HF Bluetooth PUK Programming Guide

Contents

1 History

Version	Date	Modifications
1.0	11/08/04	Document Creation
1.1	22/10/04	Added configuration and power save commands, added description of power save features
1.2	28/04/05	Minor Corrections

1 Introduction

This guide is for programmers wishing to communicate with the HF Bluetooth PUK. It defines the packet structure to and from the PUK, the error packets and the commands.

Communication between the HF Bluetooth PUK and the host computer uses Bluetooth. The PUK presents a comm. port as a Bluetooth service. This runs at 57600 baud with 8 data bits, no parity and 1 stop bit

2 Using the PUK

2.1 Turning the PUK on and off

To activate the PUK, press and release the power button. To turn the PUK off press and hold the power button for 3 seconds. The PUK will auto power off after 60 seconds in the absence of a Bluetooth connection.

2.2 Configuring the PUK

The PUK can be configured to automatically perform a set of operations whenever the button is pressed without the need for the host to issue commands. Two configuration bytes are used to determine what the operations are and in addition how the power saving should operate. The command used to program the configuration bytes is shown in section 5.6.

2.3 Button operation

The PUK can be programmed so that it will read transponders when the button is pressed. Any transponders read will be sent to the host in ASCII with a carriage return and line feed terminator. The number of times the PUK will perform this read cycle can also be programmed.

2.4 Charging

To charge the PUK connect the PUK to the power supply. The light on the PUK indicates the charging state red for fast charging, green for charged.

Please note:-

There is a safety feature built into the PUK battery such that if the battery is very flat it will shut down and may require several hours charging to recover to a state where a fast charge is allowed. This should only occur if the PUK has been left for a long period unused. Under normal operation the PUK will shut off before the battery gets to this state. If you are not using the PUK for long periods please fit the transport cap provided.

2.5 Power saving

The PUK has a power save mode which can be turned permanently on or off by command. Alternatively the PUK can be left to automatically enter power save mode. Using the power saving modes will provide a significant increase in operating time for the PUK. The transponder RF section has its own auto power save mode. This is always enabled.

2.5.1 Power Save Mode

When the power save is active there will be a delay of up to 500ms for a command to be processed. In most cases this is not a problem and power save should be permanently enabled to provide maximum battery life. Power saving works by using the standard Bluetooth sniff mode. In this mode the Bluetooth link only communicates every 500ms. Once data has been transferred the link remains active for 200ms. This means that subsequent commands will not be delayed so long as there is not a gap of more than 200ms between subsequent commands.

Please note that the host Bluetooth device must support sniff mode for power saving to work.

2.5.2 Auto Power Save

The auto power save mode offers a compromise between power consumption and speed of response. It works by taking the unit out of power save mode whenever a command is received and then returns to power save mode after about 4 seconds in the absence of further commands. The auto mode works well where there are gaps over 5 seconds between groups of commands.

2.5.3 Transponder RF Section Power Save

The power to the RF section is turned off if there is no communication from the host for 3 Seconds. The RF section is powered up whenever a communication is received from the host.

3 Packet Structure

The packet structure will be the same for both outgoing (command, host to HF PUK) and incoming (response, HF PUK to host). It will consist of the following fields:

- Start of Packet (02_{hex}) (SOP)
- RFU (byte reserved for future use, should be zero, 00_{hex})
- Destination byte (DST)
- Command byte (CMD)
- Options byte (Opt)
- Length (least significant byte) of data section = nl
- Length (most significant byte) of data section = nh
- Parameter section (nh x 256 + nl bytes total)
- Checksum (CS_1, CS_2)

This is illustrated in the diagram below:

Figure 1. General Packet Structure

SOP	RFU	DST	CMD	Opt	Length (LSB)	Length (MSB)	P_0	P_1		P_k	CS_{lsb}	CS_{msb}
02_{hex}	00_{hex}				nl	nh						

Note. k = nh x 256 +nl-1

3.1 Start of Packet

The start of packet allows the HF PUK to identify the start of a packet and is always 02_{hex}.

3.2 RFU

This byte is reserved for future use and should always be 00_{hex}.

3.3 Destination

The destination byte is used to determine where the subsequent command byte is destined.

e.g.

Destination 01hex is the HF PUK itself and the command might be to get the HF PUK version information.

Destination 04hex is an ISO15693 transponder in the HF PUK's reading zone and the command might be to read a page of data.

3.4 Command

The Command byte specifies the action to be performed and must be appropriate for the destination specified.

3.5 Length

The length bytes combine to form the length of the parameter section, least significant byte first. Although this structure allows for a maximum length parameter section of over sixty thousand bytes this will typically be limited by the HF PUK. An invalid length error will be sent if the length specified is greater than the interface can receive.

3.6 Options

The options byte is a set of eight flags that can be set or cleared. The function of each flag is specific to the command and destination byte used. In the response the option byte should only have one of two values; 00_{hex} for a successful transaction or FF_{hex} if an error has occurred. Error Packets and Error Codes are discussed in the next section of this document.

3.7 Parameter Section

The parameter section holds data appropriate to the specific incoming or outgoing packet. The length of this section in bytes is specified by the length section.

3.8 Checksum

The 16-bit checksum (LSB first) is calculated by the addition of all the preceding bytes. If the value exceeds 16 bits the result is truncated to the 16 least significant bits to form the checksum.

4 Error Packet Response

The HF PUK reports errors using the standard packet structure. A value of FF_{hex} in the option byte is used to indicate that an error has occurred. The first parameter byte contains the error code and depending on the error there may be further bytes in the parameter section giving more information about the error. The packet structure is shown below.

SOF	**RFU**	**DST**	**CMD**	**Opt**	**Length (LSB)**	**Length (MSB)**	**P_0**	**P_1**		**P_k**	**CS_{lsb}**	**CS_{msb}**
02_{hex}	00_{hex}			FF_{hex}			Error Code					

Figure 2. Error response packet

Note: The error response may be sent before the full transmission of an outgoing packet; for example an invalid length, greater than can be handled by the HF PUK, will be sent immediately after the second length byte.

4.1 Error Codes

Error Code	Description
01_{hex}	Transponder Generated Error
02_{hex}	Destination Not Recognised
03_{hex}	Command Not Recognised
04_{hex}	Invalid options
05_{hex}	Invalid length
06_{hex}	Invalid Checksum
07_{hex}	No Transponder Present
08_{hex}	Invalid Parameters
09_{hex}	Write not verified
20_{hex}	Write serial number failed
$E0_{hex}$ to EF_{hex}	Bootloader error
FF_{hex}	Undefined error

4.2 Transponder Generated Error (01_{hex})

The HF PUK has successfully received the request and the requested action has been sent to the Transponder. However the Transponder has responded with an error. The rest of the parameter section will contain the error response from the Transponder. For example, this error could be caused by an attempt to write to a locked block.

