Open Drone ID

Implementation Specification

Draft Specification Version 0.59.0

Protocol version 0

June 6, 2018

Contents

1 Introduction ... 4

2 Selection of the technical solution ... 4

3 Implementation Overview ... 5

4 BLE (Bluetooth 4.x compatible) Advertisements ... 5

Beacon Definition ... 5

Additional BLE 4.x Frame Details ... 5

Transmitting Frequency ... 6

5 Bluetooth 5.0 Extended Advertisements ... 8

Bluetooth 5 Extended Advertisement Primary (Pointer) Packet ... 9

Aux Ptr Field Details ... 10

Bluetooth 5 Extended Advertising Secondary Packet ... 10

6 Message Definitions ... 12

General ... 12

Message Header ... 12

Basic ID Message ... 12

Location Message ... 13

Authentication Message ... 14

Self ID Message ... 14

Operator Message ... 14

Enumerated Field Definitions ... 15

7 Compliance and Interoperability ... 15

Update History

Version	Date	Changes	Author
0.54	3/1/2018	Started Change Control (initial version baseline)	G. Cox, Jan S.
0.55	4/3/2018	* Changed all Messages Modes S field from 28 bit, to 32 bit UniqueID * Base ID Message: Moved Sub-Type up to after under Drone Type * Location Message: Changed final reserved field to 4 bits since the Unique ID went from 24 to 32 bits.	G. Cox
0.56	4/20/2018	Added BT5 Extended Advertisements	G. Cox
0.57	4/24/2018	Added Additional Packet Detail Tables, Moved Message Definitions to end	G. Cox
0.58	4/25/2018	Location Message: Moved “Reserved” up to after “Status”	G. Cox
0.58.4	6/06/2018	Minor changes to background, added footnote for WiFi and added IETF RFC2119 reference.	G. Cox, J. Takei
0.59.0	8/15/2018	**Major field updates** * Note: We are not incrementing protocol version until this specification goes from draft to final and after a change is made to the final version. 1. General: 1.1. Format of Messages documentation changed to show Offset/Len in 1st 2 columns and more clearly display bitfields 1.2. Clarified that multi-byte IDs shall be expressed in Network Byte Order, and multi-byte numerical values shall be expressed in little endian. 1.3. Added DRAFT watermark 2. Basic ID Message: 2.1. Shortened Make/Model to 8 bytes and provided a structured format to supply make/model. 2.2. Added “Flags” byte of which bit 0 represents Weight Multiplier 2.3. Increased Gross weight from 14 to 16 bits. This (and moving multiplier to Flag byte) is to divide the message fields along even 8bit boundaries so that it may more easily be parsed/processed. 3. Location Message: (lots of refactoring to split fields and offsets on even byte boundaries) 3.1. Move Multipliers (V-Speed, H-Speed into common byte with Status) 3.2. Increased speed fields from 6 to 8 bits 3.3. Increased Lat/Lon fields from 26 to 32 bits. Not only does this provide 10^7 precision, but it also allows for an easy/common 32bit Int processing. 3.4. Increased MSL and AGL Altitude fields to 16bit and removed multipliers. 3.5. Added Horizontal/Vertical precision confidence values 3.6. Removed all references to “heading” fields since this can be derived from the speed vector components. 3.7. Removed operator location (this is how we made all the space) and moved to a separate dedicated Operator Message. 3.8. Allocated 4 bytes as reserved 3.9. Moved enumerated status list to “Enumerated Field Definitions” 4. Added Operator Message that includes lat, lon, source.	G. Cox

1 Introduction

On December 19th 2017 the Federal Aviation Administration (FAA) published the UAS Remote Tracking & ID ARC Report[1] to update the public about the latest results from the Aviation Rulemaking Committee (ARC) chartered by the FAA.

Within the ARC recommendation were some options for “Broadcasting” a Drone ID. This specification is designed to meet such needs expressed in the ARC Report.

This document is currently in *DRAFT* and is under a standardization process within the ASTM F38 Remote ID Workgroup. The outcome of this collaboration will most certainly result in many changes as a part of this process.

