DATA SHEET

HT CM400
HT RM440 Family
HT RM800 Family

Interface Protocol Reader ↔ Host

Product Specification
Revision 1.1

February 1997

PHILIPS

Table of Contents

Author: Dieter Köckinger

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 CM400 (HITAG Core Module), HT RM440 family (HITAG Proximity Reader Module) and HT RM800 family (HITAG Long Range Reader Module).

For easy and quick development of application specific host software Philips Semiconductors provides a C-Library, Source- and Header-Files. These tools can be found on the floppy disk added to this description.

Following transponders of the 125 kHz family are supported:

- HITAG 1
- HITAG 2
- MIRO / µEM (H400x)
- PIT (PCF793x)

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 Semiconductors' 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

System Requirements to use Philips Semiconductors' C-Library PROLIB6:

- IBM-PC or compatible (minimum 286 processor) with available serial interface
- Borland C-Compiler (Version 3.1 recommended)

1.1 Abbreviations

Please find in the following a list of the abbriviations 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 Semiconductors´ customers using or selling these products for use in such applications do so on their own risk and agree to fully indemnify Philips Semiconductors for any damages resulting from such improper use or sale.

2 Provided Host Software Modules

2.1 Introduction

On the Floppy Disk added to this description you will find the following tools:

Library Files:

PROLIB6C.LIB	Compact Memory Model
PROLIB6H.LIB	Huge Memory Model
PROLIB6L.LIB	Large Memory Model
PROLIB6M.LIB	Medium Memory Model
PROLIB6S.LIB	Small Memory Model
PROLIB6T.LIB	Tiny Memory Model

Depending on the Memory Model you choose for host software you have to include the corresponding Library File in your project.
These Libraries are helpful for developing DOS software in Standard C language with a Borland C-Compiler.

Header Files:

PROLIB6.H	HITAG 1 / MIRO / RWD function declarations
PROLBLT6.H	HITAG 2 function declarations
PROLBPH6.H	PIT function declarations
PROLBMU6.H	Multiple RWD function declarations (Network)
PROLVEG6.H	Function declarations for a special project requiring a special Reader-Hardware and -Software

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.

Source Files:

PROLIB6.C	HITAG 1 / MIRO / RWD functions
PROLBLT6.C	HITAG 2 functions
PROLBPH6.C	PIT functions
PROLVEG6.C	Functions for a special project requiring a special Reader-Hardware and -Software

The code in the Source Files is identical with the code included in the Library Files.
There is no Source File for using multiple RWD’s because all necessary code is located in the Header File PROLBMU6.H.
For developing software on other platforms by using other compilers 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 using interrupts. As a consequence a host-program only has to test a flag (RWDEot) periodically to recognize the end of a communication sequence. Meanwhile the program can execute other functions while the data transfer is running in the background.

To do this, however, some flags are needed:

RWDEot:	Helps to identify the end of a communication sequence flag is set to 0 at a library function request flag is set to 1 at the end of the serial protocol
RWDErr:	Saves the error code 0 ... errorfree execution <0 ... error has occured
RWDDataLen:	Saves the number of bytes received via the serial interface. Can take any value between 0 and 24.

- To open the serial port on the host system use function (Header File PROLIB6.H)
  *proloc_open(char *ComStr) // *ComStr="COM1" for COM1*
  *// *ComStr= "COM2" for COM2*
- To close the serial port on the host system use function (Header File PROLIB6.H)
  proloc_close()
  In order to prevent undesired side effects use *proloc_close* before program end or before a new *proloc_open.*
- To change BCC calculation (when entering or leaving KeyInitMode) use function (Header File PROLIB6.H)
  proloc_SetBCCMode(BYTE_T mode) // mode=0x00: Operating Mode
  // mode=0x01: KeyInit Mode
- For communication in Extended Protocol use commands with ‘Proloc_M’-prefix (Header File PROLBMU6.h)

The names of the commands described in the following have to be prefixed with „proloc_“ to get the correspoding names in the C-Library (e.g. function *proloc_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.
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.

Character Delay ≤ 150 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.

Block Delay ≥ 160 ms

3.5 Command Set

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

Command Bytes used in a Proximity (P) and/or Long Range (L) Reader:

Operating Mode:

COMMAND BYTE		COMMAND NAME	READER	TRANSPONDERS			
				HITAG 1	HITAG 2	MIRO	PIT
'A'	0x41	MutualAuthent		P/L			
'B'	0x42	ReadBlock		P/L			
'D'	0x44	SetPowerDown	L				
'E'	0x45	ReadEEData	P/L				
'F'	0x46	StartFFT / SetBCD	L				
'G'	0x47	GetSnr		P/L			
'H'	0x48	HaltSelected		P/L			
'I'	0x49	ReadInput	P/L				
'K'	0x4B	KeyInitMode	P/L				
'L'	0x4C	SetHFMode	P/L				
'M'	0x4D	ReadMiro			P/L	P/L	
'O'	0x4F	SetOutput	P/L				
'P'	0x50	ReadPage		P/L			
'R'	0x52	Reset	P/L				
'S'	0x53	SelectSnr / SelectLast		P/L			
'T'	0x54	ReadPit					P
'V'	0x56	GetVersion	P/L				
'a'	0x61	TagAuthent		P/L			
'b'	0x62	WriteBlock		P/L			
'c'	0x63	ConfigPorts	P/L				
e'	0x65	WriteEEData	P/L				
'f'	0x66	ReadBCD	L				
'h'	0x68	HFReset	P/L				
'i'	0x69	ReadPorts	P/L				
'l'	0x6C	PollTags		P/L	P/L	P/L	P
'o'	0x6F	WritePorts	P/L				
'p'	0x70	WritePage		P/L			
'r'	0x72	ReadLRStatus	L				
't'	0x74	WritePit					P
'u'	0x75	WritePitBlock					P
	0x80	GetSnr_LT / GetSnrReset_LT			P/L		
	0x81	HaltSelected_LT			P/L		
	0x82	ReadPage_LT			P/L		
	0x83	ReadPageInv_LT			P/L		
	0x84	WritePage_LT			P/L		
	0x85	ReadPit_LT			P		
	0x90	PollKbTags	P/L	P/L	P/L		
	0x91	SetModuleAdr	P/L				
	0x9E	ReadPublicB_LT			P/L		
	0xA1	SetProxTrmTime	P/L				
	0xA2	GetSnr_Adv		P/L			
	0xA4	SetBCDOffset	L				
	0xA6	StopCommand	P/L				