4.3 Destination Not Recognised (02_{hex})

The HF PUK does not recognise the destination byte received as a destination for which it can generate a suitable response.

4.4 Command Not Recognised (03_{hex})

The HF PUK does not recognise the command byte received as a command for which it can generate a suitable response.

4.5 Invalid Options (04_{hex})

The options specified are not appropriate for the specified command and destination.

4.6 Invalid Length (05_{hex})

Typically the HF PUK has received a start of packet and length bytes but failed to receive sufficient bytes to fulfil the length requirement before a serial timeout occurred. Alternatively the length specified may be too great for the HF PUK to handle.

4.7 Invalid Checksum (06_{hex})

The packet has been received up to one of the checksum bytes at which point the calculated checksum did not match one of the received checksum bytes.

4.8 No Transponder Present (07_{hex})

This error is generated when the HF PUK has failed to find a suitable transponder in the antenna field to perform the requested command on.

4.9 Invalid Parameters (08_{hex})

The Parameters specified are not appropriate for the specified command and destination.

4.10 Write Not Verified (09_{hex})

This error is generated when the HF PUK is unable to verify a write operation. A suitable read command should be used to confirm its success.

4.11 Bootloader Errors ($E0_{hex}$ to EF_{hex})

This range of error codes can only be generated when upgrading the firmware and are handled by the firmware update software.

4.12 Undefined Error (FF_{hex})

This error code represents an unspecified error.

5 HF PUK Commands Destination 01_{hex}

All the commands that control the HF PUK are supported using a command destination byte value of 01_{hex}.

To aid clarity the example packets have the parameter section greyed.

5.1 Option Byte Flags for HF PUK commands

The option byte contains one flag which is used by the HF PUK commands.

Bit 0 is used by the carrier on/off command. If set the HF carrier is turned on and if clear then the HF carrier is turned off.

Bit 1 should be set to 0.

Bits 2 to 7 are reserved for future use and should be set to 0.

Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
0	0	0	0	0	0	0	HF Carrier Flag

5.2 HF PUK commands

Below is a list of all the available Destination 01_{hex} HF PUK commands

Command	Description	Applicable Option Byte flags
01_{hex}	Read Version	None
04_{hex}	Read Serial Number	None
10_{hex}	Carrier on/off	HF Carrier Flag
$F0_{hex}$	Write Serial Number	None
$F1_{hex}$	Reset	None

5.3 Command 01_{hex} (Read Version)

This command is used to get the version number of the firmware and loader program in the HF PUK.

5.3.1 Request packet parameter section

This command does not require any parameters.

Example:

Request Version information from the HF PUK

Request Packet:

02 00 01 01 00 00 00 04 00

5.3.2 Response packet parameter section

Six Byte Response	The first 3 bytes represent the version number of the firmware loaded into the HF PUK LSB first. The next 3 bytes represent the version number of the loader program that downloads new firmware into the HF PUK LSB first.
Three Byte Response	These bytes represent the version number of the loader program that downloads new firmware into the HF PUK LSB first.

A three byte response is only obtained if firmware is not present in the HF PUK or if control has been transferred to the loader program by initialising the HF PUK for firmware downloading.

Example:

Response Packet:

02 00 01 01 00 06 00 00 02 01 01 00 01 0F 00

Firmware Version	1.2.0
Loader Version	1.0.1

5.4 Command 04$_{hex}$ (Read Serial Number)

This command is used to read the 64 byte serial number of the HF PUK.

5.4.1 Request packet parameter section

This command does not require any parameters.

Example:

Request the serial number from the HF PUK

Request Packet:

02 00 01 04 00 00 00 07 00

5.4.2 Response packet parameter section

Serial Number (64 bytes)	The 64 bytes represent the serial number of the HF PUK LSB first in ASCII text.

Example:

Response Packet:

02 00 01 04 00 40 00 41 30 31 32 33 34 42 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 C4 01

The Serial number is “B43210A”

5.5 Command 10_{hex} (Carrier On/Off)

This command is used to turn the RF Carrier on or off. The HF carrier flag in the option byte determines whether the carrier is turned on (flag set) or off (flag clear).

5.5.1 Request packet parameter section

This command does not require any parameters.

Example:

Turn the HF carrier on.

Request Packet:

02 00 01 10 01 00 00 14 00

5.5.2 Response packet parameter section

A single byte is returned in the parameter section with a value of zero if successful.

Example:

Response Packet:

02 00 01 10 00 01 00 00 14 00

Command successful.

5.6 Command B0$_{hex}$ (Program Configuration Bytes)

This command is used to program the configuration bytes. 2 bytes are used on the HF Bluetooth PUK.

5.6.1 Request packet parameter section

The parameter section contains the 2 configuration bytes.

Example:

Program the configuration bytes with 23 0F$_{hex}$

Request Packet:

02 00 01 B0 00 02 00 23 0F E7 00

5.6.2 Response packet parameter section

There are no sections in the response parameter section.

Example:

Response Packet:

02 00 01 B0 00 00 00 B3 00

Command successful.

Note that if the configuration bytes are programmed to make the PUK return transponder UIDs on button press then responses from the PUK containing barcode or transponder data are in ASCII format terminated by Carriage Return Line Feed.

5.6.3 Configuration Byte 1

This byte determines the number of read cycles that are performed after the button is pressed. By setting this byte to zero the PUK will read continually. The operations in a read cycle are defined by the second configuration byte. The read can be cancelled at any time by a second press of the button for 1 second.

5.6.4 Configuration Byte 2

The second configuration byte is a set of eight flags that can be set or cleared. The function of each flag is detailed below:-

Bit 0: Should be set to 0.

Bit 1: Setting this bit allows ISO15693 transponders to be read when the button is pressed.

Bit 2: Setting this bit allows Tagit transponders to be read when the button is pressed.

Bit 3: Setting this bit allows Inside Technologies transponders to be read when the button is pressed.

Bit 4: Should be set to 0.

Bit 5: Setting this bit tells the unit to power up with power save off. Power saving can be turned on later by command, alternatively if the auto flag is set then the unit will use auto power save mode.

Bit 6: Setting this bit tells the unit to use auto power save. This bit will override Bit 5. Setting both bits results will result in auto power save operation rather than power save off operation.

Bit 7: Setting this bit tells the unit to cancel the read cycles immediately a transponder **or** barcode has been successfully read.

Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
Stop when read	Auto power save	Power save at Start up	0	Inside	Tagit	ISO15693	0

5.6.5 Configuration Example

Read 15 times on button press, read Tagit transponders only, stop reading if transponder is found, power up in power save mode and do not use auto power save.

Configuration byte 1 = $0F_{hex}$

Configuration byte 2 = 84_{hex} (10000100_{bin})

Request Packet to program the above configuration:

02 00 01 B0 00 02 00 0F 84 48 01

5.7 Command $B2_{hex}$ (Power Save On)

This command is used to turn the power saving on.

5.7.1 Request packet parameter section

This command does not require any parameters.