2 Selection of the technical solution

The solution outlined in this specification is for the broadcasting category of drone identification. In the ARC, several solutions were discussed. After considering all inputs from the ARC and an initial decision process, the current solution in this document is to use the Bluetooth Low Energy (BLE) standard V4.2 Advertisements and 5.0 Advertising Extensions as the underlying technology for the Open Drone ID implementation. The primary factors justifying this technology are:

- Commoditized open standard supported by most modern smartphones as a receiving device
- Very low part(s) cost to add to a drone
- Very low weight with solutions below 10g
- Robust protocol implementation with congestion handling and up to 1km range (v5.0 AE).
- Very easy to retrofit to existing drones

The range of Bluetooth V4.0 depends on the transmit power, as well as the exact transmitter and receiver setup including the antenna and external noise sources. Tests have been conducted to show a range from a drone to a smartphone of over 200m. For better antenna installations at critical sites the range could be increased to more than 1km. Based on the specifications, with Bluetooth V5.0, the range will quadruple. This range will be suitable for most scenarios while providing 2 huge benefits:

1. Public Safety Departments cost of acquiring receivers could potentially be minimized to the cost of their available smartphones.
2. The general public can use their smartphone to read the Drone IDs and help by accurately reporting airspace contention or security problems.

Bluetooth uses three different beacon channels that can broadcast messages to non-specific endpoints. Although the remaining 37 channels operate in the 2.4 Ghz range, where WiFi resides, the channels are much narrower and specifically the beacon channels are outside of the bands of typical WiFi traffic[2]. (see Figure 1 below).

[1] https://www.faa.gov/news/updates/?newsId=89404
[2] Based on common router configurations that choose channels 1,6,11 by default

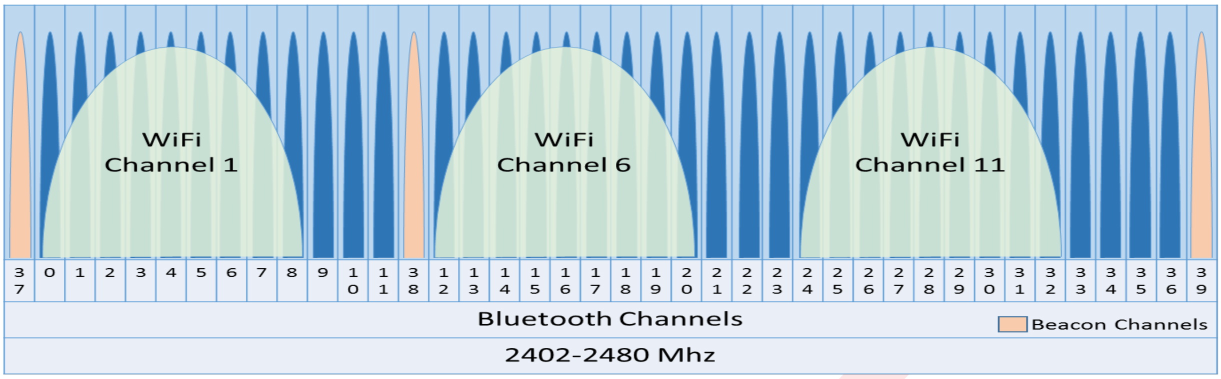

Figure 1 – Bluetooth Channels

3 Implementation Overview

In this specification, the intent is to use a Bluetooth radio to send connectionless broadcast frames (Advertisements) that work with both BT 4.x and 5.x receivers. Supporting both architectures allows for compatibility with existing BT4 receivers (like most cell phones as of 2018), yet can still take advantage of the range enhancements (4x) of BT5 which is starting to ship with newer cell phones and can also be installed as external receivers.

4 BLE (Bluetooth 4.x compatible) Advertisements

BLE supports a "Broadcast Frame" to go out on the beacon channels with a custom message length limit of 26 bytes. These broadcast messages shall be "uncoded" and conform to Bluetooth Core Specification 5.0, Volume 6, Part B, Sections 2.1 and 2.3.1.[3]

Beacon Definition

These BLE frames shall be sent as illustrated below in Figure 2.

