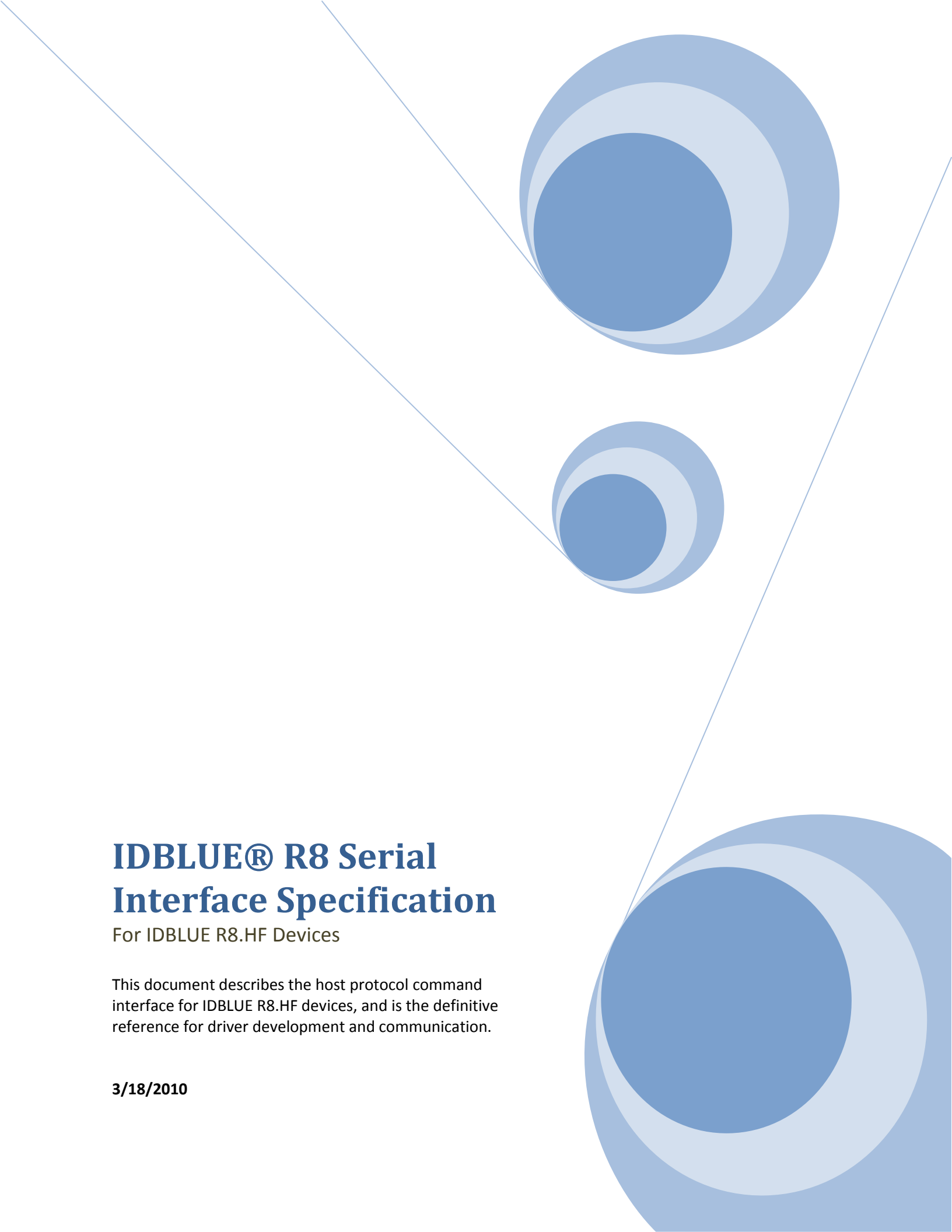


IDBLUE® R8 Serial Interface Specification

For IDBLUE R8.HF Devices

This document describes the host protocol command interface for IDBLUE R8.HF devices, and is the definitive reference for driver development and communication.

3/18/2010

Version History

Version	Timestamp	Description
3.8	March 18th, 2010	Update 56h responses to READ_BLOCKS and WRITE_BLOCKS for incomplete operations. Update how to properly use block count and block index. Update GET_PROPERTY NACK for invalid parameter. Fix maximum length of Bluetooth Name.
3.7	February 25th, 2010	Update description of READ_BLOCKS and WRITE_BLOCKS to indicate 0 is not valid.
3.6	June 10th, 2009	Added SPI commands HEARTBEAT and ENABLE_CHANNEL and remove the BLUETOOTH_ON command. Added USB support and descriptions of logical and physical communication channels. Added the Duplicate Elimination error code (58h).
3.5	March 16th, 2009	Added consistent failure indications to all commands. Updated NACK response to include command id, failure type and optional additional information. All RFID operations (GET_TAG_ID. READ_BLOCK, READ_BLOCKS, WRITE_BLOCK, WRITE_BLOCKS, GET_TAG_INFO) all indicate the timestamp of the RFID operation in the successful response. All requests and responses which pass a tag id now include the tag id length of the tag id. The GET_STATUS and Version Information Property now return the Major and Minor numbers of the hardware version and Major, Minor, Branch and Build numbers of the firmware version. SET_CONNECTED_MODE has been removed.
3.4	September 22nd, 2008	READ_SINGLE changed to READ_BLOCK READ_MULTI changed to READ_BLOCKS WRITE_SINGLE changed to WRITE_BLOCK WRITE_MULTI changed to WRITE_BLOCKS Removed READ_MULTI_16, WRITE_MULTI_16 and SELECT_TAG commands Removed ENTER_BOOTLOAD commands for both R7 and R8 devices Added BLUETOOTH_ON command
3.3	September 3rd, 2008	Updated Factory Default Settings. Added invalid time disclaimer to Time property definition.
3.2	October 20th, 2007	Added SET_CONNECTED_MODE property.
3.1	October 5th, 2007	Added complete command set and property table.
3.0	August 15th, 2007	Initial beta version of firmware V3.0 release for the IDBLUE R8 product.

Disclaimer

Cathexis Innovations Inc. assumes no responsibility for any errors which may appear in this document, reserves the right to change systems or specifications detailed herein at any time without notice, and does not make any commitment to update the information contained herein. No licenses to patents or other intellectual property of Cathexis are granted by the Company in connection with the sale of Cathexis products, expressly or by implication.

All specifications are subject to change without notice.

Cathexis, IDBLUE, CathexisWEB, and Powered by Cathexis RFID Engine are either registered trademarks or trademarks of Cathexis Innovations Inc. in Canada and/or other countries.

Trademarks

IDBLUE® is a registered trademark of Cathexis Innovations Inc. (http://www.idblue.com/).

Bluetooth® is a registered trademark of the Bluetooth SIG (http://www.bluetooth.com/).

Microsoft, Microsoft Windows XP, Microsoft Windows Vista, Microsoft Windows Server 2003, PocketPC and Windows Mobile are either registered trademarks or trademarks of Microsoft Corporation (http://www.microsoft.com/) in the United States and/or other countries.

Introduction

The intent of this document is to provide software developers with the definitive specification for developing host interfaces with the IDBLUE R8.HF product. It contains a description of the binary interface used to communicate with the IDBLUE device, enabling development of drivers and API's.

> **Note:** IDBLUE only provides access to this document and support for development of 3rd party drivers under the terms of a support agreement. This enables the company to provide the best go-to-market support possible for your solution.
>
> If you do not have a platform support contact in place with IDBLUE, please contact sales@idblue.com for details.

Background and Intended Audience

This document assumes that the reader has experience in developing device drivers for microelectronic devices with a background in data communication protocols. The reader must be familiar with their target development platform, including language and hardware capabilities.

Experience with serial communication, USB and Bluetooth® interfaces using the requisite platform library and development environment is highly recommended.

Supported Protocols

The IDBLUE R8.HF device supports the following RFID protocols:

Protocol	Tag Id Length	Block Data Length	R8
ISO 15693	8 bytes (64 bits)	4-8 bytes (varies)	X

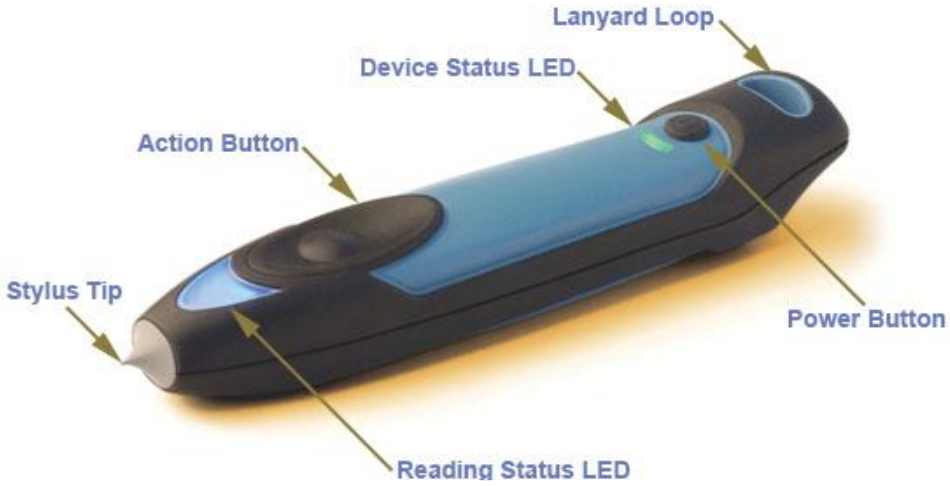
For a complete list of tested tags, please refer to the IDBLUE website at http://www.idblue.com.

