72-20PJ11-03 Rev 04

Software Interface Specification, Project GhostBuster

This Document describes the Software Interface between the GhostBuster Backpack microprocessor and the HERO Application on a remote computing platform.

Software Interface Specification, Project GhostBuster

Product Resources

4 Mulliken Way

Newburyport, MA 01950

Revision Control

|  |  |  |  |
| --- | --- | --- | --- |
| Revision | Description | Author | Date |
| 01 | Initial Release | PRI | 20 Aug 2020 |
| 02 | Detail Draft Release | PRI + Aviton | 30 Oct 2020 |
| 03 | Revised per Internal Comments | PRI + Aviton | 3 Nov 2020 |
| 04 | Further revision per internal comments | PRI + Aviton | 4 Nov 2020 |
|  |  |  |  |

Table of Contents

[1. Overview and Objective 3](#_Toc48832814)

[2. Definitions and Acronyms 3](#_Toc48832815)

[3. References 3](#_Toc48832816)

[4. Specification 4](#_Toc48832817)

[4.1. Communications Protocol 4](#_Toc48832818)

[4.2. Firmware Update 4](#_Toc48832819)

[4.3. Commands 4](#_Toc48832820)

[4.4. Error Codes 4](#_Toc48832821)

List of Figures

**No table of figures entries found.**

List of Tables

[Table 1 - Commands 4](#_Toc48832801)

[Table 2 - Error Codes 5](#_Toc48832802)

# Overview and Objective

#### The GhostBuster product will be used to spray disinfectant in and around furniture, equipment, and rooms where personnel or the public may be located. The design will allow for the efficient and even application of disinfectant while making a record of the activity to document the application of the disinfectant for quality purposes. The feature that sets the GhostBuster system apart from competitors is the optional HERO application which connects to the GhostBuster via Bluetooth Low Energy (BLE). The HERO application can then control and log data from the GhostBuster as well as update the GhostBuster firmware.

# Definitions and Acronyms

#### Identify terms and abbreviation that are not obvious and define how they are used in this document.

| Term | Definition |
| --- | --- |
| GhostBuster | The first product consisting of a backpack holding a tank of disinfectant which is sprayed onto items by means of a pump, atomizing nozzle and high voltage electrostatic dispersal system. |
| ESV | Electrostatic Voltage |
|  |  |
|  |  |
|  |  |
|  |  |

# References

|  |  |  |
| --- | --- | --- |
| Document | Title | Author |
| 72-20PJ10-01 | Product Requirements Document, Project GhostBuster | PRI |
| 72-20PJ10-02 | Software Requirements Document, Project GhostBuster | PRI |
|  |  |  |
|  |  |  |

# Specification

## Pairing Workflow

A Ghostbuster user will make the user available for pairing by pressing the pairing button on the Wand. At that point, the user shall be able to find the Ghostbuster unit available for pairing in the HERO app.

## Communications Protocol

All commands are sent by Text String of ASCII code. See the command table in section 4.3. Each command string is terminated with a Carriage Return(\r, 0x0D).

Example: (App -> GhostBuster): getSerial\r

PRI Note: Depending on how the HERO app uses the data from the Ghostbuster (especially the data related to flow rates or pumped volume), there may be a need to add some sort of message ID or ACK/NACK system to the protocol. To be discussed with the SW team. Also, Is there any reason to add authentication strings or encryption to the protocol to prevent communication with non-authorized users?

Aviton Note: I think the Commands act as the function of message ID and the replied data act as ACK, so they don’t need extra message ID and ACK. The BLE already has build-in encryption if enable MITM protection.

## Firmware Update

The HERO Application may initiate an update of a connected GhostBuster firmware. The mechanisms for performing this update are described in this section.

PRI Note: Mechanisms of this need to be further fleshed out.