BLE4 Advertisement Frame							
Preamble (1 Byte)	Acc Addr (4 Bytes)	PDU Hdr (2 Bytes)	AD Addr (6 Bytes)	AD Flags (4 Bytes)	AD Counter (1 Byte)	Open Drone ID Message (26 Bytes)	CRC (3 Bytes)
0xAA	0x8E89BED6	0x2025	<public>	0x1FFF0200	0x00	<message>	<calculated>

Open Drone ID Message			
Msg Type (4 bits)	Version (4 bits)	UniqueID	Message (21 Bytes)
0x1 – 0x4	0x0-0xF	4 Bytes	<Open Drone ID message>

Figure 2 –BLE Advertisement Frame Format

Additional BLE 4.x Frame Details

Field	Size*	Value	Contents
Preamble	1	0xAA	LE 1M Packet
Acc Address	6	0x8E89BED6	Broadcast Packet
PDU Hdr	2	0x2025	

PDU Type	0x2	ADV_NONCONN_IND – Connectionless Advertisement
RFU	0	Reserved
ChSel	0	Reserved
TxAdd	0	Indicates AD Addr is HW Address (rather than random)

[3] https://www.bluetooth.com/specifications/bluetooth-core-specification

<table>
<tr><td></td><td></td><td></td><td><table><tr><td>RxAdd</td><td>0</td><td>Reserved</td></tr><tr><td>Len</td><td>0x25</td><td>37 Bytes</td></tr></table></td></tr>
<tr><td>AD Addr</td><td>6</td><td>0xXXXXXX</td><td>Unique Hardware Address of Bluetooth MAC</td></tr>
<tr><td>AD Flags</td><td>4</td><td>0x1FFF0200</td><td><table><tr><td>Length</td><td>0x1F</td><td>31 Bytes (excluding this field)</td></tr><tr><td>Type</td><td>0xFF</td><td>Manufacturer Specific</td></tr><tr><td>Mfg Code</td><td>0x0200</td><td>Intel (this is a placeholder until a mfg code is established by the standards collaboration.</td></tr></table></td></tr>
<tr><td>AD Counter</td><td>1</td><td>0xXX</td><td>Msg Counter: Start at 0, increment for each message of the same type</td></tr>
<tr><td>ODID Msg</td><td>26</td><td><26 Bytes></td><td>Open Drone ID Message – see section 0 (
Message Definitions)</td></tr>
<tr><td>CRC</td><td>3</td><td><calculated></td><td>CRC Error Correction Data as defined in Bluetooth Core Specification 5.0, Volume 6, Part B, Section 3.1.1</td></tr>
</table>

Transmitting Frequency

These 26 byte long messages, as defined in section 0 (

Message Definitions), shall be sent by each drone Bluetooth beacon. Depending on whether the data is static or dynamic, the messages will be sent at a low or higher frequency (respectively).

As such, the following message frequencies shall be maintained:
Static: Every 3 seconds.
Dynamic: 3 per second.