Example:

Turn on the power saving.

Request Packet:

02 00 01 B2 00 00 00 B5 00

5.7.2 Response packet parameter section

There are no sections in the response parameter section.

Example:

Response Packet:

02 00 01 B2 00 00 00 B5 00

Command successful.

5.8 Command $B3_{hex}$ (Power Save Off)

This command is used to turn the power saving off.

5.8.1 Request packet parameter section

This command does not require any parameters.

Example:

Turn off the power saving.

Request Packet:

02 00 01 B3 00 00 00 B6 00

5.8.2 Response packet parameter section

There are no sections in the response parameter section.

Example:

Response Packet:

02 00 01 B3 00 00 00 B6 00

Command successful.

5.9 Command $F0_{hex}$ (Write Serial Number)

This command is used to write the 64 byte serial number to the HF PUK. The serial number is a write once option and is usually programmed at manufacture. Attempting to write a serial number a second time will generate an error response.

5.9.1 Request packet parameter section

Serial Number (64 bytes) The 64 bytes represent the serial number of the HF PUK LSB first.

Example:

Write the serial number “B43210A” to the HF PUK

Request Packet:

02 00 01 F0 00 40 00 41 30 31 32 33 34 42 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 B0 02

5.9.2 Response packet parameter section

There are no sections in the response parameter section.

Example:

Response Packet:

02 00 01 F0 00 00 00 F3 00

Write successful

Specific Error Codes

If the write fails error code 20_{HEX} is sent.

5.10 Command $F1_{hex}$ (Reset)

This command is used to reset the HF PUK.

5.10.1 Request packet parameter section

This command does not require any parameters.

Example:

Reset the HF PUK.

Request Packet:

02 00 01 F1 00 00 00 F4 00

5.10.2 Response packet parameter section

A single byte is returned in the parameter section with a value of zero to indicate that the reset is about to occur.

Example:

Response Packet:

02 00 01 F1 00 01 00 00 F5 00

Reset successful.

6 Tagit^{tm} Commands Destination 02_{hex}

Tagit^{tm} transponders are supported on the HF PUK using a command destination byte value of $02_{hex.}$

The reader is expected to be familiar with the Tagit^{tm} transponders, the HF PUK packet structure and the HF PUK error handling.

Full details of the Tagit^{tm} transponder and the Tagit^{tm} protocol are available from Texas Instruments (TI).

To aid clarity the example packets in this document have the parameter section greyed.

6.1 Option Byte Flags for Tagit^{tm} Commands

The option byte contains two flags which are used by Tagit^{tm} Commands.

Bit 0 is used as the address flag. Setting this bit tells the HF PUK to use addressed mode when communicating with Tagit^{tm} transponders. If this bit is set the first four bytes of the parameter section contain the transponder address LSB first.

Bit 1 is used as the Info flag. Setting this bit tells the HF PUK to get extra transponder information.

Bits 2 to 7 are reserved for future use and should be set to 0.

Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
0	0	0	0	0	0	Info Flag	Address flag

6.2 Tagit[tm] Errors

Errors generated by the transponder are reported using error code 01_{hex} with the specific transponder error code as the second parameter byte. Below is a table of the available transponder error codes. The error codes applicable to each command are listed with that command.

10_{hex}	The specified block does not exist.
12_{hex}	The specified block is already locked.
16_{hex}	The specified block was not programmed.
18_{hex}	The specified block was not locked.

Example error packet:

02 00 02 01 FF 02 00 01 10 17 01

The transponder error code 10_{hex} has been generated.

The block specified in the request does not exist.

6.3 Tagit^{tm} commands

Below is a list of all the available Tagit^{tm} commands.

Command	Description	Applicable Option Byte flags
01_{hex}	Get Block	Address Flag
02_{hex}	Get Version	Address Flag
03_{hex}	Put Block	Address Flag
04_{hex}	Put Block Lock	Address Flag
05_{hex}	Lock Block	Address Flag
06_{hex}	SID	Info Flag
07_{hex}	Quiet	Address Flag

6.4 Command 01_{hex} (Get Block)

This command is used to get a block of data from a Tagit[tm] transponder.

6.4.1 Request packet parameter section

Address (0 or 4 bytes).	If the address flag is set the first four bytes of the parameter section contain the transponder address LSB first
Block Number (1 Byte).	This is the number of the block you want to read.

Example:

Request data from block 4 of a transponder whose address is 01234567_{hex}

Request Packet:

02 00 02 01 01 05 00 67 45 23 01 04 DF 00

6.4.2 Response packet parameter section

Block Number (1Byte).	This is the block number read from the transponder.
Lock Status (1 Byte).	The two least significant bits reflect the two lock bits sent by the transponder.
Data (x Bytes).	The actual data read from the block LSB first. The length of this data is dependent on the transponder

Example:

Response Packet:

02 00 02 01 00 06 00 04 01 11 22 33 44 BA 00

Block: 04_{hex}

Lock status: 01_{hex} (user locked)

Data: 44332211_{hex}

6.4.3 Specific Error codes

10_{hex}	Block not available

6.5 Command 02_{hex} (Get Version)

This command is used to get information about the Tagit[tm] transponder.

6.5.1 Request packet parameter section

Address (0 or 4 bytes).	If the address flag is set the first four bytes of the parameter section contain the transponder address LSB first

Example:

Request version information from Tagit[tm] transponder.

Request Packet:

02 00 02 02 00 00 00 06 00

6.5.2 Response packet parameter section

Transponder ID (4Byte).	This is the transponder's address LSB first.
Version number (2 Byte).	This byte is the transponder's version number LSB first.
Manufacturer ID (1 Bytes).	This byte indicates the manufacturer of the transponder.
Number of Blocks (1 Byte)	This gives the number of blocks in the transponder.
Number of Bytes (1 Byte)	This gives the number of bytes per block in the transponder.

Example:

Response Packet:

02 00 02 02 00 09 00 10 32 54 76 05 00 01 08 04 2D 01

Address:	76543210_{hex}
Version:	0005_{hex}
Manufacturer:	01_{hex}
Number of blocks:	08_{hex}
Number of bytes:	04_{hex}

6.5.3 Specific Error codes

-	-

6.6 Command 03_{hex} (Put Block)

This command is used to write a block of data to a Tagit^{tm} transponder.

6.6.1 Request packet parameter section

Address (0 or 4 bytes).	If the address flag is set the first four bytes of the parameter section contain the transponder address LSB first
Block Number (1 Byte).	This is the number of the block you want to write.
Data (x Bytes).	The data to write to the block LSB first. The length of this data is dependent on the transponder

Example:

Write 01234567_{hex} to block 3 of a transponder non addressed

Request Packet:

02 00 02 03 00 05 00 03 67 45 23 01 DF 00

6.6.2 Response packet parameter section

There are no fields in the parameter section.

Example:

Response Packet:

02 00 02 03 00 00 00 07 00

Write successful.

