Bluetooth Addendum to Extended Serial Protocol

User's Guide

Revision 4, December, 2013

Thank you for choosing Valentine Research, Inc. ("Valentine", "we," "us," or "our") and our ESP Serial Protocol (the “Software”) and hardware related to the ESP Serial Protocol (the “Hardware) for use with Valentine’s VALENTINE ONE® radar detectors. Please review this License Agreement (the “License”) carefully before using the Software and Hardware as described herein. If you do not agree to the terms of this License, do not use the Software or Hardware.

1. General License. Valentine grants you a limited, non-exclusive, non-transferable, royalty-free, worldwide license to download, install, and use the Software or Hardware and any trademarks of Valentine, only as described in this License. The terms of this License will also govern any upgrades to the Software or Hardware, unless such an upgrade is accompanied by a separate license.

2. Permitted License Uses and Restrictions. You may use the Software and Hardware to create third party applications or software designed to be used with or including the Software and/or Hardware, or with Valentine’s products that use the Software or Hardware as a serial protocol. You may not copy, decompile, reverse engineer, disassemble, modify, or create derivative works of the Software or Hardware. You may not make the Software available over a network where it could be used by multiple computers or available to others for downloading and installation. The Software shall not be downloaded, exported or re-exported in violation of any law, including without limitation, any country subject to any applicable U.S. restrictions. The Hardware shall not be exported, or re-exported in violation of any law, including without limitation, any country subject to any applicable U.S. restrictions.

3. Intellectual Property. The Software and any associated proprietary code, inventions, or patents resulting from the Software or Hardware, and all trademarks and trade dress of Valentine remain the property of Valentine. By accessing the Software, you acquire no ownership rights to the Software or Hardware. In the event you use any trademarks of Valentine, you agree to do so only in the manner described in this License. Specifically, you agree only to use the trademarks of Valentine to describe any application or software you create using the Software or Hardware as “Works in connection with Valentine’s ESP Serial Protocol” or similar statement attributing ownership of any trademarks or the Software and Hardware to Valentine. You agree not to do anything that would compromise Valentine’s rights in and to the Software, Hardware, or Valentine’s trademarks.

4. Quality Assurance. Upon request, and with no obligation to return, you agree to provide suitable specimens of any application, software, or other protocol you create using the Software or Hardware to Valentine to verify your compliance with this License.

5. You agree to allow Valentine to use your name and/or product name in an index of products that work with the Software and Hardware and Valentine’s products.

6. AS IS Basis. THE SOFTWARE AND HARDWARE ARE PROVIDED TO YOU ON AN “AS IS” BASIS, AND YOU ARE SOLELY RESPONSIBLE FOR THEIR USE. VALENTINE DISCLAIMS ALL WARRANTIES REGARDING THE SOFTWARE AND HARDWARE, INCLUDING WARRANTIES OF NON-INFRINGEMENT. VALENTINE IS NOT LIABLE FOR ANY DIRECT, INDIRECT, INCIDENTAL, SPECIAL, CONSEQUENTIAL, OR PUNITIVE DAMAGES ARISING FROM OR RELATING TO YOUR USE OF THE SOFTWARE OR HARDWARE.

7. Indemnification. You agree to defend, indemnify, and hold harmless Valentine against any and all loss, liability, claims, causes of action, or suits arising out of or related to your use of the Software or Hardware, or any applications, software, or other protocols you create using the Software or Hardware, whatever their nature.

8. Support. Valentine has no obligation to provide any type of support for the Software or Hardware, or for any applications, software, or protocols you create using the Software or Hardware.

9. Termination. Valentine may terminate this License immediately, without notice to you. This License shall also terminate automatically upon your breach of any of the terms and conditions of this License. In the event of termination of this License, you will immediately cease all use of Valentine’s trademarks, and the Software.

10. No Assignment. You may not transfer or assign this License or the Software.

11. Miscellaneous. Any dispute arising under this License shall be resolved exclusively by the state and federal courts of the State of Ohio, applying Ohio law without regard to any conflict of law provisions.