Device Overview

Please take a moment to examine the IDBLUE device, and familiarize yourself with its components.

Components:

1. Reading Status LED
2. Device Status LED
3. Action Button
4. Power Button
5. Lanyard Loop
6. Stylus Tip
7. Device Name
8. USB Port



Communication

Physical Connections

The IDBLUE R8.HF reader provides USB and Bluetooth communication to interact with the device. A typical physical connection establishes a “virtual serial port” at the host operating system level. This virtual port should be configured with the following parameters:

Baud Rate	57,600 bps
Data Bits	8
Stop Bits	1
Parity	None
Flow Control	None

IDBLUE is considered to be physically connected when IDBLUE is plugged into a computer and/or there is an established Bluetooth connection. It is possible to be physically connected via both USB and Bluetooth.

Logical Connections

IDBLUE is considered logically connected when there is a logical channel established to a host application. This logical connection is established via the ENABLE_CHANNEL command, which establishes a logical communication channel over either USB or Bluetooth. IDBLUE will allow only one logical connection at a time. Once the command is received over one of the physical channels, IDBLUE will prevent another logical connection via the other physical channel. In the case of establishing a logical USB connection, IDBLUE will actually disable Bluetooth and drop any current physical connection.

The typical flow of communication with the device is through a series of request/response packets (the format of these packets is detailed in the Commands section). Asynchronous events are also generated by the device in response to certain stimuli and are typically used to notify the host application of a specific happening (such as the action button being pressed).

Developers should ensure that a connection (or disconnection) attempt is complete and a response to the ENABLE_CHANNEL command is received before starting a disconnection (or connection) attempt. Failing to wait until the operation is complete may leave the IDBLUE device in a state whereby further connection attempts will always fail. The IDBLUE device must be reset to re-connect if this occurs.

Developers should ensure that no commands are sent to IDBLUE in the first 500 ms after a connection has been established. Failing to wait will result in the commands failing.

For backwards compatibility with current applications which have not implemented the ENABLE_CHANNEL request, the IDBLUE device does not enforce the use of ENABLE_CHANNEL when communicating over Bluetooth. The IDBLUE device will enable the logical communication channel upon receipt of any request over Bluetooth. However, the IDBLUE device does not utilize the heartbeat protocol unless it receives an ENABLE_CHANNEL request. Please refer to the Communication Heartbeat section for more information.

USB

The IDBLUE device with firmware versions 4.1.0 and above supports USB via the supplied USB cable. When developing a driver library for the IDBLUE R8.HF device, this connection is typically established via a USB serial port.

Bluetooth

The IDBLUE device supports Bluetooth connections using the serial port profile (SPP). It is beyond the scope of this document on how to establish this connection as it varies greatly between platforms.

When developing a driver library for the IDBLUE R8.HF device, this connection is typically established either via a pre-configured Bluetooth serial port, or the Bluetooth libraries provided by the platform and programming language.

> The default Bluetooth PIN code for all IDBLUE devices is "0000" (i.e. four zeroes). This PIN code is required to Bond or Pair with the device (this is a prerequisite of establishing a serial connection through the SPP profile).

For firmware version 4.0.0 the Bluetooth connection is strictly a slave device (i.e. requires that an external host establish the connection). After establishing an SPP connection to the IDBLUE device, it is recommended that your application delay for 10-20 ms in order to allow the connection to stabilize.

Communication Heartbeat

Beginning in firmware version 4.1.0, a communication heartbeat has been implemented such that IDBLUE can distinguish a loss of a logical connection in a timely manner. Once a logical connection has been established via the ENABLE_CHANNEL command, IDBLUE expects a regular heartbeat to be initiated from the host application. Upon receipt of the HEARTBEAT command, IDBLUE will respond with the HEARTBEAT response such that the host application can ensure that the connection to IDBLUE remains valid.

IDBLUE expects to receive a command (HEARTBEAT or otherwise) every 3 seconds to ensure communication with the host application. If no command is received from the host application within the 3 second interval, IDBLUE will drop the logical connection.

> Receipt of the HEARTBEAT command does not reset the Device Timeout in IDBLUE. IDBLUE will turn off if only the HEARTBEAT command is received for a duration specified for the Device Timeout. IDBLUE interprets all other commands received as host communication and does reset the Device Timeout.

> For backwards compatibility with current applications which do not implement the heartbeat, the IDBLUE does not enforce the use of the ENABLE_CHANNEL command. However, the IDBLUE device does not utilize the heartbeat protocol unless it receives an ENABLE_CHANNEL request.

Device Behavior and Configuration

The behavior of the IDBLUE device is managed through a set of configuration properties. These properties are used to tune the behavior of the device, including power management, RFID, and user feedback.

Developer Tips

- Tag IDs are sent and received in LSB order, but need to be displayed as MSB order. The byte array containing the tag ID should be reversed when either sending a command to the device, or receiving a response from the device.

- The data length value does not include the checksum, only the payload.

User Feedback

This section details how the user interface elements of IDBLUE R8.HF respond to different events and device states. IDBLUE R8 provides user feedback through three different methods:

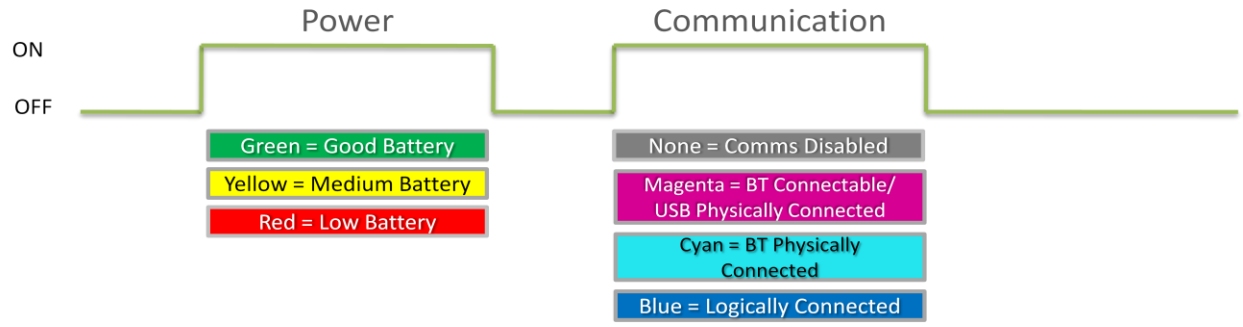
1. Device Status LED – Provides feedback on battery level and communication status
2. RFID Status LED– Provides feedback on RFID related events
3. Audio Buzzer – Provides feedback on RFID and transition events

Device Status LED

The device status LED displays the status of the power and communications of the device. The patterns of colors displayed will differ depending on if the device is on or off, and charging or not charging.

Not Charging

When on but not charging, the device status LED default value is off, and will flash two consecutive colors in a continuous pattern. The first flash indicates power (battery level) and the second communication. The color of each of these flashes will indicate the various states as shown below:



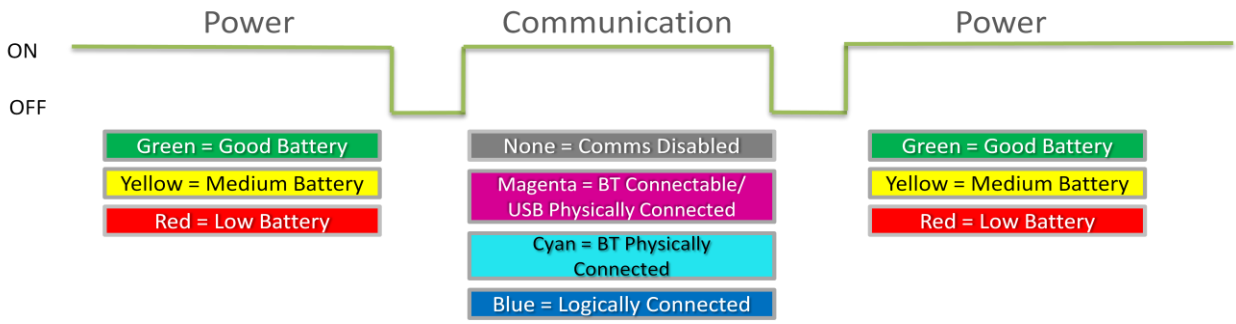
Device Status	Power Color	Communication Color
Comms Disabled – Good Battery	Green	None
Comms Disabled – Medium Battery	Yellow	None
Comms Disabled – Low Battery	Red	None
Connectable – Good Battery	Green	Magenta
Connectable – Medium Battery	Yellow	Magenta
Connectable – Low Battery	Red	Magenta
USB Physically Connected – Good Battery	Green	Magenta
USB Physically Connected – Medium Battery	Yellow	Magenta
USB Physically Connected – Low Battery	Red	Magenta
Bluetooth Physically Connected – Good Battery	Green	Cyan
Bluetooth Physically Connected – Medium Battery	Yellow	Cyan
Bluetooth Physically Connected – Low Battery	Red	Cyan
Logically Connected – Good Battery	Green	Blue
Logically Connected – Medium Battery	Yellow	Blue