6.6.3 Specific Error codes

10_{hex}	The specified block does not exist.
12_{hex}	The specified block is already locked.
16_{hex}	The specified block was not programmed.

6.7 Command 04_{hex} (Put Block Lock)

This command is used to write a block of data to a Tagit[tm] transponder and then lock the block to prevent further writes.

6.7.1 Request packet parameter section

Address (0 or 4 bytes).	If the address flag is set the first four bytes of the parameter section contain the transponder address LSB first
Block Number (1 Byte).	This is the number of the block you want to write and lock.
Data (x Bytes).	The data to write to the block LSB first. The length of this data is dependent on the transponder

Example:

Write and lock 01234567_{hex} to block 3 of a transponder non addressed

Request Packet:

02 00 02 04 00 05 00 03 67 45 23 01 E0 00

6.7.2 Response packet parameter section

There are no fields in the parameter section.

Example:

Response Packet:

02 00 02 04 00 00 00 08 00

Write and lock successful.

6.7.3 Specific Error codes

10_{hex}	The specified block does not exist.
12_{hex}	The specified block is already locked.
16_{hex}	The specified block was not programmed.
18_{hex}	The specified block was not locked.

6.8 Command 05_{hex} (Lock Block)

This command is used to lock a block of data in a Tagit[tm] transponder.

6.8.1 Request packet parameter section

Address (0 or 4 bytes).	If the address flag is set the first four bytes of the parameter section contain the transponder address LSB first
Block Number (1 Byte).	This is the number of the block you want to lock.

Example:

Lock block 3 of a transponder non addressed

Request Packet:

02 00 02 05 00 01 00 03 0D 00

6.8.2 Response packet parameter section

There are no fields in the parameter section.

Example:

Response Packet:

02 00 02 05 00 00 00 09 00

Lock successful.

6.8.3 Specific Error codes

10_{hex}	The specified block does not exist.
12_{hex}	The specified block is already locked.
18_{hex}	The specified block was not locked.

6.9 Command 06_{hex} (SID Poll)

This command is used to identify the transponders within range of the reader. If the info flag is set the transponders will include their version information in their response.

The transponders respond to the SID poll in one of 16 time slots. The slot they use is dependent on their address and the mask value. Only transponders whose addresses match the mask will respond. If more than one transponder responds in a time slot then a collision is generated.

If a non-zero mask length is used then the mask bits are used to determine which transponders respond. The mask is applied from least significant bit of the UID up to the mask length. Only transponders with UIDs that match the bits of the mask respond. The next four most significant bits above the mask are used to determine which timeslot the transponder responds in.

For example, if the mask length is zero then all transponders whose 4 LSBits are 0000_{bin} will respond in the first time slot and transponders whose 4 LSBits are 0001_{bin} will respond in the next.

If the mask length is 4 and the mask bits are 0000_{bin} then the transponders whose 8 LSBits are 00000000_{bin} will respond in the first time slot and transponders whose 8 LSBits are 00010000_{bin} will respond in the next.

6.9.1 Request packet parameter section

Mask length (1 Byte).	This is the number of bits in the following mask.
Mask	Initial match conditions for the SID address comparison by the transponder (LSB first). The MSByte of the mask should be padded with zeros if there are insufficient bits to fill the byte.

Example:

Perform an SID poll with a mask value of 0110_{bin}

(only those transponders whose 4 LSBits of their address match 0110_{bin} will respond to the SID poll)

Request Packet:

02 00 02 06 00 02 00 04 06 16 00

6.9.2 Response packet parameter section

The parameter section contains 16 sub-sections, one for each time slot. Each sub-section starts with a status byte that indicates the contents of the sub-section. There are three possible values for the status byte:-

00_{hex} The sub-section is empty because no transponder was detected.

01_{hex} The sub-section is empty because a collision was detected.

02_{hex} The sub-section contains transponder information. It the info flag is not set then only the 4 byte transponder address is sent (LSB first). If the info flag is set then the full 9 byte version response is sent (see 6.5.2).

Example:

Response Packet: (info flag not set)

02 00 02 06 00 14 00 00 00 02 26 32 54 76 00 00 01 00 00 00 00 00 00 00 00 00 00 43 01

Time Slot	*Response*
0, 1, 3, 4, 6 -15	No transponder detected
2	Transponder address 76543226_{hex} detected.
5	Collision detected

The next step in the SID process is to resolve the collisions. To achieve this in the example above a mask of 01010110_{bin} would be used on the next SID poll.

Example:

Request Packet:

02 00 02 06 00 02 00 08 56 6A 00

Response Packet:

02 00 02 06 00 18 00 00 00 02 56 32 54 76 02 56 33 54 76 00 00 00 00 00 00 00 00 00 00 00 00 CB 02

Time Slot	*Response*
0, 1, 4 -15	No transponder detected
2	Transponder address 76543256_{hex} detected.
3	Transponder address 76543356_{hex} detected.

6.9.3 Specific Error codes

-	-

6.10 Command 07_{hex} (Quiet)

This command is used to prevent a transponder from responding to SID polls.

The quiet command will work in both addressed and non addressed modes but using it in non addressed mode will turn all transponders in the field off.

6.10.1 Request packet parameter section

Address (0 or 4 bytes).	If the address flag is set the first four bytes of the parameter section contain the transponder address LSB first

Example:

Quiet the transponder whose address is $00C3213E_{hex}$

Request Packet:

02 00 02 07 00 04 00 3E 21 C3 00 31 01

6.10.2 Response packet parameter section

The transponder does not respond to this command so it is normal to get a transponder not present error.

Example:

Response Packet:

02 00 02 07 FF 01 00 07 12 01

Transponder not present error

6.10.3 Specific Error codes

-	-

7 ISO15693 Commands Destination 04_{hex}

ISO15693 transponders are supported on the HF PUK using a command destination byte value of $04_{hex.}$

The reader is expected to be familiar with the ISO15693 transponders, the HF PUK packet structure and the HF PUK error handling. This document should be read in conjunction with the current ISO/IEC15693 part 3 specification

To aid clarity the example packets in this document have the parameter section greyed.

7.1 Option Byte Flags for ISO15693 Commands

The option byte contains eight flags which are used by ISO15693 Commands.

Bit 0 is used as the Option flag setting this bit controls how the transponder responds to commands. The HF PUK is unable to verify operations that write to the transponder if the option flag is clear. Writes with the option flag clear will generate error code $09_{HEX.}$ It is then the responsibility of the host to confirm the write operation with a suitable read request.

Bit 1 is used as the transponder data rate flag. Setting this bit tells the transponder to send its data using the fast data rate.

Bit 2 is used as the transponder modulation flag. Setting this bit tells the transponder to send its data using one sub-carrier (AM).

Bit 3 is used as the HF PUK modulation flag. Setting this bit tells the HF PUK to send its data using 100% modulation. This bit must be set on the HF PUK, the hardware does not support 10% modulation.