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	ReadControl	P/L
'R'	0x52	Reset (Switch to Operating Mode)	P/L
'V'	0x56	ReadSecret_LT	P/L
'W'	0x57	WriteSecret_LT	P/L
'X'	0x58	ReadEEPROM	P/L
'Y'	0x59	WriteEEPROM	P/L
'c'	0x63	WriteControl	P/L
's'	0x73	WriteSerNum	P/L
'v'	0x76	WritePitSecurity	P
'w'	0x77	WritePitPassword	P
	0x90	ReadControl_LT	P/L
	0x91	WriteControl_LT	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 *ReadPit*).
-15	CRC ERROR	Wrong CRC from a HITAG 1 transponder in Advanced Protocol Mode.
-20	ANTENNA OVERLOAD	Long Range Reader: Broken or badly detuned antenna (error only after command *ReadLRStatus*).

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 GetSnr

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

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

C-Function: void proloc_GetSnr (DWORD_T *snr, BYTE_T *more);
Header-File: PROLIB6.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 GetSnr_Adv

This command provides the serial number of a HITAG 1 transponder and sets the transponder into Advanced Protocol Mode (command is not available for HITAG 1 transponders based on ASIC HT1 ICS30 0**1**x ; only available for HITAG 1 transponders based on ASIC HT1 ICS30 0**2**x).

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

C-Function: void proloc_GetSnr_Adv (DWORD_T *snr, BYTE_T *more);
Header-File: PROLIB6.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 SelectSnr

This command selects the HITAG 1 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 proloc_SelectSnr (DWORD_T snr, DWORD_T *otp);
Header-File: PROLIB6.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
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

3.7.4 SelectLast

Selects the HITAG 1 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 proloc_SelectLast (void);
Header-File: PROLIB6.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

3.7.5 HaltSelected

Puts the selected HITAG 1 transponder into Halt Mode, which means that this transponder remains silent until it leaves and reenters the RF field.

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

C-Function: void proloc_HaltSelected (void);
Header-File: PROLIB6.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

3.7.6 ReadPage

Reads a page (4 bytes) of a selected HITAG 1 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 proloc_ReadPage (BYTE_T crypto, BYTE_T pagenr, char *data);
Header-File: PROLIB6.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

3.7.7 ReadBlock

Reads a block (16 bytes) of the selected HITAG 1 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 proloc_ReadBlock (BYTE_T crypto, BYTE_T pagenr, char *data);
Header-File: PROLIB6.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

3.7.8 WritePage

Writes a page (4 bytes) to the selected HITAG 1 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 proloc_WritePage (BYTE_T crypto, BYTE_T pagenr, char *data);
Header-File: PROLIB6.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

3.7.9 WriteBlock

Writes a block (16 bytes) to the selected HITAG 1 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 proloc_WriteBlock (BYTE_T crypto, BYTE_T pagenr, char *data);
Header-File: PROLIB6.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

3.7.10 TagAuthent

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 proloc_TagAuthent (BYTE_T keyinfo);
Header-File: PROLIB6.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

3.7.11 MutualAuthent

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 proloc_MutualAuthent (BYTE_T keyinfo);
Header-File: PROLIB6.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

3.7.12 GetSnr_LT

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 proloc_GetSnr_LT (BYTE_T mode, DWORD_T *snr, BYTE_T *config);
Header-File: PROLBLT6.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

3.7.13 GetSnrReset_LT

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 proloc_GetSnrReset_LT (BYTE_T mode, DWORD_T *snr, BYTE_T *config);

Header-File: PROLBLT6.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

3.7.14 HaltSelected_LT

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 *HFReset* or putting it out of RF field.

C-Function: void proloc_HaltSelected_LT (void);
Header-File: PROLBLT6.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

3.7.15 ReadPage_LT

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 proloc_ReadPage_LT (BYTE_T pagenr, char *data);
Header-File: PROLBLT6.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

3.7.16 ReadPageInv_LT

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 proloc_ReadPageInv_LT (BYTE_T pagenr, char *data);
Header-File: PROLBLT6.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

3.7.17 WritePage_LT

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 *WritePage_LT* was successful it is important that the immediately following command is a *ReadPage_LT*. If *ReadPage_LT* does not return „no error“ and the right data, you have to repeat *WritePage_LT*.

C-Function: void proloc_WritePage_LT (BYTE_T pagenr, char *data);
Header-File: PROLBLT6.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

3.7.18 ReadPit_LT

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

The read/write device attempts continuously to synchronize on and read a HITAG 2 transponder in Public Mode C. If it succeeds and all checks report positive results, the device sends the 16 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).

C-Function: void proloc_ ReadPit_LT (char *data);
Header-File: PROLBLT6.H

Serial protocol:

HOST - READ/WRITE DEVICE

0x02	0x85	BCC

READ/WRITE DEVICE - HOST

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

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

3.7.19 ReadPublicB_LT

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 proloc_ReadPublicB_LT (BYTE_T *data);
Header-File: PROLBLT6.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

3.7.20 ReadMiro

Sets the read/write device to Permanent Reading Mode for MIRO compatible transponders (e.g. µEM H400x).