Logically Connected – Low Battery	Red	Blue
Off	None	None

Table 1 : Device Status LED When Not Charging

Charging

When charging, the power LED is normally on, and will flash a single color. The solid color indicates the current charge level of the battery and the second will indicate the communication status as shown below:



Device Status	Power Color	Communication Color
Comms Disabled – Good Battery	Green	None
Comms Disabled – Medium Battery	Yellow	None
Comms Disabled – Low Battery	Red	None
Connectable – Good Battery	Green	Magenta
Connectable – Medium Battery	Yellow	Magenta
Connectable – Low Battery	Red	Magenta
USB Physically Connected – Good Battery	Green	Magenta
USB Physically Connected – Medium Battery	Yellow	Magenta
USB Physically Connected – Low Battery	Red	Magenta
Bluetooth Physically Connected – Good Battery	Green	Cyan
Bluetooth Physically Connected – Medium Battery	Yellow	Cyan
Bluetooth Physically Connected – Low Battery	Red	Cyan
Logically Connected – Good Battery	Green	Blue
Logically Connected – Medium Battery	Yellow	Blue
Logically Connected – Low Battery	Red	Blue
Off	Green	None

Table 2 : Device Status LED When Charging

Bluetooth Discoverable

When IDBLUE is in Bluetooth Discoverable mode, the LED pattern will flash blue to indicate this unique state. The LED pattern will return to either the charging or not charging states after the Discovery timeout (60 seconds) or IDBLUE is connected via Bluetooth, whichever occurs first.

RFID Status LED

The Reading Status LED will display feedback on RFID operations. Refer to the table below.

Audio Buzzer

The audio buzzer will provide audio feedback on various connection and RFID events. Refer to the table below.

Events

The following table should be used as a reference to identify an event.

Event	Audio Buzzer	RFID Status LED	Device Status LED
Power On	Two High Tones	Two Green Flashes	Two Green Flashes
Power Off	Two High Tones	Two Green Flashes	Two Green Flashes
Physical Bluetooth Connection Established	Low to High Transition	None	Blue Flash and Communication Flash Sequence = Cyan
Physical Bluetooth Connection Dropped	High to Low Transition	None	Blue Flash and Communication Flash Sequence = Magenta
Physical USB Connection Established	Low to High Transition	None	None
Physical USB Connection Dropped	High to Low Transition	None	None
Logical Connection Established	None	None	Communication Flash Sequence = Blue
Logical Connection Dropped	None	None	Communication Flash Sequence = Cyan/Magenta
Action Button Pressed	None	Solid Blue	None
RFID Operation Success	High Tone	Green Flash	None
RFID Operation Failure	None	Red Flash	None
Tag Store – Invalid Clock	None	Two Yellow Flashes	None
Power Button Press and Hold	None	None	Solid Cyan (for 2 sec)
Power + Action Button Press and Hold	None	Solid Cyan (for 3 sec)	Solid Cyan (for 3 sec)

Table 3 : User Feedback on Events

Operating Modes

The IDBLUE device supports a number of different operating modes that govern its behavior when the action button is pressed. These operating modes are used to streamline the user experience of interacting with the device and to improve scanning performance.

Choosing the appropriate operating modes for a given scenario and workflow is a crucial aspect of good IDBLUE integration. This section outlines what the various operating modes do and how they are best applied. The device supports two sets of operating modes; those for use when the device has an active connection to a host device (Connected mode) and when the device is operating independently (Disconnected mode).

An IDBLUE device supports three scanning modes:

- **Single Scan**. When the user of an IDBLUE device presses the Action Button, the IDBLUE device will scan a single tag.
- **Hold to Scan**. The user of an IDBLUE device can press and hold the Action Button to continuously scan multiple tags.
- **Continuous Scan**. The IDBLUE device can also be configured for a continuous scan mode whereby pressing the Action Button will activate continuously scanning tags. The IDBLUE device will continuously scan tags until the Action Button is pressed again, or until the Continuous Scan Timeout elapses without scanning any tag. Continuous Scan will take precedence over the Hold To Scan.

The IDBLUE device can also be configured to prevent repeatedly scanning tags within a certain timeframe. The duplication elimination property specifies the timeframe for which subsequent scans of a particular tag will be ignored. This is particularly important in the Hold to Scan and Continuous Scan modes of operation where tags will likely be scanned multiple times by the IDBLUE device.

Connected

The IDBLUE device supports five connected operating modes.

Normal

The purpose of this mode is to streamline tag inventory scenarios wherein user data (i.e. data stored in user memory sections of RFID tags) is not required. In normal mode (also known as TAG_ID mode), when the Action Button is pressed and released the device will:

- Indicate a start of RFID operations by turning the Action LED blue.
- Attempt to scan a tag using the currently configured RFID protocol.
- If a tag is successfully scanned, the device will indicate with a green LED and audio beep (if audio feedback is enabled).
- An asynchronous GET_TAG_ID response packet will be sent to the connected host.

If the Continuous Scanning Enabled property is set to true, the device will:

- Enter continuous scanning mode when the Action Button is pressed and released.
- Attempt to scan a tag using the currently configured RFID protocol.
- If a tag is successfully scanned, the device will indicate with a green LED and audio beep (if audio feedback is enabled).
- An asynchronous GET_TAG_ID response packet will be sent to the host.
- Attempt to scan another tag

The device will exit continuous scanning mode if:

- The Continuous Scanning Timeout period elapses without a successful tag scan.
- The action button is pressed and released.

Reactive

The purpose of this mode is to handle non-trivial RFID interaction scenarios involving sequences of commands. In Reactive mode (also known as BUTTON PRESS mode), when the Action Button is pressed and released the device will:

- Indicate a start of RFID operations by turning the Action LED blue.
- Send an asynchronous button pressed packet to the connected host.

After an elapsed timeout of 1 second, with no commands received from the host, the device will:

- Indicate an unsuccessful end of RFID operations by flashing the Action LED red.

The host application should issue a END_COMMANDS command after all RFID commands have been sent (in response to the asynchronous button press event) to signal that the host is finished sending commands.

The Continuous Scanning Enabled property has no effect in Reactive mode.

Read Block N

The purpose of this mode is to streamline tag inventory scenarios wherein a small amount of user data (i.e. data stored in user memory sections of RFID tags) is required. In *Read Block N* mode, when the Action Button is pressed and released, the device will:

- Indicate a start of RFID operations by turning the Action LED blue.
- Attempt to read block N (where N is the current value of the BlockIndex property) using the currently configured RFID protocol.
- If a tag is successfully scanned, the device will indicate with a green LED and audio beep (if audio feedback is enabled).
- An asynchronous READ_BLOCK response packet will be sent to the connected host.

If the Continuous Scanning Enabled property is set to true, the device will attempt to scan another tag. The device will exit continuous scanning mode if:

- The Continuous Scanning Timeout period elapses without a successful tag scan.
- The Action Button is pressed and released.

Write Block N

The purpose of this mode is to streamline tag inventory scenarios wherein one wishes to initialize a small amount of user data (i.e. data stored in user memory sections of RFID tags). In *Write Block N* mode, when the Action Button is pressed, the device will:

- Indicate a start of RFID operations by turning the Action LED blue
- Attempt to write block N (where N is the current value of the BlockIndex property) using the currently configured RFID protocol, using the Block data found in the BlockData property.
- If a tag is successfully scanned, the device will indicate with a green LED and audio beep (if audio feedback is enabled).
- An asynchronous WROTE_BLOCK response packet will be sent to the connected host.

If the Continuous Scanning Enabled property is set to true, the device will attempt to scan another tag. The device will exit continuous scanning mode if:

- The Continuous Scanning Timeout period elapses without a successful tag scan.
- The Action Button is pressed and released.

Read Blocks

The purpose of this mode is to streamline tag inventory scenarios wherein a moderate amount of user data (i.e. data stored in user memory sections of RFID tags) is required. In *Read Blocks* mode, when the Action Button is pressed, the device will:

- Indicate a start of RFID operations by turning the Action LED blue
- Attempt to read block N+M (where N is the current value of the BlockIndex property and M is the number of blocks to read from the BlockCount property) using the currently configured RFID protocol.
- If a tag is successfully scanned, the device will indicate with a green LED and audio beep (if audio feedback is enabled).
- An asynchronous READ_BLOCKS response packet will be sent to the connected host.