‘V1connection’ vs ‘V1conection LE’

The V1connection communicates with an Android device using mobile device using Bluetooth 2.0+EDR. The V1connection uses a special packet wrapper to communicate with the Android device.

The V1connection LE communicates with an iOS device using Bluetooth 4.0 Low Energy. The V1connection LE does not use a packet wrapper when communicating with the iOS device.

With the exception of the packet wrapper and underlying Bluetooth specification differences, the V1connection and V1connection LE communicate with a mobile device the same way. **Unless the difference is specified, descriptions of the operation of the V1connection in this document also apply to the V1connection LE.**

Overview

The addition of a V1connection to the Extended Serial Protocol (ESP) bus allows a wireless device to communicate through a Bluetooth connection with the Valentine One (V1) and accessories on the ESP bus. Figure 1 shows a typical ESP-Bluetooth connection with the V1 being the only other device on the wired ESP bus. Other accessories may be added to the wired ESP bus as long as the accessory has an accessory socket (i.e. SAVVY) to accommodate the V1connection. The V1connection in Figure 1 will act as a wire extender for the ESP bus. Any valid ESP packet received on the wired ESP bus will be transmitted on the Bluetooth connection. Similarly, any valid ESP packet received on the Bluetooth connection will be placed on the wired ESP bus by the V1connection in the appropriate time slice. This addendum describes the interaction between the V1connection and the mobile device on the Bluetooth connection.

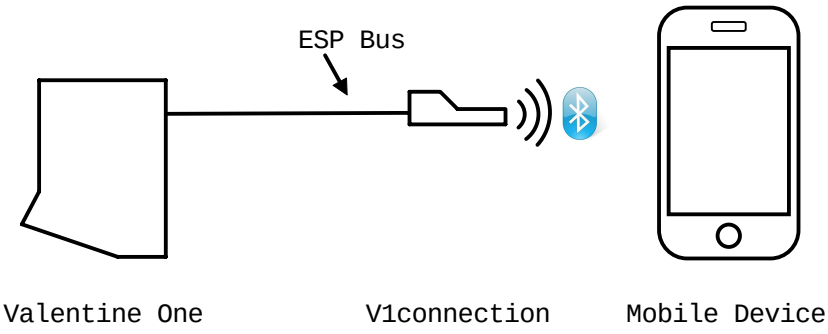

Figure 1 - A typical Bluetooth ESP connection

Device Identification

The device identifier used in the mobile device application will determine the ESP time slice utilized by the V1connection on the ESP bus, so the application creator must choose an appropriate device identifier.

Bluetooth Packet Confirmation

When the V1connection receives a packet on the Bluetooth connection it will place the packet in a buffer. Buffered packets are transmitted on the wired ESP bus on the next time slice that matches the origin identifier in the packet. After a packet is transmitted on the wired ESP bus, the V1connection will echo the packet on the Bluetooth connection as confirmation that the packet has been sent. This confirmation is the only way the application on the mobile device knows that the packet was sent out on the ESP bus.

Bluetooth ESP Device Requirements

The V1connection acts as a pass-through device for the ESP. Any valid ESP packets that are received on the wired ESP bus will be transmitted on the Bluetooth connection. With the exception of the version request discussed below, any valid ESP packet received on the Bluetooth connection will be placed on the wired ESP bus in the appropriate time slice.

Reading the V1connection Firmware Version

Only the mobile device is allowed to request the version of the V1connection. To request the version of the V1connection, the mobile device must send a r*eqVersion* packet with the same originator and destination identifier. When the originator and destination device identifiers are the same, the V1connection will recognize the request and return its version number using a *respVersion* packet with the same originator and destination device identifier. When the mobile device is requesting the version from the V1connection, neither the *reqVersion* or *respVersion* are placed on the wired ESP bus.

Bluetooth Error Handling

There may some instances when the V1connection will discard a packet received on the Bluetooth connection instead of transmitting it on the wired ESP bus. This will happen if the packet is not valid or the V1connection has a full buffer. When the V1connection discards a packet from the mobile device, it will notify the mobile device using the *respRequestNotProcessed* packet with the originator and destination device identifiers the same as the originator device identifier of the packet being discarded. In this scenario, the *respRequestNotProcessed* packet is not placed on wired bus. Errors on the wired ESP bus will be handled as described in the ESP specification.