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
total	64 bit

C-Function: void proloc_ReadMiro (char *data);
Header-File: PROLIB6.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

3.7.21 ReadPit

This Proximity Reader command sets the read/write device to Permanent Reading Mode for PIT transponders.

The read/write device attempts continuously to synchronize on and read a block of a PIT transponder. If it succeeds and all checks report positive results, the device sends a block of 16 data bytes via the serial interface. After reading the specified number of data blocks, 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).

ATTENTION:
If *blknmb*>1 to read several successive blocks, data must be read by the host-software in a special manner. After having received the first block the host-software must place further *ReadPit* commands for each new incoming block. It is important to place a new *ReadPit* as soon as possible after data from the last *ReadPit* have been received and stored in a data-buffer. Otherwise a data overflow can occur. This can be checked by reading the library-variable RWDErr immediately before a new command *ReadPit (*RWDErr = -14 for PIT DATA OVERFLOW). The further ReadPit commands mentioned above are only necessary, if the Philips Semiconductors' C-Library is used.
For further information see Header File PROLBPH6.h.

C-Function: void proloc_ReadPit (BYTE_T blknmb, BYTE_T next, char *data);
Header-File: PROLBPH6.H

Serial protocol:

HOST - READ/WRITE DEVICE

0x03	'T'	blknmb	BCC

blknmb: number of blocks to read

READ/WRITE DEVICE - HOST

0x12	Status	data[0]	....	data[15]	BCC
:					
:					
0x12	Status	data[0]	....	data[15]	BCC

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

3.7.22 WritePit

This Proximity Reader command sets the read/write device to permanent writing mode for a PIT transponder to write a maximum of 16 data bytes onto it (beginning at a byte-address).

The read/write device attempts continuously to synchronize on and write to a PIT transponder. If it succeeds and all checks report positive results, the device sends a *no error* message 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).

Additional information:
Data can also be written to the transponder using the Password Mode (see also commands *WritePitSecurity* and *WritePitPassword*).

C-Function: void proloc_WritePit (BYTE_T byteaddr, BYTE_T bytenmb, char *data);
Header-File: PROLBPH6.H

Serial protocol:

HOST - READ/WRITE DEVICE

bytenmb+4	't'	byteaddr	bytenmb	data[0]	.....	data[bytenmb-1]	BCC

byteaddr: start address of write data ($0 \leq byteaddr \leq 127$)
bytenmb: number of data bytes to write ($1 \leq bytenmb \leq 16$)

READ/WRITE DEVICE - HOST

0x02	Status	BCC

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

3.7.23 WritePitBlock

This Proximity Reader command sets the read/write device to permanent writing mode for a PIT transponder to write a maximum of 16 data bytes onto it (from beginning of a block-address).

The read/write device attempts continuously to synchronize on and write to a PIT transponder. If it succeeds and all checks report positive results, the device sends a *no error* message 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).

Additional information:
Data can also be written to the transponder using the Password Mode (see also commands *WritePitSecurity* and *WritePitPassword*).

C-Function: void proloc_WritePitBlock (BYTE_T blkaddr, BYTE_T bytenmb, char *data);
Header-File: PROLBPH6.H

Serial protocol:

HOST - READ/WRITE DEVICE

bytenmb+3	'u'	blkaddr	data[0]	.....	data[bytenmb-1]	BCC

blockaddr: start address of write data ($0 \leq$ *blockaddr* ≤ 7)
bytenmb = number of data bytes to write ($1 \leq$ *bytenmb* ≤ 16)

READ/WRITE DEVICE - HOST

0x02	Status	BCC

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

3.7.24 PollTags

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 proloc_PollTags (BYTE_T mode, char *data);
Header-File: PROLIB6.H

Serial protocol:

HOST - READ/WRITE DEVICE

0x03	'l'	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 2 Public Mode C	Poll HITAG 2 Crypto Mode	Poll HITAG 2 Password Mode	Poll PIT	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 PIT was successful:

0x13	Status	0x04	data[0]	-----	data[15]	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 HITAG 2 Public Mode C was successful:

0x13	Status	0x20	data[0]	-----	data[15]	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

3.7.25 PollKbTags

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 Semiconductors' Core Modules).

C-Function: void proloc_PollKBTags (BYTE_T mode, char *data);
Header-File: PROLIB6.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-filed
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

3.7.26 GetVersion

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 proloc_GetVersion (char *data);
Header-File: PROLIB6.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

3.7.27 Reset

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 proloc_Reset (void);
Header-File: PROLIB6.H

Serial protocol:

HOST - READ/WRITE DEVICE

0x02	'R'	0x50

READ/WRITE DEVICE - HOST

0x02	Status	BCC

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

3.7.28 HFReset

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 proloc_HFReset (void);
Header-File: PROLIB6.H

Serial protocol:

HOST - READ/WRITE DEVICE

0x02	'h'	0x6A

READ/WRITE DEVICE - HOST

0x02	Status	BCC

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

3.7.29 StopCommand

The command *StopCommand* 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 *Reset* instead, since *Reset* can cause undesirable side effects (resetting output pins).

C-Function: void proloc_StopCommand (void);
Header-File: PROLIB6.H

Serial protocol:

HOST - READ/WRITE DEVICE

0x02	0xA6	BCC

READ/WRITE DEVICE - HOST

0x02	Status	BCC

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

3.7.30 ReadInput

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

ATTENTION:

- **Pins are internally pulled up!**
- **Using Philips Semiconductors' 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 proloc_ReadInput (BYTE_T *input);
Header-File: PROLIB6.H

Serial protocol:

HOST - READ/WRITE DEVICE

0x02	'I'	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

3.7.31 SetOutput

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

ATTENTION: Using Philips Semiconductors' 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 Semiconductors' 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 Semiconductors' Core Module)

C-Function: void proloc_SetOutput (BYTE_T output);
Header-File: PROLIB6.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

3.7.32 ConfigPorts

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-configurated pins to '1' (5V). The status of output-configurated 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 Semiconductors' 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 proloc_ConfigPorts (BYTE_T config);
Header-File: PROLIB6.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

3.7.33 ReadPorts

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 Semiconductors' Core Modules).