## Accessible Variables in Ghostbuster Non-Volatile Memory

| Ref | Variable Name | Data Type (Unit / Format) | Description |
| --- | --- | --- | --- |
|  | SerialNum | String | Unit Serial Number  Aviton: For the String data type, we suggest to fix the length of char in each variable. |
|  | ModelID | String | Unit Model Identification |
|  | TotalPumpActiveTime | Int32 (seconds) | Amount of time the Pump has been active |
|  | TotalPumpedVolume | Int32 (mL) | Total gallons pumped |
|  | FWVersion | String | Unit firmware version |
|  | UnitName | String | Unit Name (for HERO app identification) |
|  | HWVersion | String | Unit Hardware Version |
|  | BatteryVersion | String | Battery HW Version (Fixed value, set at factory only) |
|  | BatteryCapacity | Int16 (mAHrs) | Battery Capacity, in mAHrs (Fixed Value, set at factory) |

## Accessible Variables in Volatile Memory

| Ref | Variable Name | Data Type (Unit / Format) | Description |
| --- | --- | --- | --- |
|  | PumpRunState | uInt8  0: Disable  1: Enable  Others: Error States TBD | Whether the pump is currently Active. Default is 0 at startup. |
|  | PumpFlowRate | Int32 (mL/hr) | Pump flow rate, in mL/hr (not a measured value). Reads 0 when pump is inactive. |
|  |  |  |  |
|  | ESVEnabledState | uInt8  0: Disabled  1: Enabled  Others: Error States TBD | Whether ESV is currently enabled. Default is 1 at startup. |
|  | ESVActivityState | uInt8  0: Inactive  1: Active  Others: Error States TBD | Whether ESV is currently active. Default is 0 at startup. |
|  |  |  |  |
|  | BatteryLevel | Int16 (% of total capacity) | Percentage of total battery capacity remaining |
|  | TriggerLatchState | uInt8  0: Latch Not Engaged  1: Latch Engaged | State of Trigger Latch. Default is 0 at startup. |
|  |  |  |  |
|  | SprayLockoutState | uInt8  0: Lockout Disabled (system can spray)  1: Lockout Enabled (system cannot spray) | Describes whether the Ghostbuster is permitted to spray or not. Default is 0 at startup.  Note: this is not anticipated to be used for the Betas or even at Product launch – this is more intended for future iterations in which the HERO app authenticates the chemistry being sprayed. |
|  | TriggerLatchMode | uInt8  0: Engaging latch prevents unit from spraying while trigger is pulled  1: Engaging latch causes unit to spray continuously from one trigger pull until next trigger pull | Describes trigger latch button behavior. Default is 0 at startup.  . |

## Commands

The GhostBuster will respond to queries or commands over the communications channel. The commands and descriptions are in Table 1 - Commands.

| Ref | Command | Description | Argument |
| --- | --- | --- | --- |
|  | getSerial | Return unit serial number (returns variable SerialNum) | n/a |
|  | getModel | Return unit model number (returns variable ModelNum) | n/a |
|  | getPumpTime | Return unit pump active time, in hrs (returns variable TotalPumpTime) | n/a |
|  | getPumpedVolume | Return unit total gallons pumped (returns variable TotalPumpVolume) | n/a |
|  | getFirmware | Return unit firmware version (returns variable FWVersion) | n/a |
|  | getUnitName | Returns unit name (returns variable UnitName) | n/a |
|  | getHWVersion | Returns unit HW version (returns variable HWVersion) | n/a |
|  |  |  |  |
|  | updateFirmware | Update firmware (methodology TBD – See section 4.2) | TBD |
|  | setUnitName | Sets the name of the Ghostbuster Unit (changes UnitName) | String (new Unit Name) |
|  | getTriggerLatchState | Returns the Ghostbuster Trigger Latch State | n/a |
|  | setTriggerLatchState | Allows the HERO app to engage the Trigger Latch (changes TriggerLatchState) | String (new Latch state) |
|  | getTriggerLatchMode | Returns the Ghostbuster Trigger Latch Mode | n/a |
|  | setTriggerLatchMode | Changes Trigger Latch Mode (variable TriggerLatchMode) | String (new latch mode) |
|  | resetPump | Resets the pump active time and pumped volume variables to 0. Should only be used for service / factory applications where the pump has been replaced | n/a |
|  |  |  |  |
|  | getPumpState | Return pump state (Returns variable PumpRunState) | n/a |
|  | getESVState | Return ESV state (on/off) (Returns variable ESVEnabledState) | n/a |
|  | getFlowRate | Return Spray Flow (returns variable PumpFlowRate) | n/a |
|  | getBatteryLevel | Return battery level (returns variable BatteryLevel) | n/a |