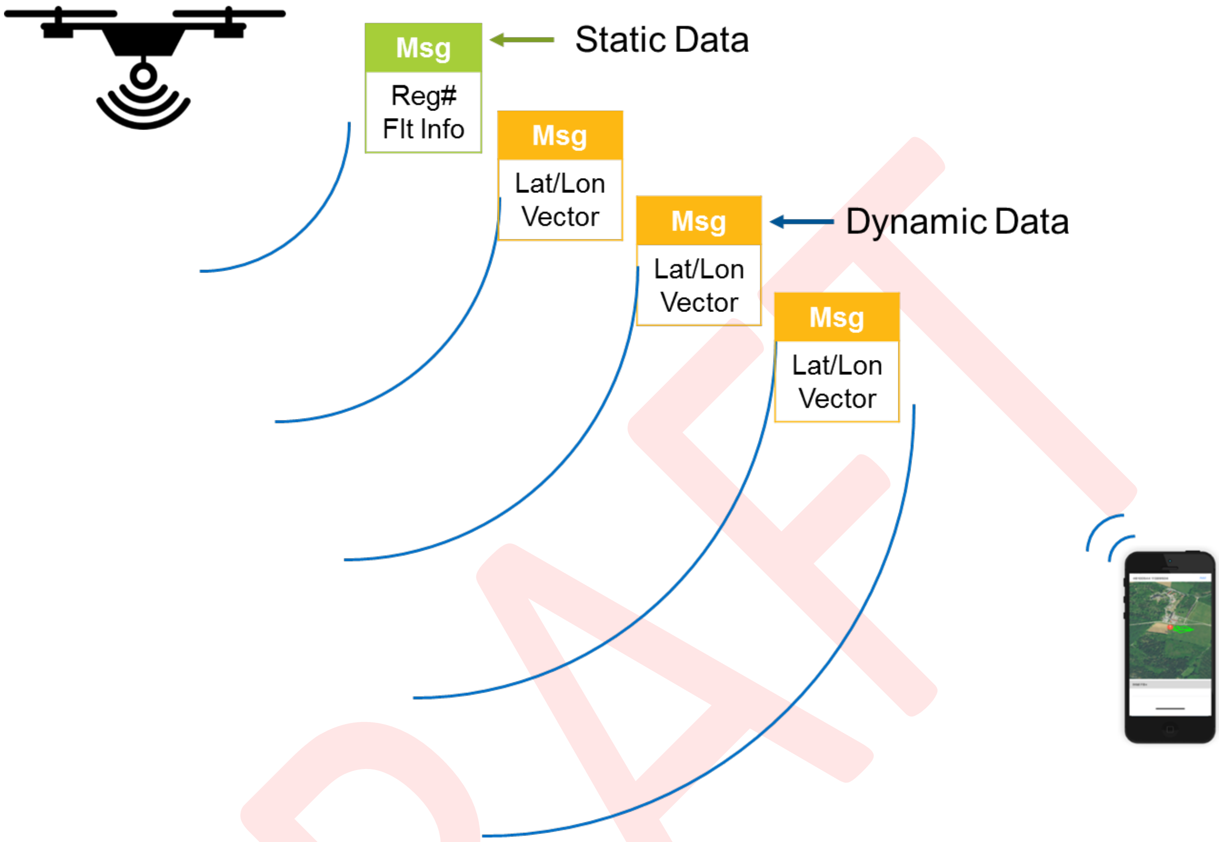

Figure 3 - Static and Dynamic Messages

5 Bluetooth 5.0 Extended Advertisements

If Implementing this specification using Bluetooth 5, In addition to sending standard (ADV_NONCONN_IND) BLE (4.2) advertisements, Bluetooth 5 Extended Advertisements (ADV_EXT_IND + AUX_ADV_IND) must be sent as well at the same rate as Dynamic Data (see Section 3 *Transmitting Frequency*) and they must be sent on an LE Coded (S=8) PHY. This will add Forward Error Correction (FEC) and increase the range of the advertisements by 4x. These messages shall conform to Bluetooth Core Specification 5.0, Volume 6, Part B, Sections 2.2 (LE Coded PHY, S=8).

While BLE (4.2) advertisements broadcast on the beacon channels 37,38,39, Bluetooth 5 adds Extended Advertising that allow for up to 255 byte advertisements on the “non-beacon” channels by implementing a pointer in the primary beacons directing the receiver to read from the secondary channel.