Identifying a Legacy Valentine One

The V1connection will automatically switch between ESP and Legacy mode based on the mode the Valentine One is operating in. The V1connection will convert the Legacy display data into an infDisplayData ESP packet. The new packet will modify the Aux 0 byte of the infDisplayData packet to indicate that the data is generated from a Valentine One operating in Legacy Mode. When using a V1connection, the new definition of the Aux 0 byte in the infDisplayData packet is

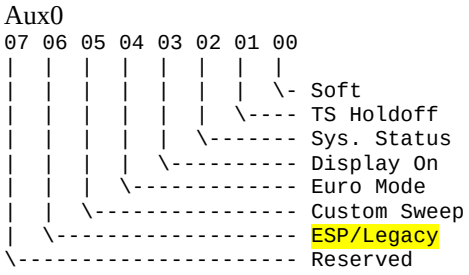

Table 1 - Bit descriptions for *infDisplayData* packet's Aux0 byte when using the V1connection

Bit #	Bit Name	Bit Description
0	Soft	This bit describes the current mute status: 0 - indicates the audio is not muted. 1 - indicates the audio is muted.
1	TS Holdoff	This bit tells the accessories if a time slice is allowed. 0 - indicates that all accessories are allowed to have a time slice following this packet. 1 - indicates that none of the accessories are allowed to have a time slice following this packet.
2	System Status	This bit describes the Valentine One status. 0 - indicates that the Valentine One is not actively searching for alerts. 1 - indicates that the Valentine One has successfully signed on and is actively searching for alerts.
3	Display On	This bit describes the status of the Valentine One display. 0 - indicates the Valentine One has turned off the main display. 1 - indicates the main display is turned on.
4	Euro Mode	This bit describes the Valentine One European Mode status. 0 - indicates the Valentine One is not operating in Euro Mode. 1 - indicates the Valentine One is operating in Euro Mode.
5	Custom Sweep	This bit describes the custom sweep status. 0 - indicates that custom sweeps have not been defined. 1 - indicates that custom sweeps have been defined and custom modes will be used if operating in Euro Mode.
6	ESP/Legacy	This bit describes the current ESP bus mode. 0 – indicates that the Valentine One is operating in ESP Mode 1 – indicates that the Valentine One is operating in Legacy Mode
7	Reserved	This bit has been reserved for future use

Android Packet Description

This section does not apply to the V1connection LE.
The V1connection uses a special packet format to communicate with an Android Bluetooth device. This packet will make it easier to determine when a complete packet is received over a Bluetooth communication link. A single Bluetooth packet may contain multiple ESP packets.

Packet Description

This section does not apply to the V1connection LE.
When ESP packet(s) need to be transmitted between an Android Bluetooth device and the V1connection, the data will be wrapped in a separate packet before transmission. A special control character is placed around the message to delimit it. Each packet starts and ends with the same packet delimiter (PACK) byte, $7F. A packet has a start delimiter ($7F), a packet length byte, message data byte(s), a packet checksum, and an end delimiter ($7F).

Packet Delimiter	Packet Length	Message Data	...	Message Data	Packet Checksum	Packet Delimiter

There are two special control bytes that are used for a packet:

Packet delimiter (PACK) byte	$7F
Data Link Escape (DLE) byte	$7D

If any byte between the start and end packet delimiter equals a PACK or DLE byte, then a DLE is added before that byte. The byte that equaled a PACK or DLE byte is changed by exclusive-or'ing the byte with $20. The following two bytes would be changed to the following bytes:

$7F	changes to	$7D $5F
$7D	changes to	$7D $5D

The Packet Length byte is the length of the Message Data bytes only. The length byte does not include the PACK bytes, the Packet Checksum byte, added DLE bytes or the Packet Length byte itself.