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

C-Function: void proloc_ReadPorts (BYTE_T *input);
Header-File: PROLIB6.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

3.7.34 WritePorts

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 Semiconductors' Core Modules).

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

C-Function: void proloc_WritePorts (BYTE_T mode, BYTE_T output);
Header-File: PROLIB6.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

3.7.35 ReadEEData

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 proloc_ReadEEData (BYTE_T addr, BYTE_T bytenmb, char *data);
Header-File: PROLIB6.H

Serial protocol:

HOST - READ/WRITE DEVICE

0x04	'E'	addr	bytenmb	BCC

addr: EEPROM user address (0 ≤ *addr* ≤ 84)
bytenmb: number of bytes to read (1 ≤ *bytenmb* ≤ 16)

READ/WRITE DEVICE - HOST

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

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

3.7.36 WriteEEData

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 proloc_WriteEEData (BYTE_T addr, BYTE_T bytenmb, char *data);
Header-File: PROLIB6.H

Serial protocol:

HOST - READ/WRITE DEVICE

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

addr: EEPROM user address (0 ≤ *addr* ≤ 84)
bytenmb: number of bytes to write (1 ≤ *bytenmb* ≤ 16)

READ/WRITE DEVICE - HOST

0x02	status	BCC

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

3.7.37 SetProxTrmTime

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 necesarry to use this command when working with Philips **Semiconductors** Proximity Readers because EEPROM is already initialized to following standard values:

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

ATTENTION: The values for t_0, t_1, t_p do not represent the RF-bit-times in µs. They have to be computed. If you provide T0, T1, TP in µs you can compute t_0, t_1 and t_p using following code sequence:

```
/* TP >= 43 µs; T0 >= T_P + 40 µs; T1 >= T_P + 40 µs; */
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 proloc_SetProxTrmTime (BYTE_T t_0, BYTE_T t_1, BYTE_T t_p);
Header-File: PROLIB6.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

3.7.38 SetModuleAdr

The command *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

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 proloc_SetModuleAdr (BYTE_T addr, char *snr);
Header-File: PROLIB6.H

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

3.7.39 SetHFMode

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

In standard applications (e.g. using standard reader hardware from Philips **Semiconductors**) *SetHFMode* 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 proloc_SetHFMode (BYTE_T mode);
Header-File: PROLIB6.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

3.7.40 StartFFT

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 proloc_StartFFT (void);
Header-File: PROLIB6.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.

3.7.41 ReadBCD

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 proloc_ReadBCD (BYTE_T *bitclockdata);
Header-File: PROLIB6.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

3.7.42 SetBCD

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 necesarry 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 startet in addition.

Standard value stored in the EEPROM:
bitclockdata = 0x90

C-Function: void proloc_SetBCD (BYTE_T bitclockdata);
Header-File: PROLIB6.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.

3.7.43 SetBCDOffset

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 necesarry to use this command when working with Philips **Semiconductors** Long Range Readers because EEPROM is already initialized to following standard value:

bcd_offset = 5

C-Function: void proloc_SetBCDOffset (BYTE_T bcd_offset);
Header-File: PROLBLT6.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

3.7.44 ReadLRStatus

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 proloc_ReadLRStatus (void);
Header-File: PROLIB6.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

3.7.45 SetPowerDown

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 proloc_SetPowerDown (BYTE_T mode);
Header-File: PROLIB6.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

3.7.46 ReadAllPage

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 proloc_ReadAllPage (BYTE_T mode, BYTE_T pagenr, char *data, WORD_T *data_len);

Header-File: PROLVEG6.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

3.7.47 GetDspVersion

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 proloc_GetDspVersion (char *data);
Header-File: PROLVEG6.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

3.7.48 KeyInitMode

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 proloc_KeyInitMode (DWORD_T password);
Header-File: PROLIB6.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 *KeyInitMode* is used to get from Operating Mode to KeyInit Mode.
Exit of KeyInit Mode is done by the command *Reset* or by a failing *WriteEEPROM, WriteSecret_LT or WriteControl_LT*.

3.8.1 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 proloc_Reset (void);
Header-File: PROLIB6.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 WriteSerNum

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

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

C-Function: void proloc_WriteSerNum (char *snr);