Bit 4 is used as the HF PUK modulation encoding flag. Setting this bit tells the HF PUK to send its data using 1 out of 256 encoding.

Bit 5 is used as the AFI flag. Setting this bit tells the HF PUK to use AFI mode when communicating with ISO15693 transponders. If this bit is set the first byte of the parameter section contain the AFI value.

Bit 6 is used as the address flag. Setting this bit tells the HF PUK to use addressed mode when communicating with ISO15693 transponders. If this bit is set the first eight bytes of the parameter section contain the transponder address (UID) LSB first.

Bit 7 is used as the select flag. Setting this bit tells the HF PUK to use select mode when communicating with ISO15693 transponders. If this bit is set only the transponder in the selected state will respond.

Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
Select Flag	Address Flag	AFI Flag	1:4 / 1:256 Modulation	1	FM / AM Modulation	Slow / Fast Data rate	Option Flag

Note: Bits 6 and 7 are mutually exclusive; setting both these bits will generate an error response.

7.2 ISO 15693 Errors

Errors generated by the transponder are reported using error code 01_{hex} with the specific transponder error code as the second parameter byte. Below is a table of the available transponder error codes.

01_{hex}	The command is not supported.
02_{hex}	The command is not recognised.
03_{hex}	The command option is not supported.
$0F_{hex}$	Unspecified error.
10_{hex}	The specified block is not available.
11_{hex}	The specified block is already locked.
12_{hex}	The specified block is locked and cannot be changed.
13_{hex}	The specified block was not successfully programmed.
14_{hex}	The specified block was not successfully locked.

Example error packet:

02 00 04 20 FF 02 00 01 10 38 01

The transponder error code 10_{hex} has been generated.

The block specified in the request is not available.

7.3 ISO15693 Commands

Below is a list of the available ISO15693 commands and applicable option byte flags.

Command Description	Command Value	Option Byte Flags							
		Bit 7 Select	**Bit 6 Address**	**Bit 5 AFI**	**Bit 4 1:4 / 1:256**	**Bit 3 10% / 100%**	**Bit 2 FM / AM**	**Bit 1 Slow / Fast**	**Bit 0 Option**
Inventory	0x01	0	0	U	U	1	U	U	0
Stay Quiet	0x02	0	1	0	U	1	U	U	0
Read Single Block	0x20	U	U	0	U	1	U	U	U
Write Single Block	0x21	U	U	0	U	1	U	U	U*
Lock Block	0x22	U	U	0	U	1	U	U	U*
Read Multiple Blocks	0x23	U	U	0	U	1	U	U	U
Write Multiple Blocks	0x24	U	U	0	U	1	U	U	U*
Select	0x25	0	1	0	U	1	U	U	0
Reset to Ready	0x26	U	U	0	U	1	U	U	0
Write AFI	0x27	U	U	0	U	1	U	U	U*
Lock AFI	0x28	U	U	0	U	1	U	U	U*
Write DSFID	0x29	U	U	0	U	1	U	U	U*
Lock DSFID	0x2A	U	U	0	U	1	U	U	U*
Get System Information	0x2B	U	U	0	U	1	U	U	0
Get Multiple Block Status	0x2C	U	U	0	U	1	U	U	0

U = User selectable value
0 = The HF PUK will set this value to 0
1 = The HF PUK will set this value to 1
Bits 6 and 7 are mutually exclusive setting both of these bits will generate an error

* The HF PUK is unable to verify operations that write to the transponder if the option flag is clear. Writes with the option flag clear will generate error code $09_{HEX.}$ It is then the responsibility of the host to confirm the write operation with a suitable read request.

7.4 Command 01_{hex} (Inventory Request)

This command is used to identify the transponders within range of the HF PUK.

The transponders respond to the Inventory request in one of 16 time slots. The slot they use is dependent on their address (UID) and the mask value. Only transponders whose UID match the mask will respond. If more than one transponder responds in a time slot then a collision is generated.

If a non-zero mask length is used then the mask bits are used to determine which transponders respond. The mask is applied from least significant bit of the UID up to the mask length. Only transponders with UIDs that match the bits of the mask respond. The next four most significant bits above the mask are used to determine which timeslot the transponder responds in.

For example, if the mask length is zero then all transponders whose 4 LSBits are 0000_{bin} will respond in the first time slot and transponders whose 4 LSBits are 0001_{bin} will respond in the next.

If the mask length is 4 and the mask bits are 0000_{bin} then the transponders whose 8 LSBits are 00000000_{bin} will respond in the first time slot and transponders whose 8 LSBits are 00010000_{bin} will respond in the next.

For further details on the inventory sequence, the AFI and the DSFID please refer to part 3 of the ISO15693 standard.

7.4.1 Request packet parameter section

AFI (1 Byte)	If the AFI flag is set then this field contains the AFI value. For transponders to respond their AFI value must match that of the command.
Mask length (1 Byte).	This is the number of bits in the following mask.
Mask	Initial match conditions for the UID address comparison by the transponder (LSB first). The MSByte of the mask should be padded with zeros if there are insufficient bits to fill the byte.

Example:

Perform an Inventory request without AFI using a mask value of 0110_{bin} using 100% modulation with 1:4 encoding and Fast data rate using two sub-carriers (FM).

(only those transponders whose 4 LSBits of their UID match 0110_{bin} will respond to the Inventory request)

Request Packet:

02 00 04 01 0A 02 00 04 06 1D 00

7.4.2 Response packet parameter section

The parameter section contains 16 sub-sections one for each time slot. Each sub-section starts with a status byte that indicates the contents of the sub-section. There are three possible values for the status byte:-

00_{hex} The sub-section is empty because no transponder was detected.

01_{hex} The sub-section is empty because a collision was detected.

02_{hex} The sub-section contains transponder information. The first byte is the DSFID followed by the 8 byte transponder UID (LSB first).

Example:

Response Packet:

02 00 04 01 00 19 00 00 00 02 00 26 32 54 76 00 00 00 E0 00 00 01 00 00 00 00 00 00 00 00 00 00 25 02

Time Slot	*Response*
0, 1, 3, 4, 6 -15	No transponder detected
2	Transponder whose UID is $E000000076543226_{hex}$ and whose DSFID is $00_{hex.}$
5	Collision detected

The next step in the inventory process is to resolve the collisions. To achieve this in the example above a mask of 01010110_{bin} would be used on the next inventory request.

Example:

Request Packet:

02 00 04 01 0A 02 00 08 56 71 00

Response Packet:

02 00 04 01 00 22 00 00 00 02 06 56 32 54 76 00 00 00 E0 02 03 56 33 54 76 00 00 00 E0 00 00 00 00 00 00 00 00 00 00 00 00 9B 04

Time Slot	*Response*
0, 1, 4 -15	No transponder detected
2	Transponder whose UID is $E000000076543256_{hex}$ and whose DSFID is 06_{hex}.
3	Transponder whose UID is $E000000076543356_{hex}$ and whose DSFID is 03_{hex}.