If the Continuous Scanning Enabled property is set to true, the device will attempt to scan another tag. The device will exit continuous scanning mode if:

- The Continuous Scanning Timeout period elapses without a successful tag scan.
- The Action Button is pressed and released.

Disconnected

The IDBLUE device supports two disconnected operating modes.

Tag Verify

The purpose of this mode is to streamline tag inventory scenarios wherein user data (i.e. data stored in user memory sections of RFID tags) is not required, and the data does not need to be recorded. In *Tag Verify* mode, when the Action Button is pressed and released, the device will:

- Indicate a start of RFID operations by turning the Action LED blue
- Attempt to scan a tag using the currently configured RFID protocol.
- If a tag is successfully scanned, the device will indicate with a green LED and audio beep (if audio feedback is enabled).

If the Continuous Scanning Enabled property is set to true, the device will attempt to scan another tag. The device will exit continuous scanning mode if:

- The Continuous Scanning Timeout period elapses without a successful tag scan.
- The Action Button is pressed and released.

Store Timestamp + Tag Id + Block 0 Data

The purpose of this mode is to streamline tag inventory and recording scenarios wherein user data (i.e. data stored in user memory sections of RFID tags) is not required, and the data does need to be recorded and saved. In *Tag Store* mode (aka *Store Timestamp + Tag Id + Block 0 Data*), when the Action Button is pressed and released, the device will:

- Check for a valid timestamp value[1].
- If no valid timestamp is present (i.e. the device's timestamp has not been set), the LED will blink double-yellow.
- If a valid timestamp is present, the device will indicate a start of RFID operations by turning the Action LED blue.
- Attempt to read block 0 from a tag using the currently configured RFID protocol.
- If a tag is successfully scanned, the device will indicate with a green LED and audio beep (if audio feedback is enabled).

The current timestamp, tag ID, and contents of byte 0 (in block 0) on the tag will then be stored in the internal tag database (see Tag Store for details). This data can later be accessed through the GET_ENTRY command.

If the Continuous Scanning Enabled property is set to true, the device will attempt to scan another tag. The device will exit continuous scanning mode if:

- The Continuous Scanning Timeout period elapses without a successful tag scan.
- The Action Button is pressed and released.

[1] This check can be disabled through the TimeStampRequired property.

Note that disabling Duplicate Elimination and/or setting the Duplicate Elimination timeout to 0 while Continuous Scan or Hold To Scan are enabled will quickly fill up the Tag Store memory of the IDBLUE device. It is not recommended to enable Continuous Scan or Hold To Scan without enabling Duplicate Elimination and setting a Duplicate Elimination timeout.

Commands

Communication between the IDBLUE device and host occurs using a series of requests and responses. Requests are packets sent to the IDBLUE device from the host. They are used by the host to ask for information or for the device to perform some task.

Responses are sent from the IDBLUE device. They are the result sent back to the host after a request has been issued (or after an asynchronous event has occurred). Responses indicate either success and return information or indicate failure and an indication of the reason for failure.[2]

IDBLUE R8.HF will queue most commands and process them in sequential order, with the exception of SET_SCANNING(0) which stops a continuous scan operation.

Communication Packet Format

All communication between a connected host and an IDBLUE device takes the form of request/response or asynchronous response packets. Each packet has a command code, length, data payload and a checksum. The formats for the request and response packets are:

Content	Index	Type and Format	Description
Header	00	Single byte command code	Identifies the type of packet.
Data Length	01 - 02	Unsigned 16-bit integer, MSB format.	The remaining number of bytes in the packet, not including the checksum (i.e. the size of the data payload).
Payload	03-X	Data payload, format depends on command type.	The data payload for the given command. The exact format of this data is given in the description of the command.
Checksum	Final Byte	Unsigned 8-bit integer.	Checksum of the entire data packet, from Header through the Payload. See Appendix A for details on checksum generation.

For example, the POWER_DOWN command is used to check for the presence of a connected IDBLUE device. The POWER_DOWN command code is 0x91, and has no data payload. Therefore a POWER_DOWN request packet would take the form:

0x91, 0x00, 0x00, 0x91

Where the data length is 0000 (no payload), and the checksum is 0x91.

[2] **All byte values are displayed in a hexadecimal format.**

Asynchronous Events

In addition to the standard request/response packets, the IDBLUE device also supports a set of asynchronous notifications. These notifications are typically generated in response to the user pressing the Action Button on the device.

When the user presses the Action Button on the device, an asynchronous event is created, sending a response packet to the host. To designate this packet as an asynchronous event and not a possible response to an earlier command issued over the serial connection, a special packet (ASYNC) is sent before the main packet to indicate that it is an asynchronous event. This has the form:

Response format (Asynchronous Header)

Header	Data Length	Data	Checksum
70h	0000h	None	1-byte

This is true for all modes of operation during a button press. Applications (and driver libraries) should interpret the response packet immediately following an ASYNCH (asynchronous header) packet as an asynchronous event.

Response format (Button Pressed)

Header	Data Length	Data	Checksum
FFh	0000h	None	1-byte

Supported IDBLUE Commands

The table below provides a summary of all currently supported IDBLUE commands. Driver developers should incorporate support for all of these commands into any offering.

Command	Code	Description
NO_OP	00h	No operation; used to check for the presence of a connected device.
GET_TAG_ID	01h	Performs a tag inventory and returns the tag id of the RFID tag in the RF field.
BEEP	03h	Causes the device to emit one of the standard audio tone sequences.
SET_PROPERTY	08h	Sets one of the device's configuration properties. These values are volatile unless saved by issuing a SAVE_PROPERTIES command.
GET_PROPERTY	09h	Gets the current value of one of the device's configurable properties.
SAVE_PROPERTIES (10h)	10h	Saves the current configuration properties to FLASH (i.e. these values will be stored when the device is powered down).
LOAD_PROPERTIES (11h)	11h	Load the stored configuration properties.
READ_BLOCK	12h	Reads a single block of tag memory from a specified tag ID.
READ_BLOCKS	13h	Read multiple blocks of tag memory from a specified tag ID.
WRITE_BLOCK	15h	Write a single block of data to a specified tag ID.
WRITE_BLOCKS	16h	Write multiple blocks of data to a specified tag ID.
GET_TAG_INFO	18h	Returns information about the tag in the field.
LOCK_BLOCK	19h	Reserved for future use.
STREAM_READ	1Ah	Reserved for future use.
GET_STATUS	23h	Requests an information snapshot of the device's current status (includes battery level and hardware and firmware versions).
SET_SCANNING	32h	When the device is in continuous scanning mode, start or stop scanning for tags using the current connected mode.
SET_BT_PIN	40h	Sets the Bluetooth PIN.
SET_BT_NAME	42h	Assigns a Bluetooth device display name.
GET_BT_NAME	43h	Returns the Bluetooth device display name.
GET_ENTRY_COUNT	60h	Returns the number of tags stored on the device. Each tag record contains the tag ID, an optional timestamp and a single byte of data from the tag.
GET_ENTRY	61h	Retrieves a specific stored tag record from the device.
CLEAR_ENTRIES	62h	Deletes all stored tag information on the device.
FACTORY_RESET	74h	Resets all properties to factory defaults.
BEGIN_COMMANDS	80h	Informs the device that a connected host application is about to issue a series of RFID commands. The device then displays appropriate audible and visual feedback to the user.
END_COMMANDS	88h	Informs the device that a connected host application has

		finished issuing a series of RFID commands. The device then displays appropriate audible and visual feedback to the user.
POWER_DOWN	91h	Powers down the IDBLUE device.
BLUETOOTH_OFF	92h	Turns off the Bluetooth transmitter.
HEARTBEAT	96h	Command used for the heartbeat communication.
ENABLE_CHANNEL	97h	Enables or disables the logical communication channel.

Standard Responses

The table below lists the standard (non-command specific) responses that the IDBLUE device may return.

Command	Code	Description
NACK	1Fh	Non-acknowledgement that an error occurred in processing a communication packet.
ASYNC	70h	Notification that the following packet is an asynchronous notification.
BUTTON_PRESS	FFh	Asynchronous notification that the user has pressed the Action Button of the device.

NACK Responses

A NACK response is returned to the host to indicate errors during command processing or operation.

Header	Data Length	Data	Checksum
1Fh	Variable (2 bytes)	[command code (1 byte)] | [failure type (1 byte)] | [information (variable)]	1-byte

The failure type can be either a generic failure indication, or specific to certain operations. The failure indications are indicated in the following table, however it is expected that further failure types will be added and therefore any driver or application implementation should accommodate additions.

