWD and Keys

- You change values on the read/write device and then try to access the transponder. This is not possible because there is no identity between any of the keys on the transponder and the read/write device.

The same scenario applies if you first change values on the transponder but leave the corresponding values on the read/write device unchanged (transport key).

5.4.2.3 Changing Password TAG

To change Password TAG on HITAG 2 transponders in Password Mode or Crypto Mode use the same procedure as described for changing Password RWD and keys. Be careful to change them by pairs (on the read/write device and on the transponder):

1. Change Password TAG on the transponder. Password TAG and the Configuration Byte are located on Page 3 of the transponder. In order not to change the value of the Configuration Byte it is recommended to read Page 3 from the transponder. Byte 0 is left unchanged, and Bytes 1..3 are set to the new Password TAG value. Then Byte 0 to Byte 3 are written to the transponder.
2. Change Password TAG on the read/write device to the new value.

Again, you need to know the old values before they can be changed on the read/write device.

6 Security Considerations

Developing the read/write device special consideration was given to aspects of security. The following items represent the fundamental framework of the security concept:

- cryptography
- mutual authentication
- password verification and
- Cyclic Redundancy Check (CRC)

6.1 Data Reliability

6.1.1 Data Stream between Read/Write Device and Transponder

HITAG 1 transponders:

All the commands and data transferred from the read/write device to the transponder are secured by Cyclic Redundancy Check (CRC).

Every data stream sent (commands, addresses, user data) from the read/write device to the transponder is checked for data errors by the transponder by means of an integrated 8-bit CRC generator.

The CRC is formed over commands and addresses or the plain data respectively and in the case of Crypto Mode it is also encrypted.

The generator polynomial of the transponder CRC generator reads:

$$u^8 + u^4 + u^3 + u^2 + 1 = 0x1D$$

The CRC preassignment is: 0xFF

HITAG 2 transponders:

Every command sent from the read/write device to the transponder is checked for data errors by the transponder.

Standard commands transferred from the read/write device to the transponder are divided into two Bit Streams. The second Bit Stream is generated by inverting the bits of the first Bit Stream. This redundancy increases data security.

6.1.2 Checking User Data

Security of the data read from the transponder by the read/write device remains with the user for reasons of flexibility.

Therefore, you can choose flexible check sums and store them in the transponder memory together with the data. You can protect sensitive data better than less sensitive data, thus permitting optimized operation times.

Detailed instructions how to use and calculate Cyclic Redundancy Check (CRC) are available at Philips in the following Application Note:

HT1 (resp. HT2) Transponder Family, Reliability and Integrity of Data Transmission.

6.2 Data Privacy

The use of cryptography (Stream Cypher), mutual authentication, and password verification prevents monitoring and copying the data channel. Therefore, the area of the transponder that only can be accessed enciphered is called “secret area“.

To make use of cryptography you need secret data: keys (HITAG 1 and HITAG 2 transponders) and logdata (HITAG 1 transponders).

The transponders and the read/write device are provided with identical transport keys and transport logdata by Philips so that you can start operating them right away.

In order to offer our OEM clients high flexibility, the configuration of the transponder memory, password, keys and logdata can be changed.

We strictly recommend to rigorously restrict these possibilities for the end customers (e.g. for HITAG 1 transponders by setting the Configuration Page to read only, setting password, keys and logdata to neither read nor write).

INTENTIONALLY LEFT BLANK