7.5 Command 02_{hex} (Stay Quiet)

This command is used to prevent a transponder from responding to the inventory request.

The quiet command will only work in addressed modes.

7.5.1 Request packet parameter section

Address (8 bytes).	The eight bytes of the parameter section contain the transponder UID LSB first

Example:

Quiet the transponder whose address is $E000000000C3213E_{hex}$ using 100% modulation with 1:256 encoding and slow data rate using one sub-carriers (AM)

Request Packet:

02 00 04 02 5C 08 00 3E 21 C3 00 00 00 00 E0 6E 02

Note:

It is also permissible to use an option byte value of 14 because the HF PUK will ignore the address flag as addressed mode is compulsory.

7.5.2 Response packet parameter section

The transponder does not respond to this command so it is normal to get a transponder not present error.

Example:

Response Packet:

02 00 04 02 FF 01 00 07 0F 01

Transponder not present error

7.6 Command 20_{hex} (Read Single Block)

This command is used to read a block of data from the transponder. If the option flag is clear then the transponder only sends block data. If the option flag is set the transponder also includes the block security status.

7.6.1 Request packet parameter section

Address (0 or 8 bytes).	If the address flag is set the first eight bytes of the parameter section contain the transponder UID LSB first
Block Number (1 Byte).	This is the number of the block you want to read.

Example:

Request data from block 4 of a transponder whose address is $E000000001234567_{hex}$ and include the security status using 100% modulation with 1:4 encoding and Fast data rate using two sub-carriers (FM).

Request Packet:

02 00 04 20 4B 09 00 67 45 23 01 00 00 00 E0 04 2E 02

7.6.2 Response packet parameter section

Security Status (0 or 1 Byte).	If the option flag was set then the least significant bit indicates the lock status of the block.
Data (x Bytes).	The actual data read from the block LSB first. The length of this data is dependent on the transponder block size.

Example:

Response Packet:

02 00 04 20 00 05 00 01 11 22 33 44 D6 00

Lock status: 01_{hex} (locked)

Data: 44332211_{hex}

7.7 Command 21_{hex} (Write Single Block)

This command is used to write a block of data to a transponder.

7.7.1 Request packet parameter section

Address (0 or 8 bytes).	If the address flag is set the first eight bytes of the parameter section contain the transponder UID LSB first
Block Number (1 Byte).	This is the number of the block you want to write.
Data (x Bytes).	The data to write to the block LSB first. The length of this data is dependent on the transponder

Example:

Write 01234567_{hex} to block 3 of a transponder non-addressed using 100% modulation with 1:4 encoding and fast data rate using two sub-carriers (FM).

Request Packet:

02 00 04 21 0B 05 00 03 67 45 23 01 0A 01

7.7.2 Response packet parameter section

There are no fields in the parameter section.

Example:

Response Packet:

02 00 04 21 00 00 00 27 00

Write successful.

The HF PUK will generate error code 09_{HEX} if the option flag is clear. This does not mean the command failed just that it is not verified.

It is recommended that this command be used with the option flag set. If the transponder you are using does not support this then perform a suitable read command after the write to confirm the write operation.

7.8 Command 22_{hex} (Lock Block)

This command is used to lock a block of data in a transponder.

7.8.1 Request packet parameter section

Address (0 or 8 bytes).	If the address flag is set the first eight bytes of the parameter section contain the transponder UID LSB first
Block Number (1 Byte).	This is the number of the block you want to lock.

Example:

Lock block 3 of a transponder non-addressed using 100% modulation with 1:4 encoding and fast data rate using two sub-carriers (FM).

Request Packet:

02 00 04 22 0B 01 00 03 37 00

7.8.2 Response packet parameter section

There are no fields in the parameter section.

Example:

Response Packet:

02 00 04 22 00 00 00 28 00

Lock successful.

The HF PUK will generate error code 09_{HEX} if the option flag is clear. This does not mean the command failed, just that it is not verified.

It is recommended that this command be used with the option flag set. If the transponder you are using does not support this then perform a suitable read command after the write to confirm the write operation.

7.9 Command 23_{hex} (Read Multi-Block)

This command is used to read several blocks of data from the transponder. If the option flag is clear then the transponder only sends block data. If the option flag is set the transponder also includes the block security status.

7.9.1 Request packet parameter section

Address (0 or 8 bytes).	If the address flag is set the first eight bytes of the parameter section contain the transponder UID LSB first
First Block Number (1 Byte).	This is the number of the first block you want to read.
Number of blocks -1(1 Byte).	This is the number of blocks you want to read minus 1. (A value of 00_{hex} will read one block)

Example:

Request data from blocks 4, 5 and 6 of a transponder whose address is $E000000001234567_{hex}$ and include the security status using 100% modulation with 1:4 encoding and Fast data rate using two sub-carriers (FM).

Request Packet:

02 00 04 23 4B 0A 00 67 45 23 01 00 00 00 E0 04 02 34 02

7.9.2 Response packet parameter section

Security Status (0 or 1 Byte).	If the option flag was set then the least significant bit indicates the lock status of the block.
Data (x Bytes).	The actual data read from the block LSB first. The length of this data is dependent on the transponder block size.

The Security Status and the Data section are repeated for each block requested.

Example:

Response Packet:

02 00 04 23 00 0F 00 01 11 22 33 44 00 55 66 77 88 00 01 23 45 67 6D 03

Lock status:	01_{hex} (locked)	(Block 4)
Data:	44332211_{hex}	
Lock status:	00_{hex} (unlocked)	(Block 5)
Data:	88776655_{hex}	
Lock status:	00_{hex} (unlocked)	(Block 6)
Data:	67452301_{hex}	

7.10 Command 24_{hex} (Write Multi-Block)

This command is used to write several blocks of data to a transponder.

7.10.1 Request packet parameter section

Address (0 or 8 bytes).	If the address flag is set the first eight bytes of the parameter section contain the transponder UID LSB first.
First Block Number (1 Byte).	This is the number of the block you want to write.
Number of blocks – 1 (1 Byte).	This is the number of blocks you want to write minus one.
Data (x Bytes).	The data to write to the blocks LSB first. The length of this data is dependent on the transponder.

Example:

Write $0123456789ABCDEF_{hex}$ to blocks 3 and 4 of a transponder non addressed using 100% modulation with 1:4 encoding and Fast data rate using two sub-carriers (FM).

Request Packet:

02 00 04 24 0B 0A 00 03 01 EF CD AB 89 67 45 23 01 03 04

7.10.2 Response packet parameter section

There are no fields in the parameter section.

Example:

Response Packet:

02 00 04 24 00 00 00 2A 00

Write successful.

The HF PUK will generate error code 09_{HEX} if the option flag is clear. This does not mean the command failed just that it is not verified.