General Failure	Failure Code	Description
General Failure	01h	A general failure indication.
Fatal Error	02h	A fatal error has occurred.
Not Implemented	03h	A specified command is not implemented. Generally used for commands reserved for future use.
Timeout	04h	A timeout has expired while performing an operation.

Invalid Command	05h	The specified command is invalid.
Not Permitted	06h	The specified operation is not permitted.
Checksum Error	07h	A Checksum error occurred on the received packet.

Command Specific Failure	Failure Code	Description
Invalid Property	51h	An invalid property value was specified.
Invalid Value	52h	The specified property value is not of the correct type or within the allowed range.
Invalid Index	53h	An invalid index value was specified.
Tag Block Count Exceeded	54h	The number of blocks to read/write for the READ_BLOCKS and WRITE_BLOCKS commands exceeds the available blocks on the tag. Reserved for future use.
Buffer Overflow	55h	An internal IDBLUE buffer overflow has occurred.
Incomplete Operation	56h	The requested operation could not be completed.
Muted	57h	This indicates the Beep command was executed but no tone was emitted because the BuzzerEnabled property is disabled.
Duplicate Tag	58h	This indicates that the application has requested an RFID operation on a tag which has been accessed within the Duplicate Elimination timeout window.

Functions

The following sections describe the request and response formats for all commands. For each command, the request packet format is specified. The successful and any command specific failures are outlined below. **In addition, though not specified per command, a general failure can be returned for any command which can send a response.**

Core Functions

This section describes the non-RFID, non-configuration commands supported by the IDBLUE device.

NO_OP (00h)

This request does not affect the operation of the IDBLUE device, and is primarily used to verify a valid connection to an IDBLUE device.

Request format

Header	Data Length	Data	Checksum
00h	0000h	None	00h

Successful Response format

Header	Data Length	Data	Checksum
00h	0000h	None	00h

HEARTBEAT (96h)

This command is used for the communication protocol heartbeat between IDBLUE and the host application. This command does not reset IDBLUE's Device Timeout.

Request format

Header	Data Length	Data	Checksum
96h	0000h	None	96h

Successful Response format

Header	Data Length	Data	Checksum
96h	0000h	None	96h

GET_STATUS (23h)

This request returns status information about the device including battery level, firmware and hardware version information.

Request format

Header	Data Length	Data	Checksum
23h	0000h	none	23h

Successful Response format

Header	Data Length	Data	Checksum
23h	0008h	[battery level (1 byte)] | [major hardware version (1 byte)] | [minor hardware version (1 byte)] | [major firmware version (1 byte)] | [minor firmware version (1 byte)] | [branch firmware version (1 byte)] | [build firmware version (2 bytes)]	1 byte

Parameters

- **Battery level**; the remaining battery life of the device, expressed as a percentage (0 – 100%)
- **Hardware version**; the primary hardware version of the device (8 for R8) and the minor hardware version of the device.
- **Firmware version**; the major/minor/branch/build number of the firmware on the device (e.g. 4.0.0.342).

BEGIN_COMMANDS (80h)

This request signals the start of one or more reactive mode requests. The host should send this packet when preparing to issue a sequence of RFID-related commands to the device.

Request format

Header	Data Length	Data	Checksum
80h	0000h	None	80h

Successful Response format

Header	Data Length	Data	Checksum
80h	0000h	None	80h

When this command is issued and the device is in REACTIVE mode, the device sets the Action LED to Blue.

END_COMMANDS (88h)

This packet signals the end of one or more reactive mode requests. The host sends this packet when it has finished streaming commands to the IDBLUE device (see BEGIN_COMMANDS).

This request should only be issued if a BEGIN_COMMANDS request has already been sent to the device, or if the host has finished issuing a series of commands in response to a ButtonPress event. When the device receives this request, it provides user feedback based on the status value.

Request format

Header	Data Length	Data	Checksum
88h	0001h	[status (1 byte)]	1 byte

Parameters

- **Status**. The host uses this field to tell the IDBLUE device if the stream of commands it sent was successful, or unsuccessful. A value of 1 indicates success, 0 indicates failure. The host application should decide if the series of requests was successful or not.

Successful Response format

Header	Data Length	Data	Checksum
88h	0000h	None	88h

The LED on the device will glow green on success and red on failure, for 500 ms. If the audio buzzer is enabled, the device will also beep on success.

POWER_DOWN (91h)

This requests the device to turn off. It offers a method to shut down the IDBLUE device from a software application.

Request format

Header	Data Length	Data	Checksum
91h	0000h	None	91h

Response format

If the command is successfully received the device shuts down immediately, and no response is sent.

BEEP (03h)

This command causes the device to emit one of the standard audio tones, or beeps. If audio feedback is disabled (through the **BUZZERENABLED** property) the command will fail and no audio tone will be emitted. This command is intended for providing additional user feedback in a custom application.

Request format

Header	Data Length	Data	Checksum

03h	0001h	Beep type (1 byte)	1 byte

Parameters

The available beep types are:

- Low (0)
- High (1)
- High-to-Low (2)
- Low-to-High (3)
- Low-Low (4)
- High-High (5)

Successful Response format

Header	Data Length	Data	Checksum
03h	0000h	None	03h

Invalid Beep Type Failure Response format

Header	Data Length	Data	Checksum
1Fh	0002h	0352h	4Ch

Buzzer Muted Failure Response format

Header	Data Length	Data	Checksum
1Fh	0002h	0357h	49h

BLUETOOTH_OFF (92h)

This requests that the Bluetooth module on the device be turned off (immediately dropping the existing connection). The Bluetooth module can be turned back on by pressing and releasing the power button on the device.

Request format

Header	Data Length	Data	Checksum
92h	0000h	None	92h

Successful Response format

Header	Data Length	Data	Checksum
92h	0000h	None	92h

Note that in case of success when the device is connected via Bluetooth no response will be sent.

SET_SCANNING (32h)

When the device is in continuous scanning mode, start or stop scanning for tags using the current connected mode.[3]

Request format

Header	Data Length	Data	Checksum
32h	0001h	[mode (1 byte)]	1 byte

Parameters

- Mode value is 0 for stop scanning, any other value for start scanning.

Successful Response format

Header	Data Length	Data	Checksum
32h	0000h	None	32h

[3] The IDBLUE device will queue multiple SET_SCANNING commands in sequence. It is not recommended to send multiple sequential SET_SCANNING commands within the Continuous Scan Timeout interval.

Configuration Commands

This section details the commands used to manage the configuration of an IDBLUE device.

SET_PROPERTY (08h)

This command sets one of the configurable properties on the device. This property is set "in-memory" as long as the device remains powered up.

In order to remember this property, a SAVE_PROPERTIES command must be issued.

Request format

Header	Data Length	Data	Checksum
08h	Variable (2 bytes)	[property (2 bytes)] | [value (Variable)]	1 byte

Parameters

- **Property**. Property code value. Refer to the Configurable Properties section for a list of the available properties.
- **Value**. The value of the property to be set.

Successful Response format

Header	Data Length	Data	Checksum
08h	0002h	[property (2 bytes)]	1 byte

Parameters

- **Property**. Property code whose value was set.

Invalid Property Failure Response format

Header	Data Length	Data	Checksum
1Fh	Variable (2 bytes)	0851h | [invalid property (2 bytes) (optional)]	1 byte

Parameters

- **Invalid Property**. The property value supplied in the request which was invalid. It is possible that the property value cannot be determined from the request, in which case this parameter will not be present.

Invalid Value Failure Response format

Header	Data Length	Data	Checksum
1Fh	Variable (2 bytes)	0852h | [property (2 bytes) (optional)]	1 byte

Parameters

- **Property**. The property value supplied in the request which for which the supplied value is invalid. It is possible that the property value cannot be determined from the request, in which case this parameter will not be present.

GET_PROPERTY (09h)

This command retrieves the current setting for a given property.

Request format

Header	Data Length	Data	Checksum
09h	0002h	[Property (2 bytes)]	1 byte

Parameters

- **Property**. Property code whose value is to be retrieved.

Successful Response format

Header	Data Length	Data	Checksum
09h	Variable (2 bytes)	[property (2 bytes)] | [value (variable)]	1 byte

Parameters

- **Property**. Property code retrieved.
- **Value**. The value of the requested property.

Invalid Property Failure Response format

Header	Data Length	Data	Checksum
1Fh	0004h	0951h | [invalid property (2 bytes) (optional)]	1 byte

SAVE_PROPERTIES (10h)

This command saves the currently configured properties as the default configuration. This default configuration is written to non-volatile memory, and loaded when the device powers up. The session properties are configured using the SET_PROPERTY (08h) command.

Request format

Header	Data Length	Data	Checksum
10h	0000h	None	10h

Successful Response format

Header	Data Length	Data	Checksum
10h	0000h	None	10h

LOAD_PROPERTIES (11h)

This command loads the default configuration from non-volatile memory. This overwrites any of the current configuration settings.