Header-File: PROLIB6.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 ReadEEPROM

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 proloc_ReadEEPROM (BYTE_T num, DWORD_T *data);
Header-File: PROLIB6.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 WriteEEPROM

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 proloc_WriteEEPROM (BYTE_T num, DWORD_T od,
DWORD_T nd);

Header-File: PROLIB6.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 ReadControl

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 proloc_ReadControl (BYTE_T *data);
Header-File: PROLIB6.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 WriteControl

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 proloc_WriteControl (BYTE_T control_rw, BYTE_T control_wo);
Header-File: PROLIB6.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 ReadSecret_LT

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 proloc_ReadSecret_LT (BYTE_T num, DWORD_T *data);
Header-File: PROLBLT6.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

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 WriteSecret_LT

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 proloc_WriteSecret_LT (BYTE_T num, DWORD_T od, DWORD_T nd);
Header-File: PROLBLT6.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 ReadControl_LT

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 proloc_ ReadControl_LT (BYTE_T *data);
Header-File: PROLBLT6.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 WriteControl_LT

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 proloc_WriteControl_LT (BYTE_T control_lt);
Header-File: PROLBLT6.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.

3.8.11 WritePitSecurity

This command writes a new value for the PIT transponder Password Mode into the EEPROM of the read/write device.

Initial value stored in the EEPROM of a delivered read/write device:
mode = 0x00

If the Password Mode is activated, the 7-byte-PIT-Password (see also command *WritePitPassword*) is sent to the transponder during every write access.

Important: The bit PAC (Password Checkbit) on the transponder must be set (write 0x01 to byte address 0x07 using command *WritePit*).

C-Function: void proloc_WritePitSecurity (BYTE_T mode);
Header-File: PROLBPH6.H

Serial protocol:

HOST - READ/WRITE DEVICE

0x03	'v'	mode	BCC

mode: 0x00 ... Password Mode off
0x01 ... Password Mode on

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!)

3.8.12 WritePitPassword

This command writes a new 7-byte-Password for PIT transponders into the EEPROM of the read/write device.

Important: The bit (Password Checkbit) on the transponder must be set (write 0x01 to byte address 0x07 using command *WritePit*).
See also command *WritePitSecurity*.

C-Function: void proloc_WritePitPassword (char *data);
Header-File: PROLBPH6.H

Serial protocol:

HOST - READ/WRITE DEVICE

0x09	'w'	data[0]	.....	data[6]	BCC

READ/WRITE DEVICE - HOST

0x02	Status	BCC

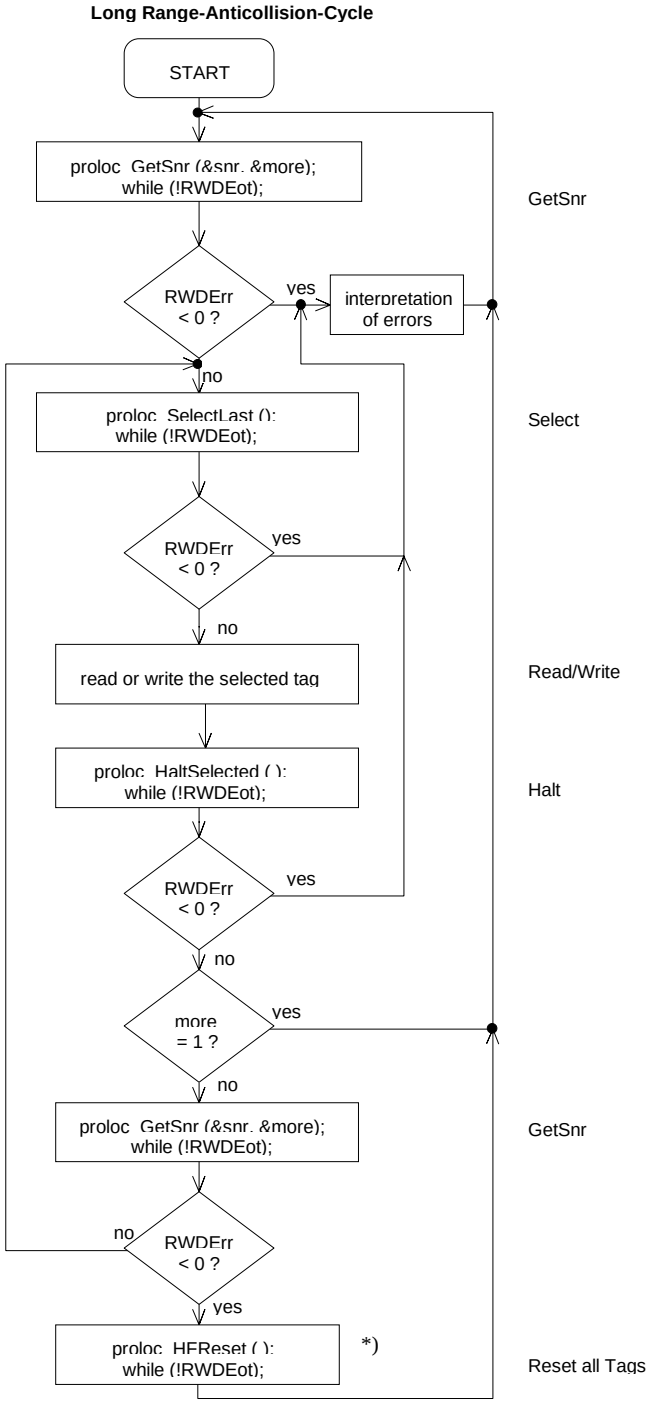
Status: 0 ... no error
- 1 ... SERIAL ERROR (in this case the error can also mean, that the read/write device is not in the KeyInit Mode!)

3.9 Examples to Access HITAG 1 Transponders

In the following please find examples of read/write cycles both for plain and encrypted access in order to illustrate the command sequence.

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 *proloc_GetSnr* or *proloc_GetSnr_Adv*.

SelectSnr Selects (prepares) the transponder for a following read process.
Use C-Function *proloc_SelectSnr* or *proloc_SelectLast*.

Read (Plain) Reads a transponder.
Use C-Function *proloc_ReadPage* or *proloc_ReadBlock*.

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 *proloc_WritePage* or *proloc_WriteBlock*.