It is recommended that this command be used with the option flag set. If the transponder you are using does not support this then perform a suitable read command after the write to confirm the write operation.

7.11 Command 25_{hex} (Select)

This command is used to put the addressed transponder into the select state so it can be accessed using the select flag. Only one transponder can be in the select state therefore if a selected transponder sees a select command that does not match with its address then that transponder will return to the ready state.

The select command will only work in addressed modes.

7.11.1 Request packet parameter section

Address (8 bytes).	The eight bytes of the parameter section contain the transponder UID LSB first

Example:

Select the transponder whose address is $E000000000C3213E_{hex}$ using 100% modulation with 1:256 encoding and slow data rate using one sub-carrier (AM)

Request Packet:

02 00 04 25 5C 08 00 3E 21 C3 00 00 00 00 E0 91 02

7.11.2 Response packet parameter section

There are no fields in the parameter section.

Example:

Response Packet:

02 00 04 25 00 00 00 2B 00

Select successful.

7.12 Command 26_{hex} (Reset to Ready)

This command is used to return transponders in the quiet or selected state back to the ready state.

If the select flag is set then only the selected transponder is reset.

If the address flag is set then only the addressed transponder is reset.

If the neither the select or address flag is set then all quieted transponders are reset.

7.12.1 Request packet parameter section

Address (0 or 8 bytes).	If the address flag is set the first eight bytes of the parameter section contain the transponder UID LSB first.

Example:

Reset to ready the selected transponder using 100% modulation with 1:4 encoding and Fast data rate using two sub-carriers (FM).

Request Packet:

02 00 04 26 8A 00 00 B6 00

7.12.2 Response packet parameter section

There are no fields in the parameter section.

Example:

Response Packet:

02 00 04 26 00 00 00 2C 00

Select successful.

7.13 Command 27_{hex} (Write AFI)

This command is used to write the AFI value to the transponder.

7.13.1 Request packet parameter section

Address (0 or 8 bytes).	If the address flag is set the first eight bytes of the parameter section contain the transponder UID LSB first
AFI Value (1 Byte).	This is the value of the AFI you want to write.

Example:

Write 03_{hex} to the AFI of a transponder non-addressed using 100% modulation with 1:4 encoding and fast data rate using two sub-carriers (FM).

Request Packet:

02 00 04 27 0B 01 00 03 3C 00

7.13.2 Response packet parameter section

There are no fields in the parameter section.

Example:

Response Packet:

02 00 04 27 00 00 00 2D 00

Write successful.

The HF PUK will generate error code 09_{HEX} if the option flag is clear. This does not mean the command failed, just that it is not verified.

It is recommended that this command be used with the option flag set. If the transponder you are using does not support this then perform a suitable read command after the write to confirm the write operation.

7.14 Command 28_{hex} (Lock AFI)

This command is used to lock the AFI value in the transponder. This operation cannot be reversed, once locked the AFI value can never be changed.

7.14.1 Request packet parameter section

Address (0 or 8 bytes).	If the address flag is set the first eight bytes of the parameter section contain the transponder UID LSB first

Example:

Lock the AFI value of a transponder non-addressed using 100% modulation with 1:4 encoding and fast data rate using two sub-carriers (FM).

Request Packet:

02 00 04 28 0B 00 00 39 00

7.14.2 Response packet parameter section

There are no fields in the parameter section.

Example:

Response Packet:

02 00 04 28 00 00 00 2E 00

Lock successful.

The HF PUK will generate error code 09_{HEX} if the option flag is clear. This does not mean the command failed, just that it is not verified.

It is recommended that this command be used with the option flag set. If the transponder you are using does not support this then perform a suitable read command after the write to confirm the write operation.

7.15 Command 29_{hex} (Write DSFID)

This command is used to write the DSFID value to the transponder.

7.15.1 Request packet parameter section

Address (0 or 8 bytes).	If the address flag is set the first eight bytes of the parameter section contain the transponder UID LSB first
DSFID Value (1 Byte).	This is the value of the DSFID you want to write.

Example:

Write 03_{hex} to the DSFID of a transponder non-addressed using 100% modulation with 1:4 encoding and fast data rate using two sub-carriers (FM).

Request Packet:

02 00 04 29 0B 01 00 03 3E 00

7.15.2 Response packet parameter section

There are no fields in the parameter section.

Example:

Response Packet:

02 00 04 29 00 00 00 2F 00

Write successful.

The HF PUK will generate error code 09_{HEX} if the option flag is clear. This does not mean the command failed, just that it is not verified.

It is recommended that this command be used with the option flag set. If the transponder you are using does not support this then perform a suitable read command after the write to confirm the write operation.

7.16 Command $2A_{hex}$ (Lock DSFID)

This command is used to lock the DSFID value in the transponder. This operation cannot be reversed, once locked the DSFID value can never be changed.

7.16.1 Request packet parameter section

Address (0 or 8 bytes).	If the address flag is set the first eight bytes of the parameter section contain the transponder UID LSB first

Example:

Lock the DSFID value of a transponder non-addressed using 100% modulation with 1:4 encoding and fast data rate using two sub-carriers (FM).

Request Packet:

02 00 04 2A 0B 00 00 3B 00

7.16.2 Response packet parameter section

There are no fields in the parameter section.

Example:

Response Packet:

02 00 04 2A 00 00 00 30 00

Lock successful.

The HF PUK will generate error code 09_{HEX} if the option flag is clear. This does not mean the command failed, just that it is not verified.

It is recommended that this command be used with the option flag set. If the transponder you are using does not support this then perform a suitable read command after the write to confirm the write operation.

7.17 Command $2B_{hex}$ (Get System Information)

This command is used to retrieve information about the transponder.

7.17.1 Request packet parameter section

Address (0 or 8 bytes). If the address flag is set the first eight bytes of the parameter section contain the transponder UID LSB first

Example:

Get the system information from a transponder non-addressed using 100% modulation with 1:4 encoding and fast data rate using two sub-carriers (FM).

Request Packet:

02 00 04 2B 0A 00 00 3B 00

7.17.2 Response packet parameter section

Info Flags (1 Bytes). This flags in this byte indicate which fields follow the UID.

Transponder UID (8 Byte) This is the transponder's UID LSB first.

DSFID (0 or 1 Byte) If supported this byte is the transponder's DSFID.

AFI (0 or 1 Byte) If supported this byte is the transponder's AFI.

Memory size (2 Bytes) If supported This indicates the memory size and structure of the transponder LSB first.

IC reference (1 Byte) If supported this byte is the transponder's IC reference.

Info Flag Details

Bit	Flag Name	Value	Description
0	DSFID	0	DSFID is not supported. DSFID field is not present.
		1	DSFID is supported. DSFID field is present.
1	AFI	0	AFI is not supported. AFI field is not present.
		1	AFI is supported. AFI field is present.
2	Memory Size	0	Memory size is not supported. Memory size field is not present.
		1	Memory size is supported. Memory size field is present.
3	IC reference	0	IC reference is not supported. IC reference field is not present.
		1	IC reference is supported. IC reference field is present.
4	RFU	0	
5	RFU	0	
6	RFU	0	
7	RFU	0	

Memory Size

The memory size field is 16 bits.