Request format

Header	Data Length	Data	Checksum
11h	0000h	None	11h

Successful Response format

Header	Data Length	Data	Checksum
11h	0000h	None	11h

FACTORY_RESET (74h)

This command resets all of the configuration settings on the device to factory defaults. This includes the Bluetooth configuration parameters (name and PIN code).

Notes:

- This command will erase all stored tag data on the device.
- This command will reset the Bluetooth settings, causing any existing Bluetooth connection to drop. In this instance, no response will be received from the device.

For a list of the factory default settings, please refer to Appendix B.

Request format

Header	Data Length	Data	Checksum
74h	0000h	None	74h

Response format

If the command is successfully received the device resets and shuts down immediately, and no response is sent.

NOTE: Do not perform a hard reset during the factory reset process. Doing so can lead to the improper setting of some device settings.

SET_BT_PIN (40h)

This command assigns a PIN code (passkey) for Bluetooth connectivity. This command is reserved for future use and will return a Not Implemented NACK.

Request format

Header	Data Length	Data	Checksum
40h	Variable (2 bytes)	[pin (1 – 16 bytes)]	1 byte

Parameters

- **Pin**. The Bluetooth security PIN (also known as a passkey) for connecting to the IDBLUE device. The default PIN is "0000". The PIN is usually a series of ASCII characters, of length up to 16 characters.

Successful Response format

Header	Data Length	Data	Checksum
40h	0000h	None	40h

Invalid Value Failure Response format

Header	Data Length	Data	Checksum
1Fh	0002h	4052h	0Fh

WARNING: If an end-user modifies the PIN code from its factory default, and forgets the PIN code it will no longer be possible to connect to the device via Bluetooth.

The only way to undo this change will be to connect to the device via USB, and issue a FACTORY_RESET command.

If developing an application or interface that allows an end-user to modify the PIN code – you MUST display a warning to this effect.

SET_BT_NAME (42h)

Assign a new Bluetooth name to the IDBLUE device.

Request format

Header	Data Length	Data	Checksum
42h	Variable (2 bytes)	Name (0 – 16 bytes)	1 byte

Parameters

- **Name**. This is the display name of the IDBLUE device. This identifying name will be displayed when a host scans for Bluetooth devices. The name has a maximum length of 16 ASCII characters (bytes).

Successful Response format

Header	Data Length	Data	Checksum
42h	0000h	None	42h

GET_BT_NAME (43h)

Description

Retrieve the Bluetooth display name of the IDBLUE device.

Request format

Header	Data Length	Data	Checksum
43h	0000h	None	43h

Successful Response format

Header	Data Length	Data	Checksum
43h	Variable (2 bytes)	[name (1-16 bytes)]	1 byte

Parameters

- **Name**. This is the Bluetooth display name of the device, represented as a string of ASCII encoded characters. The length of the name can be no greater than 16 characters.

ENABLE_CHANNEL (97h)

Description

Enables or disables a logical communication channel with a host. This command must be used by the host application to indicate when the logical communication channel is to be established and dropped. Any other commands received by the IDBLUE device before enabling the logical channel will be ignored.

Request format

Header	Data Length	Data	Checksum
97h	Variable (2 bytes)	[mode (1 byte)]	1 byte

Successful Response format

Header	Data Length	Data	Checksum

97h	0000h	none	97h

Parameters

- **Mode**. Specifies whether to enable (1) or disable (0) the logical communication channel.

This command is new as of firmware version 4.1.0. When communicating over USB, IDBLUE will not respond to any other requests until this command is received. For backwards compatibility, the use of this command is not enforced when communicating over Bluetooth. However, IDBLUE does not utilize the heartbeat protocol unless it receives an ENABLE_CHANNEL request.

Stored Tag Functions

This section details the commands used to access the tag information stored on-board the IDBLUE device when operating in disconnected TAG_STORE mode.

GET_ENTRY_COUNT (60h)

Retrieve the number of tag records currently in the device's internal tag store.

Request format

Header	Data Length	Data	Checksum
60h	0000h	None	60h

Successful Response format

Header	Data Length	Data	Checksum
60h	0002h	[entry count (2 bytes)]	1 byte

Parameters

- **Entry Count**. An unsigned 16-bit integer representing the number of entries in the tag memory database. The MSB is sent first, followed by the LSB.

GET_ENTRY (61h)

Retrieve a single entry from the device tag memory database.

Request format

Header	Data Length	Data	Checksum
61h	0002h	Index (2 bytes)	1 byte

Parameters

- **Index**. An unsigned 16-bit integer representing the index of the entry to read from the tag memory data.

Successful Response format

Header	Data Length	Data	Checksum
61h	0011h	[timestamp (6 bytes)] | [tag id length (1 byte)] | [tag id (variable)] | [block data (1 byte)]	1 byte

Parameters

- **Timestamp**. The timestamp for this entry[4]. Uses the same format as returned by the Timestamp property.
- **Tag Id Length**. The length of the Tag ID of tag read.
- **Tag Id**. The tag id associated with this record.
- **Block data**. The first byte of the first block of the tag associated with this record.

Invalid Index Failure Response format

Header	Data Length	Data	Checksum
1Fh	0002h	6153h	2Fh

CLEAR_ENTRIES (62h)

Delete all of the stored tag records on the device[5].

Request format

Header	Data Length	Data	Checksum
62h	0000h	None	62h

Successful Response format

Header	Data Length	Data	Checksum
62h	0000h	None	62h

[4] If the year for this timestamp is 0000 then it is an invalid timestamp and should be disregarded. The device will only record invalid timestamps when the *RequireTimestamp* property is disabled.

[5] This physically erases the records from storage; as such this command may take up to 16 seconds to complete, and should be invoked asynchronously from host applications.

RFID Commands

This section describes all of the RFID-specific commands that are used to select tag ID's, and read and write on-board tag data.

Any failure (NACK) responses to an RFID command will include a timestamp in the optional *information* field. The general failure NACK format will be as below. All command specific failures are indicated in each of the following sections.

Header	Data Length	Data	Checksum
1Fh	0008h	[command code (1 byte)] | [failure type (1 byte)] | [timestamp (6 bytes)]	1-byte

Parameters

- **Command Id**. The code of the command which failed.
- **Failure Type**. An indication of the type of failure which occurred.
- **Timestamp**. Timestamp at which the RFID operation was attempted or failed.

GET_TAG_ID (01h)

Return the tag ID of the tag in the RF field[6].

Request format

Header	Data Length	Data	Checksum
01h	0000h	none	01h

Successful Response format

Header	Data Length	Data	Checksum
01h	Variable (2 bytes)	[timestamp (6 bytes)] | [tag id length (1 byte)] | [tag id (variable)]	1 byte

Parameters

- **Timestamp**. Time which the RFID operation took place.
- **Tag Id Length**. The length of the Tag ID of tag read.
- **Tag Id**. Unique tag identification number[7]. The least-significant byte (LSB) of the tag id is sent first.

[6] Tag data is not retrieved, only the unique tag ID.

Duplicate Elimination Failure Response format

This NACK is returned if a GET_TAG_ID request fails because a duplicate elimination timeout has been configured and the tag has been previously accessed with the duplicate elimination timeout window.

Header	Data Length	Data	Checksum
1Fh	0008h	0158h | [timestamp (6 bytes)]	1 byte

Parameters

- **Timestamp**. Time which the attempted RFID operation took place.

READ_BLOCK (12h)

Read a single block of memory from the specified tag.

Request format

Header	Data Length	Data	Checksum
12h	Variable (2 bytes)	[tag id length (1 byte)] | [tag id (variable)] | [block index (1 byte)]	1 byte

Parameters

- **Tag Id Length**. The length of the Tag ID of tag read.
- **Tag Id**. Tag ID of tag to read. (Use all zeros (0) to read any tag). The least-significant byte (LSB) of the tag id is sent first.
- **Block Index**. Index number of block in tag memory.

Successful Response format

Header	Data Length	Data	Checksum
12h	Variable (2 bytes)	[timestamp (6 bytes)] | [tag id length (1 byte)] | [tag id (variable)] | [block index (1 byte)] | [block data (variable)]	1 byte

Parameters

- **Timestamp**. Time which the RFID operation took place.
- **Tag Id Length**. The length of the Tag ID of tag read.
- **Tag Id**. Tag ID of tag read. The least-significant byte (LSB) of the tag id is sent first.

[7] When developing a driver library, the byte order for the tag ID should be reversed before passing the tag ID back to a user application (as the standard method for printing and displaying the tag ID is in MSB order).

- **Block Index**. Index number of block in tag memory.
- **Block Data.** Data contained at the specified block index.

Invalid Index Failure Response format

A general failure NACK is returned if the supplied index is not valid for the tag.

Header	Data Length	Data	Checksum
1Fh	0002h	1201h	0Eh

Duplicate Elimination Failure Response format

This NACK is returned if a READ_BLOCK request fails because a duplicate elimination timeout has been configured and the tag has been previously accessed with the duplicate elimination timeout window.