Read (Plain) 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 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

GetSnr_LT 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.

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

HaltSelected_LT Mutes the just treated transponder.

3.10.2 Proximity/Long Range: WRITE

GetSnr_LT 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.

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

ReadPage_LT 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!

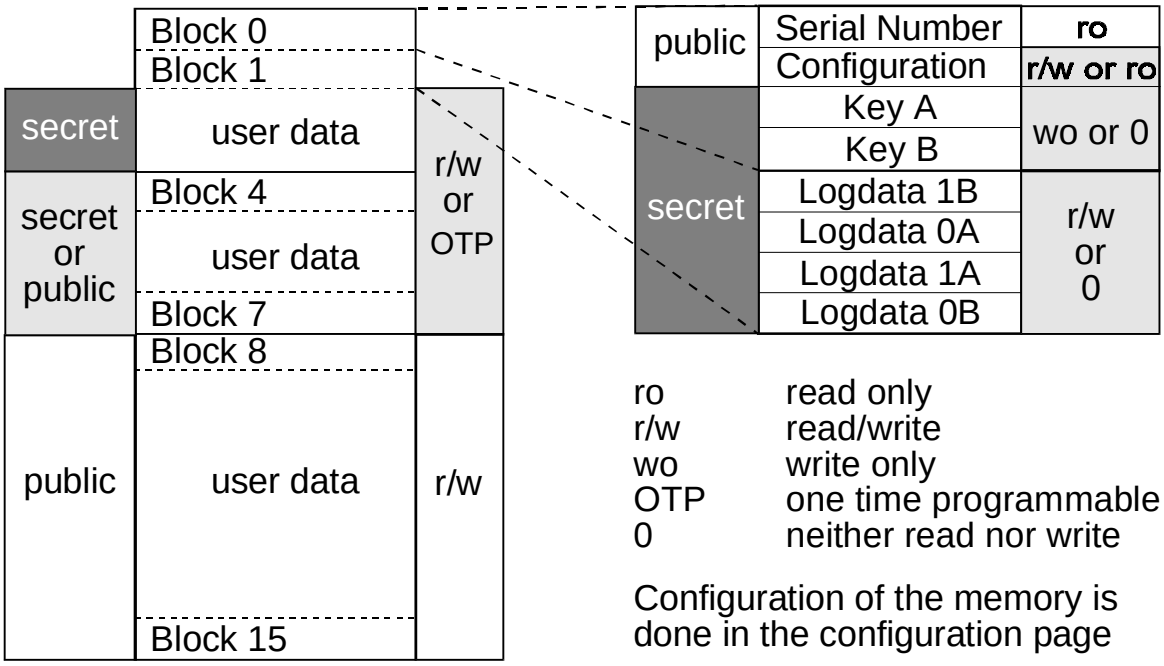
HaltSelected_LT Mutes the just treated transponder.

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 transponders permits you to process several transponders simultaneously. Theoretically up to 2^{32} transponders can be processed simultanously. 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 0**1**x (only available for HITAG 1 transponders based on ASIC HT1 ICS30 0**2**x).
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.

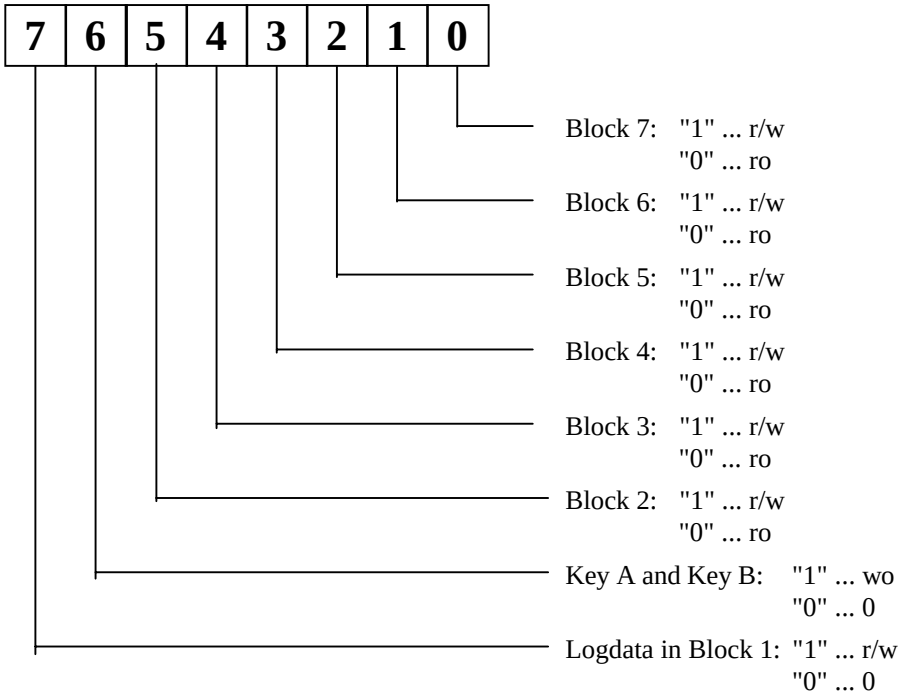
ATTENTION: Once set to read only the Configuration Page cannot be changed back to r/w again (transponder is hardware protected)!
If you change the configuration, you have to place the transponder directly on the antenna or hold it directly to it (0-distance)! In order to avoid any errors do not move 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 Configuraion 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“.

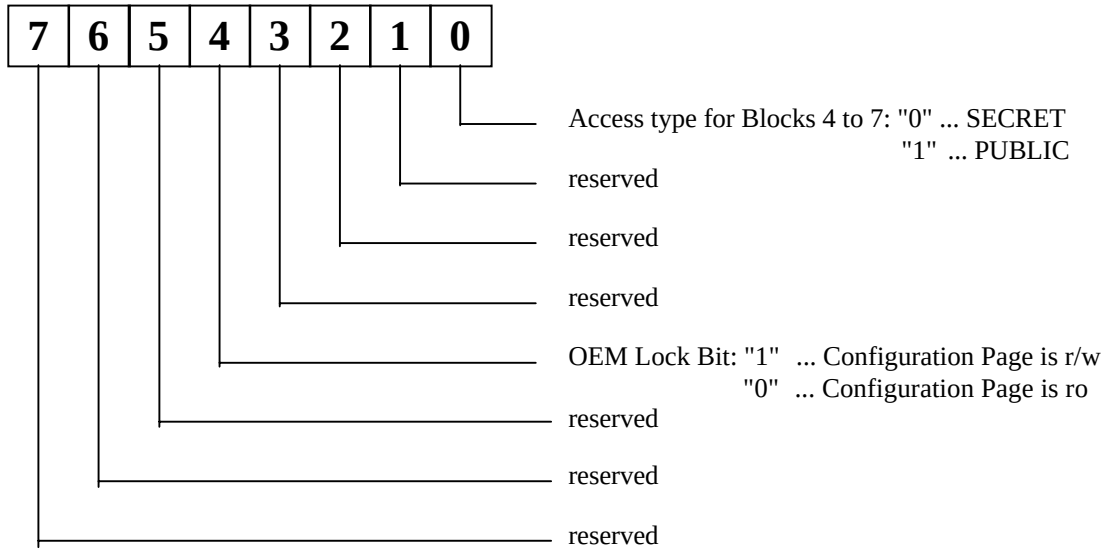
Configuration Byte 0 / Bits 0 ... 5:

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’.

Configuration Byte 1:



Configuration Byte 1 / Bit 5 ... 7:

These three bits are reserved.

ATTENTION: When writing a new value to byte Configuration Byte 1, bit positions marked as „reserved“ must not be altered. To meet that condition read the current Configuration Byte 1 value and mask in your new values for bit positions you are allowed to change.

Configuration Byte 1 / Bit 4:

Bit= ‘1’: Configuration Page can be read and written to.
Bit= ‘0’: Configuration Page can only be read. This process is irreversible !

ATTENTION: Do not set Bit 4 of Configuration Byte 1 to ‘0’ before having written the final data into the Configuration Page of the transponder.

Configuration Byte 1 / Bits 1 ... 3:

These three bits are reserved.

ATTENTION: When writing a new value to byte Configuration Byte 1, bit positions marked as „reserved“ must not be altered. To meet that condition read the current Configuration Byte 1 value and mask in your new values for bit positions you are allowed to change.

Configuration Byte 1 / Bit 0:

Bit= ‘0’: Access type for Blocks 4 to 7 is SECRET.
Bit= ‘1’: Access type for Blocks 4 to 7 is PUBLIC.

This bit 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 **Semiconductors**:

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):

Read-only mode emulating Philips Semiconductors´ MIRO transponders resp. µEM H400x transponders.
The 64 bits of the user Pages 4 and 5 are cyclically transmitted to the base station.

See chapter „Communication Reader-Host" (command description for ReadMiro) for an example of allocating Pages 4 and 5 with MIRO-compatible data.

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 Semiconductors:

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:

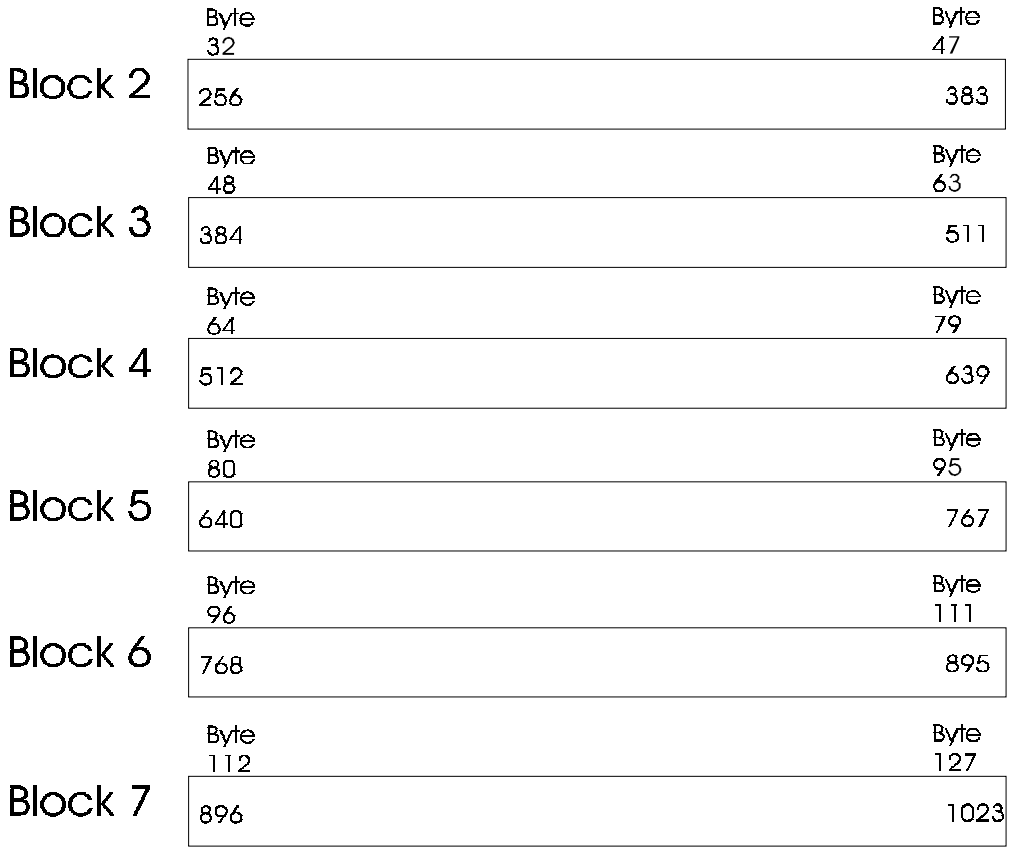
Before delivering transponders to end users, Pages 1 to 3 should be locked (set Configuration Byte / Bit 6 to ‘1’ for Page 3 and set Configuration Byte / Bit 7 to ‘1’ for Pages 1 and 2).

4.3 PIT (PCF793x) Transponders

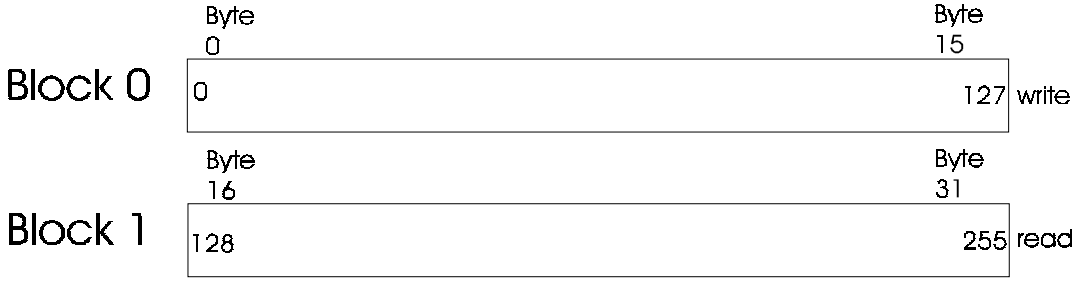