The Packet Checksum byte is the sum of the Packet Length byte and all of the Message Data bytes before any bytes have been changed to DLE $5D or DLE $5F. When calculating the checksum, sum all of the original Message Data bytes and the Packet Length byte. If any Message Data bytes or checksum byte equals a PACK or DLE byte, then a DLE $5D or DLE $5F are substituted for each of those bytes after the checksum has been calculated.

Packet Example

The following ESP packet will be placed into a packet:

AA D8 EA 31 09 7F 7F 1F 7D 7D 0C 00 00 C9 AB

The Packet Length is set to $0F. The Packet Checksum is set to $4C. Two of the data bytes equal a DLE so they must both be replaced by the two bytes $7D $5D. Two of the data bytes equal the PACK byte they must both be replaced with the two bytes $7D $5F. The packet built is shown below:

7F 0F AA D8 EA 31 09 7D 5F 7D 5F 1F 7D 5D 7D 5D 0C 00 00 C9 AB 4C 7F

V1connection LE Service and Characteristics

This section does not aply to the V1connection

The V1connection LE supports one service with four characteristics. The 128 bit UUIDs for the service and characteristics are defined in Table 2.

Table 2 – V1connection LE service and characteristic UUIDs

Item	UUID
V1connection LE Service	0x92A0AFF49E0511E2AA59F23C91AEC05E
V1outClientInShortCharacteristic	0x92A0B2CE9E0511E2AA59F23C91AEC05E
V1outClientInLongCharacteristic	0x92A0B4E09E0511E2AA59F23C91AEC05E
ClientOutV1inShortCharacteristic	0x92A0B6D49E0511E2AA59F23C91AEC05E
ClientOutV1inLongCharacteristic	0x92A0B8D29E0511E2AA59F23C91AEC05E

The name of the characteristic indicates the direction of data. The V1outClientInShortCharacteristic and V1outClientInLongCharacteristic characteristics are used to transfer data from the wired ESP bus to the iOS device. The ClientOutV1inShortCharacteristic and ClientOutV1inLongCharacteristic characteristics are used to transfer data from the iOS device to the wired ESP bus.

The V1connection LE uses V1outClientInShortCharacteristic and ClientOutV1inShortCharacteristic for ESP packets that are 20 bytes or less. Packets sent or received using these characteristics do not need any additional processing. They can be treated as a raw packet that follows the format in the ESP specification.

The V1connection LE uses the V1outClientInLongCharacteristic and ClientOutV1inLongCharacteristic for ESP packets that are more than 20 bytes long. These long ESP packets are transferred by splitting the packet into smaller chunks before transmitting them. Each chunk is indexed so the receiver can reassemble the packet in the correct order.

Long ESP Packets will be split into chunks using the following rules.

1. The first byte of each chunk will be the Chunk Index Byte, which indicates the chunk index and the number of chunks in the packet.
2. The maximum chunk size, including the Chunk Index Byte, is 20 bytes.
3. Chunks will be transmitted in the correct order. There is not guarantee that the chunks will be received in the correct order so care must be taken when reassembling chunked packets.

A packet chunk with 19 data bytes will have the format

<Chunk Index Byte> <Data Byte 0> <Data Byte 1> ….. <Data Byte 17> <Data Byte 18>

Chunk Index Byte

```
07 06 05 04 03 02 01 00
|  |  |  |  |  |  |  |
|  |  |  |  |  |  |  \- chunk count bit 0
|  |  |  |  |  |  \---- chunk count bit 1
|  |  |  |  |  \------- chunk count bit 2
|  |  |  |  \---------- chunk count bit 3
|  |  |  \------------- chunk index bit 0
|  |  \---------------- chunk index bit 1
|  \------------------- chunk index bit 2
\---------------------- chunk index bit 3
```

Packet chunking example

Source ESP packet is 22 bytes long:

AA D6 EA 23 0B 13 8C E8 89 23 23 89 1F 87 D6 33 87 D2 82 67 68 AB

Packet is split into 2 chunks

Chunk #1: 12 AA D6 EA 23 0B 13 8C E8 89 23 23 89 1F 87 D6 33 87 D2 82

Chunk #2: 22 67 68 AB