Header	Data Length	Data	Checksum
1Fh	0008h	1258h | [timestamp (6 bytes)]	1 byte

Parameters

- **Timestamp**. Time which the attempted RFID operation took place.

READ_BLOCKS (13h)

For supported tags, this command will read multiple blocks of tag memory as specified. The maximum number of bytes that IDBLUE can read in a single operation is 128 bytes, regardless of the block structure of the tag. It is recommended to use multiple READ_BLOCK(S) requests to read more than the 128 byte maximum.

Request format

Header	Data Length	Data	Checksum
13h	Variable (2 bytes)	[tag id length (1 byte)] | [tag id (variable)] | [block index (1 byte)] | [block count (1 byte)]	1 byte

Parameters

- **Tag Id Length**. The length of the Tag ID of tag read.
- **Tag Id**. Tag ID of tag to read. (Use all zeros (0) to read any tag). The least-significant byte (LSB) of the tag id is sent first.
- **Block Index**. Index number of block in tag memory from which to start reading.
- **Block Count.** Number of blocks of memory to read, beginning at start index.

Successful Response format

Header	Data Length	Data	Checksum
13h	Variable (2 bytes)	[timestamp (6 bytes)] | [tag id length (1 byte)] | [tag id (variable)] | [block index (1 byte)] [block count (1 byte)] | [block data (variable)]	1 byte

Parameters

- **Timestamp**. Time which the last RFID read operation took place.
- **Tag Id Length**. The length of the Tag ID of tag to read.
- **Tag Id**. Tag ID of tag to read. (Use all zeros (0) to read any tag). The least-significant byte (LSB) of the tag id is sent first.
- **Block Index**. Index number of block in tag memory from which to start reading.
- **Block Count.** Number of blocks of memory to read, beginning at start index.
- **Block Data.** Returned block data from the tag.

Tag Block Count Failure Response format

This NACK is returned if the read request block count exceeds the number of blocks on the tag.

Header	Data Length	Data	Checksum
1Fh	0008h	1354h | [timestamp (6 bytes)]	1 byte

Parameters

- **Timestamp**. Time which the last RFID read operation took place.

IDBLUE Buffer Overflow Failure Response format

This NACK is returned if an internal buffer overflow occurs.

Header	Data Length	Data	Checksum
1Fh	0008h	1355h | [timestamp (6 bytes)]	1 byte

Parameters

- **Timestamp**. Time which the last RFID read operation took place.

Incomplete Read Failure Response format

This NACK is returned if a read is incomplete. This can generally occur if the IDBLUE device is moved away from the tag before all requested blocks are read.

Header	Data Length	Data	Checksum
1Fh	0008h	1356h [timestamp (6 bytes)] [blocks unread (1 byte)]	1 byte

Parameters

- **Timestamp**. Time which the last RFID read operation took place.
- **Blocks unread**. The number of blocks it was unable to read.

Invalid Index Failure Response format

An Invalid Index failure NACK is returned if the supplied index is not valid for the tag or a value of 0 is passed for the block count.

Header	Data Length	Data	Checksum
1Fh	0002h	1353h	5Dh

Duplicate Elimination Failure Response format

This NACK is returned if a READ_BLOCKS request fails because a duplicate elimination timeout has been configured and the tag has been previously accessed with the duplicate elimination timeout window.

Header	Data Length	Data	Checksum
1Fh	0008h	1358h | [timestamp (6 bytes)]	1 byte

Parameters

- **Timestamp**. Time which the attempted RFID operation took place.

WRITE_BLOCK (15h)

For supported tags, this command will write one single block of tag memory as specified.

Request format

Header	Data Length	Data	Checksum
15h	Variable (2 bytes)	[tag id length (1 byte)] | [tag id (variable)] | [block index (1 byte)] | [block data (variable)]	1 byte

Parameters

- **Tag Id Length**. The length of the Tag ID of tag to write.
- **Tag Id**. Unique tag id number of the tag to modify (use all zeros (0) to write to any tag). The least-significant byte (LSB) of the tag id is sent first.
- **Block Index**. Index number of block in tag memory

- **Block Data**. Block data to be written.

Successful Response format

Header	Data Length	Data	Checksum
15h	Variable (2 bytes)	[timestamp (6 bytes)] | [tag id length (1 byte)] | [tag id (variable)] | [block index (1 byte)	1 byte

Parameters

- **Timestamp**. Time which the RFID operation took place.
- **Tag Id Length**. The length of the Tag ID of tag written.
- **Tag Id**. Unique tag id number of the tag written. The least-significant byte (LSB) of the tag id is sent first.
- **Block Index**. Index number of block in tag memory

Invalid Index Failure Response format

A general failure NACK is returned if the supplied index is not valid for the tag.

Header	Data Length	Data	Checksum
1Fh	0002h	1501h	09h

Duplicate Elimination Failure Response format

This NACK is returned if a WRITE_BLOCK request fails because a duplicate elimination timeout has been configured and the tag has been previously accessed with the duplicate elimination timeout window.

Header	Data Length	Data	Checksum
1Fh	0008h	1558h | [timestamp (6 bytes)]	1 byte

Parameters

- **Timestamp**. Time which the attempted RFID operation took place.

WRITE_BLOCKS (16h)

For supported tags, this command will write multiple blocks of tag memory as specified. The maximum number of bytes that IDBLUE can write is 128 bytes regardless of the block structure of the tag. It is recommended to use multiple WRITE_BLOCK(S) requests to write more than the 128 byte maximum.

Request format

Header	Data Length	Data	Checksum

16h	Variable (2 bytes)	[tag id length (1 byte)] | [tag id (variable)] | [block index (1 byte)] | [block count (1 byte)] | [block data (variable)]	1 byte

Parameters

- **Tag Id Length**. The length of the Tag ID of tag to write.
- **Tag Id**. Unique tag id number of the tag to modify (use all zeros (0) to write to any tag). The least-significant byte (LSB) of the tag id is sent first.
- **Block Index**. Index number of block in tag memory at which to start writing.
- **Block Count**. Number of blocks of memory to write, starting from *start index*.
- **Block Data**. Block data to be written.

Successful Response format

Header	Data Length	Data	Checksum
16h	Variable (2 bytes)	[timestamp (6 bytes)] | [tag id length (1 byte)] | [tag id (variable)] | [block index (1 byte)] | [block count (1 byte)]	1 byte

Parameters

- **Timestamp**. Time which the last RFID write operation took place.
- **Tag Id Length**. The length of the Tag ID of tag written.
- **Tag Id**. Unique tag id number of the tag written. The least-significant byte (LSB) of the tag id is sent first.
- **Block Index**. Index number of block in tag memory which was written.
- **Block Count**. Number of blocks of memory wrote, starting from *start index*. It is possible that a partial write can occur, in which case the block count in the response indicates the number of blocks actually written and may be less than the number of blocks requested.

Tag Block Count Failure Response format

This NACK is returned if the read request block count exceeds the number of blocks on the tag.

Header	Data Length	Data	Checksum
1Fh	0008h	1654h | [timestamp (6 bytes)]	1 byte

Parameters

- **Timestamp**. Time which the last RFID read operation took place.

IDBLUE Buffer Overflow Failure Response format

This NACK is returned if an internal buffer overflow occurs.

Header	Data Length	Data	Checksum
1Fh	0008h	1655h | [timestamp (6 bytes)]	1 byte

Parameters

- **Timestamp**. Time which the last RFID read operation took place.

Incomplete Write Failure Response format

This NACK is returned if a write is incomplete. This can generally occur if the IDBLUE device is moved away from the tag before all requested blocks are written.

Header	Data Length	Data	Checksum
1Fh	0008h	1656h [timestamp (6 bytes)] [blocks unwritten (1 byte)]	1 byte

Parameters

- **Timestamp**. Time which the last RFID read operation took place.
- **Blocks unwritten**. The number of blocks it was unable to write.

Invalid Index Failure Response format

An Invalid Index failure NACK is returned if the supplied index is not valid for the tag or 0 is passed as a block count value.

Header	Data Length	Data	Checksum
1Fh	0002h	1653h	58h

Duplicate Elimination Failure Response format

This NACK is returned if a WRITE_BLOCKS request fails because a duplicate elimination timeout has been configured and the tag has been previously accessed with the duplicate elimination timeout window.

Header	Data Length	Data	Checksum
1Fh	0008h	1658h | [timestamp (6 bytes)]	1 byte

Parameters

- **Timestamp**. Time which the attempted RFID operation took place.

GET_TAG_INFO (18h)

This request will return information about the tag currently in the IDBLUE device read field. It will return the unique tag identifier, block size, and block count (the number of accessible blocks) for the tag.

Request format

Header	Data Length	Data	Checksum
18h	Variable (2 bytes)	[tag id length (1 byte)] | [tag id (variable)]	1 byte

Parameters

- **Tag Id Length**. The length of the Tag ID of tag scanned.
- **Tag Id**. Unique tag id number of the tag to scan (use all zeros (0) to use any tag). The least-significant byte (LSB) of the tag id is sent first.