4.3.1 Memory Organization

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 !

4.4 MIRO / µEM (H400x) Transponders

4.4.1 Memory Organization

In the 64 bit memory the unique 40 bit serial numer of the transponder is stored as well as 24 bit header and parity bits. The data is read-only and cannot be changed.

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 Semiconductors 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 Semiconductors. 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 authorization 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.

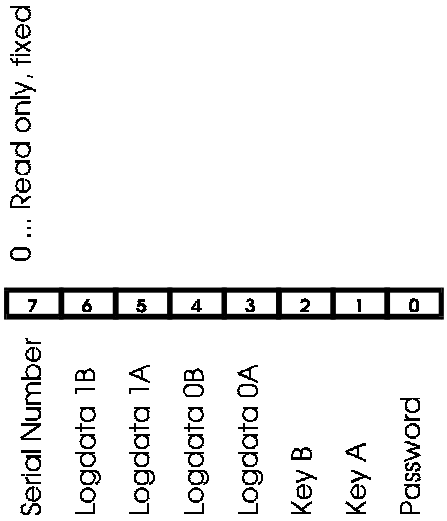
You cannot reset levels, e.g. from level 2 to level 1. Once a security level has been chosen it becomes irreversible.

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:

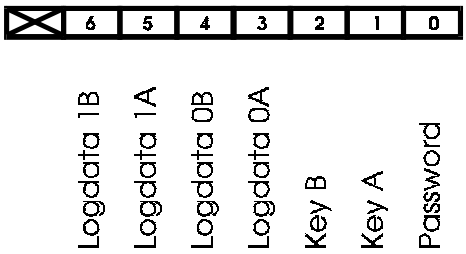


ATTENTION:
You cannot change bits that have once been set to 0 !

BIT NUMBER	BIT NAME	BIT-VALUE = 1	BIT-VALUE = 0
7	Serial Number	-	read allowed
6	Logdata1B	read allowed	read prohibited
5	Logdata1A	read allowed	read prohibited
4	Logdata1A	read allowed	read prohibited
3	Logdata1A	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:

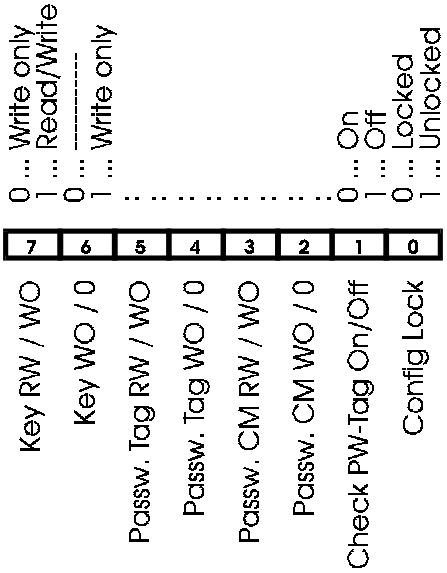


ATTENTION:
You cannot change bits that have once been set to 0 !

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 Personalization of HITAG 1 Transponders

In order to be able to read data from the secret area of a transponder, you have to carry out a procedure called authentication. To do this you need special data (keys).
After transmitting the according command the authentication is automatically carried out by the read/write device.

5.3.1 Definition of Keys and Logdata

Keys are cryptographic codes, which determine data encryption during data transfer between read/write device and transponder.
Two keys (Key A and Key B) which you can use independently of each other, have been installed for security and flexibility reasons. The identity of either Key A or Key B on the read/write device and on the transponder is sufficient.

Logdata represent "passwords" needed to gain access to secret areas on the transponder. A pair of logdata is included with every cryptographic key (Key A and Key B). This logdata pair has to be identical both on the transponder and the read/write device.

ad Key A:	Logdata 0 A	"Password" which the transponder sends to the read/write device and which is verified by the latter.
	Logdata 1 A	"Password" which the read/write device sends to the transponder and which is checked for identity by the latter.
ad Key B:	Logdata 0 B and Logdata 1 B	analogous to Key A

The keys and logdata are predefined by Philips Semiconductors by means of defined Transport Keys (both keys show the same bit map) and Transport Logdata (all logdata show the same bit map). They can be written to, which means that they can be changed.

ATTENTION: Keys and Logdata 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:

on the read/write device		**on the transponder**	
KEY A	⇔	KEY A	⎫
LOGDATA 0A	⇔	LOGDATA 0A	⎬ Set A
LOGDATA 1A	⇔	LOGDATA 1A	⎭
KEY B	⇔	KEY B	⎫
LOGDATA 0B	⇔	LOGDATA 0B	⎬ Set B
LOGDATA 1B	⇔	LOGDATA 1B	⎭

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 Semiconductors) 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 Semiconductors 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 Semiconductors) 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 optimised operation times.

Detailed instructions how to use and calculate Cyclic Redundancy Check (CRC) are available at Philips Semiconductors 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 Semiconductors 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