The 8 LSB's represent the number of blocks minus one.

The next 5 bits represent the number of bytes per block minus one.

The 3 MSB's are reserved for future use.

This allows for up to 256 blocks of data and up to 32 bytes per block.

Example:

Response Packet:

02 00 04 2B 00 0E 00 0F 10 32 54 76 00 00 00 E0 05 00 3F 03 04 85 02

Info flags	$0F_{hex}$	All fields are present
UID:	$E000000076543210_{hex}$	
DSFID:	05_{hex}	
AFI:	00_{hex}	
Memory Size:	$033F_{hex}$	64 blocks 4 bytes per block
IC Reference:	04_{hex}	

7.18 Command $2C_{hex}$ (Get Multi-Block Security Status)

This command is used to retrieve the security status of several blocks.

7.18.1 Request packet parameter section

Address (0 or 8 bytes).	If the address flag is set the first eight bytes of the parameter section contain the transponder UID LSB first
First Block Number (1 Byte).	This is the number of the first block you want to know the status of.
Number of blocks -1(1 Byte).	This is the number of blocks you want to know the status of minus 1. (A value of 00_{hex} will read one block)

Example:

Request the security status from blocks 4, 5 and 6 of a transponder whose address is $E000000001234567_{hex}$ using 100% modulation with 1:4 encoding and Fast data rate using two sub-carriers (FM).

Request Packet:

02 00 04 2C 4B 0A 00 67 45 23 01 00 00 00 E0 04 02 3D 02

7.18.2 Response packet parameter section

Security Status (1 Byte).	The least significant bit indicates the lock status of the block.

The Security Status section is repeated for each block requested.

Example:

Response Packet:

02 00 04 2C 00 03 00 01 00 00 36 00

Lock status:	01_{hex}	(locked)	(Block 4)
Lock status:	00_{hex}	(unlocked)	(Block 5)
Lock status:	00_{hex}	(unlocked)	(Block 6)

8 Inside Technology PicoTag Commands Destination 05_{hex}

Inside Technology PicoTag transponders are supported on the HF PUK using a command destination byte value of $05_{hex.}$

The reader is expected to be familiar with Inside Technologies PicoTag transponders, the HF PUK packet structure and the HF PUK error handling. This document should be read in conjunction with the relevant Inside Technology transponder data sheets.

To aid clarity the example packets in this document have the parameter section greyed.

Before you can read and write data to a PicoTag transponder the transponder must be selected either using the Select (02_{hex}) command if the serial number of the transponder is known or by using the Anti-collision Select (01_{hex}) command.

The Anti-collision Select (01_{hex}) command will select one transponder from the field and return its serial number. The selected transponder may now be read from using the Read Block (04_{hex}) command, written to using the Write Block (05_{hex}) command or halted using the Halt (03_{hex}) command. Once a transponder is halted it will take no further part in the anti-collision process. By repeating Anti-collision Select, Read, Write and Halt commands all of the transponders in the field can be worked with.

8.1 Inside Technologies PicoTag Option Byte Flags

The option byte is not used with Inside Technology Transponders. All bits should be set to zero.

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

8.2 Inside Technologies PicoTag Errors

No transponder specific errors are generated.

8.3 Inside Technologies PicoTag Commands

Below is a list of the available Inside Technologies PicoTag Commands.

Command Description	Command Value
Anti-Collision Select	0x01
Select	0x02
Halt	0x03
Read Block	0x04
Write Block	0x05

8.4 Command 01_{hex} (Anti-Collision Select)

This command is used to select one transponder from all the transponders within the range of the HF PUK and return its serial number.

8.4.1 Request packet parameter section

There are no fields in the parameter section.

Example:

Perform an Anti-Collision Select request.

Request Packet:

02 00 05 01 00 00 00 08 00

8.4.2 Response packet parameter section

Serial Number (8 Bytes). This is the serial number of the selected transponder.

Example:

Response Packet:

02 00 05 01 00 08 00 73 92 E4 00 00 00 C0 00 B9 02

The selected transponder's serial number is $00C0000000E49273_{hex}$

8.5 Command 02_{hex} (Select)

This command is used to directly select a specific transponder using its serial number.

8.5.1 Request packet parameter section

Serial Number(8 bytes). The eight byte serial number of the transponder to be selected.

Example:

Select the transponder whose serial number is $00C0000000C3213E_{hex}$.

Request Packet:

02 00 05 02 00 08 00 3E 21 C3 00 00 00 C0 00 F3 01

8.5.2 Response packet parameter section

Serial Number (8 Bytes). This is the serial number of the selected transponder.

Example:

Response Packet:

02 00 05 02 00 08 00 3E 21 C3 00 00 00 C0 00 F3 01

The selected transponder's serial number is $00C0000000E49273_{hex}$

8.6 Command 03_{hex} (Halt)

This command is used to halt the selected transponder. The halted transponder will remain halted until it is directly selected using the Select (02_{hex}) command or it is removed from the RF field.

8.6.1 Request packet parameter section

There are no fields in the parameter section.

Example:

Halt the currently selected transponder.

Request Packet:

02 00 05 03 00 00 00 0A 00

8.6.2 Response packet parameter section

There are no fields in the parameter section.

Example:

Response Packet:

02 00 05 03 00 00 00 0A 00

The transponder was successfully halted.

8.7 Command 04_{hex} (Read Block)

This command is used to read a block of data from the selected transponder.

8.7.1 Request packet parameter section

Block Number (1 Byte). This is the number of the block you want to read.

Example:

Read block 5 of the selected transponder.

Request Packet:

02 00 05 04 00 01 00 05 11 00

8.7.2 Response packet parameter section

Block Data (8 Bytes). This is the data read from the requested block of the selected transponder LSB first.

Example:

Response Packet:

02 00 05 04 00 08 00 00 11 22 33 44 55 66 77 EF 01

Block data 7766554433221100_{hex}

8.8 Command 05_{hex} (Write Block)

This command is used to write a block of data to the selected transponder.

8.8.1 Request packet parameter section

Block Number (1 Byte).	This is the number of the block you want to write.
Data (8 Bytes)	The data to write to the block LSB first.

Example:

Write 0011223344556677_{hex} to block 5 of the selected transponder.

Request Packet:

02 00 05 05 00 09 00 05 77 66 55 44 33 22 11 00 F6 01

8.8.2 Response packet parameter section

Block Data (8 Bytes).	This is the data read back from the block after the write has been completed on the selected transponder LSB first. This should be checked to ensure the write operation was successful.

Example:

Response Packet:

02 00 05 05 00 08 00 77 66 55 44 33 22 11 00 F0 01

Block data 0011223344556677_{hex}