Table 1 - Commands

## BLE Profile

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Name | Type | UUID | Properties | Notes |
| System Statues | Service | 0xD973F2E0-B19E-11E2-9E96-0800200C9A66 | N/A |  |
| Characters Write | Characteristics | 0xD973F2E1-B19E-11E2-9E96-0800200C9A66 | Notify Write | Characters will be sent from Server(Product Unit) to Client(HERO App) |
| Characters Read | Characteristics | 0xD973F2E2-B19E-11E2-9E96-0800200C9A66 | Notify Read | Characters will be sent from Client(HERO App) to Server(Product Unit |

## Command Detail Examples

Get Serial Number

(App -> GhostBuster): getSerial\r

(GhostBuster -> App): 2020-000001\r

Format of serial number: Year(4-digits)-Sequence(6-digits)

Sequence: Assign in factory, decimal number start with 1, reset every year.

Get Model Number

(App -> GhostBuster): getModel\r

(GhostBuster -> App): TBD\r

Aviton comment: ‘GhostBuster’ may have copyright issue

Get total pump time

(App -> GhostBuster): getPumpTime\r

(GhostBuster -> App): [value of TotalPumpActiveTime]\r

Example: 36000\r for a unit that has had the pump active for 10 hours.

Get total pumped volume

(App -> GhostBuster): getPumpedVolume\r

(GhostBuster -> App): [value of TotalPumpedVolume]\r

Note: This is a calculated value, based on the product of TotalPumpActiveTime and the nominal pump flow rate. It is not a measured quantity.

Get firmware version

(App -> GhostBuster): getFirmware\r

(GhostBuster -> App): 1.00\r

Example: 1.00 is version 1.00.

Get Unit Name

(App -> GhostBuster): getUnitName\r

(GhostBuster -> App): 2020-000001\r

Note: Factory Default Unit name will be the same as the serial number.

Update firmware

(App -> GhostBuster): updateFirmware\r

(GhostBuster -> App): TBD\r

GhostBuster will enter firmware re-programming mode, this function will be development after Beta prototype.

Lock Unit

(App -> GhostBuster): sprayDisable\r

(GhostBuster -> App): OK\r

Note: This feature is intended for future use cases in which the HERO app is required to authenticate the chemistry before an operator can spray.

Unlock Unit

(App -> GhostBuster): sprayEnable\r

(GhostBuster -> App): OK\r

Set new Unit Name

(App -> GhostBuster): setUnitName ‘New Unit Name’\r

(GhostBuster -> App): OK\r

Toggle Trigger Latch Mode

(App -> GhostBuster): setTriggerLatchMode 0\r

(GhostBuster -> App): OK\r

Aviton suggests to add.

Reset total pump time and volume

(App -> GhostBuster): resetTotalTimeVolume\r

(GhostBuster -> App): OK\r

Get current pump status

(App -> GhostBuster): getPumpState\r

(GhostBuster -> App): On\r or Off\r

Example: On means the pump currently turns on.

Get current ESV selection status

(App -> GhostBuster): getESVState\r

(GhostBuster -> App): On\r or Off\r

Example: On means the ESV selection is enabled, it will be turned on when pump is running.

(App -> GhostBuster): getFlow\r

(GhostBuster -> App):[value of PumpFlowRate]\r

Note: This is not necessarily a measured value. It may simply be 0 when the pump is inactive, and a nominal flow rate when the pump is active.

Get current battery level

(App -> GhostBuster): getBatteryLevel\r

(GhostBuster -> App): 100\r to 000\r

Example: 085 is 85%