Successful Response format

Header	Data Length	Data	Checksum
18h	Variable (2 bytes)	[timestamp (6 bytes)] | [tag id length (1 byte)] | [tag id (variable)] | [block size (1 byte)] | [block count (1 byte)]	1 byte

Parameters

- **Timestamp**. Time which the RFID operation took place.
- **Tag Id Length**. The length of the Tag ID of tag scanned.
- **Tag Id**. Unique tag id number of the tag scanned. The least-significant byte (LSB) of the tag id is sent first.
- **Block Size.** Size of each data block in bytes.
- **Block Count**. Total number of blocks on the tag.

Duplicate Elimination Failure Response format

This NACK is returned if a GET_TAG_INFO request fails because a duplicate elimination timeout has been configured and the tag has been previously accessed with the duplicate elimination timeout window.

Header	Data Length	Data	Checksum
1Fh	0008h	1858h | [timestamp (6 bytes)]	1 byte

Parameters

- **Timestamp**. Time which the attempted RFID operation took place.

Configurable Properties

All of the configurable behavior of the IDBLUE device is managed through the use of device properties. These properties are configured via the **GET PROPERTY**, **SET PROPERTY**, **LOAD PROPERTIES AND SAVE PROPERTIES** commands.

The supported properties are listed in the table below, along with their property codes, value format, description, and allowable/default values.

Property	Code	Type	Description
Continuous scanning	0000	Byte	Whether or not continuous scanning mode is enabled. Refer to section Operating Modes for details on continuous scanning mode. **Format:** Disabled (00h) Enabled (01h)
Timestamp Required	0001	Byte	Whether or not a valid timestamp (i.e. the timestamp property has been set since the device has been turned on) is required when storing tag ID's in disconnected mode. If this property is enabled, and the user attempts to scan a tag in disconnected mode without setting a timestamp, the device's front LED will flash and refuse to read the tag (see UI Feedback table for details). If the property is disabled, tags will be stored without a valid timestamp (i.e. for scenarios wherein a timestamp is unnecessary). **Format:** Disabled (00h) Enabled (01h)
Duplicate Elimination Timeout	0002	UInt16	When scanning tags in disconnected mode, this property specifies the timeout (in milliseconds) between reads that duplicate scans of the same tag will be ignored. **Format:** 0 - no timeout 100 - 100 ms Maximum value is 10,000 (10 seconds).
Time	0003	Special	The timestamp stored on the device. When the device first turns on, the device will not know what time it is and will start tracking time from 0000-00-00 00:00:00. Once this property is set, the device will keep track of time until it is turned off. Note: if the year of the timestamp returned from this property is 00, the

			timestamp may be safely assumed to be invalid. [year (1 byte)] | [month (1 byte)] | [day (1 byte)] | [hour (1 byte)] | [minute (1 byte)] | [second (1 byte)] Note: The time can be set to invalid values (i.e. month = 13, day = 32) when set through the API. Please use caution when setting the time property using this method.
Disconnected Mode	0004	Byte	Configures the disconnected (i.e. **not** connected to a host via USB or Bluetooth) operating mode of the IDBLUE device. The valid operating modes are: **Tag Verify (00h)** Allows users to verify if a tag is compatible with IDBLUE using the currently configured RFID protocol. **Store Timestamp + Tag Id + Block 0 Data (01h)** Reads the tag id and the first byte of block 0 and stores this information, along with a timestamp, in onboard memory. This data may be retrieved using the GET_ENTRY command.
Connected Mode	0005	Byte	Configures the connected (i.e. connected to a host via USB or Bluetooth) operating mode of the IDBLUE device. The valid modes are listed to the right, with additional information on operating modes available in section Operating Modes. **Normal (00h or 03h)** Performs a GET_TAG_ID operation, attempting to read the Id of whatever tag is in the RF field. **Reactive (01h)** Sends an event to the host when the button is pressed. This allows the host to control what operations will occur on a button press. **Read Block N (04h)** Performs a READ_BLOCK operation on the block index specified in the Block Index property. Uses the Block Index property to determine the block to read. **Write Block N (05h) **** Performs a WRITE_BLOCK BLOCK operation on the block index specified in the Block Index property using data from the Block Data property. Uses the Block Index and Block Data properties. **Read Blocks (06h) **** Performs a READ_BLOCKS operation, starting at the block index specified

			in the Block Index property, retrieving the number of blocks specified in the Block Count property.
RFID Protocol	0006	Byte	The currently selected RFID Protocol. The supported RFID Protocols and their command code values are listed below. More information about the supported RFID protocols is available in section Supported Protocols. ISO15693 (01h)
Buzzer	0007	Byte	Whether or not the on-board audio buzzer is enabled. Disabled (00h) Enabled (01h)
Device Timeout	0008	Byte	The idle time (in minutes) before the device shuts down. A value of 0 means that the device will not shut down automatically. The idle time is the amount of time since the last action button press, or connected command sent from a host. No timeout (0) Timeout (1-255)
RFID Timeout	0009	Byte	The time in seconds before an RFID operation (select tag, read block, etc) times out. The timer begins as soon as the RFID operation begins and the operation will fail (timeout) if the timeout elapses before the operation is complete. Min value: 1 second Max value: 16 seconds
Bluetooth Timeout	000A	Byte	The time in minutes before the Bluetooth link times out and shuts down. Note: this only applies when the device does not have an active Bluetooth connection. In order to re-enable the Bluetooth link after it has been disabled, the user must press and release the power button. At this point, the Bluetooth link on the device will be re-enabled. 1-255 minutes. 0 = no timeout
Continuous Scanning Timeout	000B	Byte	In continuous scanning mode, this property controls how long the device will “scan” without finding a tag before ceasing continuous operations. For more information on how the continuous scanning mode operates, refer to the section Operating Modes. 1-255 seconds 0 = no timeout

Block Index	000C	Byte	For connected modes **Read Block N** and **Write Block N** this property configures which block index is used for the operation. For connected mode **Read Blocks** this property configures the starting block index. Valid values: 0-255
Block Data	000D	Byte Array	For connected mode **Write Block N** this property sets the block data used for the operation. The maximum length of this array is 8 bytes.
Block Count	000E	Byte	For connected mode **Read Blocks**, this property configures the block count (i.e. number of blocks read). Valid values: 1-255 Note: the block index value should be set prior to setting the block count, and the block count correctly calculated to ensure that read or write operations will not exceed the number of blocks available on the tag.
Version Information	0017	Byte Array	Extended versioning information. Contains full major/minor/build information for both hardware and firmware versions, as well as a version comment string. This is a **read-only** property. Format: [hardware major (1 byte)] [hardware minor (1 byte)] [firmware major (1byte)] [firmware minor (1 byte)] [firmware branch (1 byte)] [firmware build (2 bytes)] [description (variable)]
Bootloader version	0018	Byte array	Bootloader version information. This is a read-only property. Format: [major version] (1 byte)] [minor version] (1 byte)] [branch version] (1 byte)] [build version] (2 bytes)]
Hold To Scan	0019	Boolean	Specifies whether Hold To Scan is enabled. Disabled (00h) Enabled (01h)

Appendix A – Checksum Generation

Current versions of IDBLUE devices use a simple XOR checksum algorithm. The checksum is generated by XOR'ing all bytes in a request/response packet, including header, length, and data. The result of this operation is stored in a 1 byte (or 8 bit) field and attached to the end of the packet.

Below is a request packet for reading block 5 of a specific RFID tag.

Header	Length	Data
12h	00h 09h	99h 01h 0h 00h 07h 00h 00h E0h 05h

The checksum is calculated by XOR'ing the bytes in the packet:

12h ^ 00h ^ 09h ^ 99h ^ 01h ^ 00h ^ 00h ^ 07h ^ 00h ^ 00h ^ E0h ^ 05h = 61h

This value is attached to the end of the packet and the final request packet is:

Header	Length	Data	Checksum
12h	00h 09h	99h 01h 00h 00h 07h 00h 00h E0h 05h	61h

Appendix B – Factory Default Configuration

The table below lists the default factory settings.

Property	Value
Block Count	1
Block Data	00 00 00 00
Block Index	0
Bluetooth Name	IDBLUE_R8<last 4 characters of Bluetooth MAC address)
Bluetooth PIN Code	0000
Bluetooth Timeout	0 (disabled)
Buzzer	Enabled (01)
Connected Mode	Tag ID (03)
Continuous Scanning	Disabled (00)
Continuous Scanning Timeout	16 seconds
Device Timeout	10 minutes
Disconnected Mode	Tag Verify (00)
Duplicate Elimination Timeout	0 (no duplicate elimination)
Hold to Scan	Enabled (01)
IDBLUE Name	IDBLUE_R8.HF
RFID Protocol	ISO15693
RFID Timeout	4 seconds
Timestamp Required	Enabled (01)

Table 4 : Factory Default Configuration