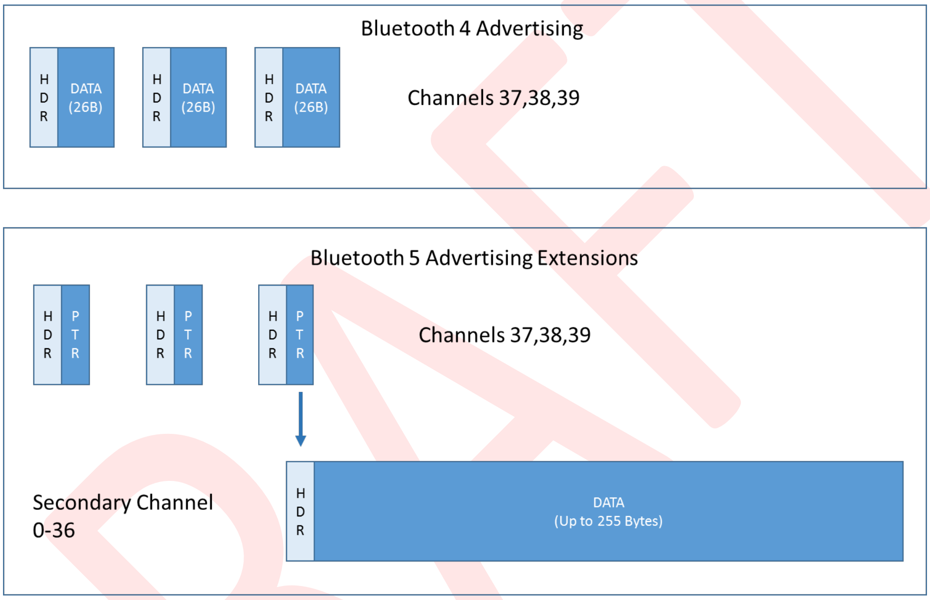

Figure 4 - Bluetooth 4 and 5 Extended Advertising Comparison

When performing a Bluetooth 5 Advanced Advertisement, all messages must be sent together as a single “message pack” as illustrated below in Figure 5Figure 1.

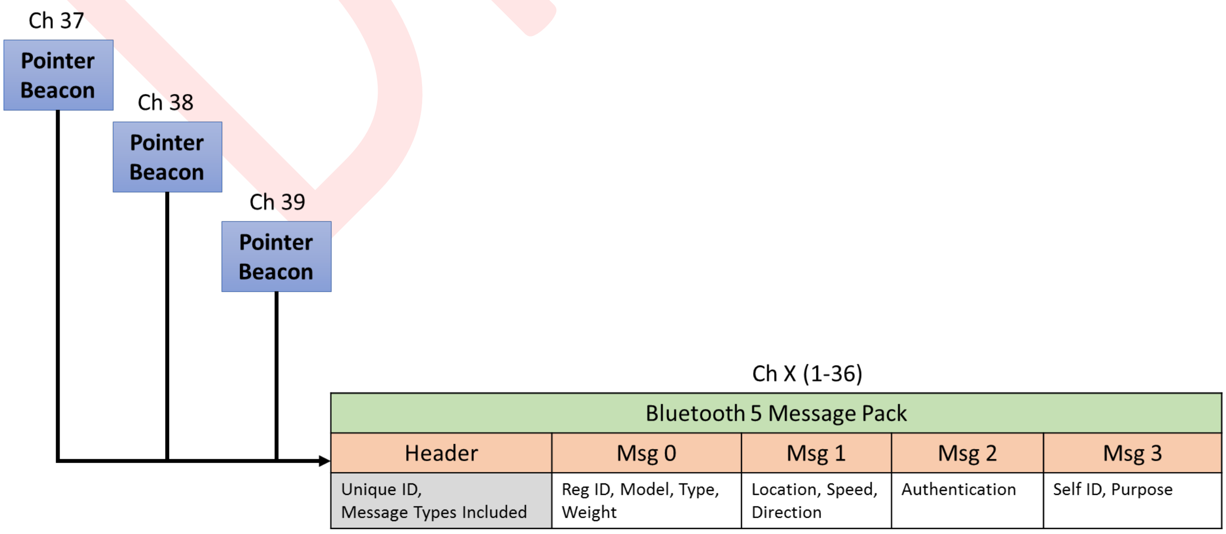

Figure 5 -- Bluetooth 5 Message Pack

Bluetooth 5 Extended Advertisement Primary (Pointer) Packet

The Bluetooth 5 Extended Advertisement Primary packet includes a pointer to the Secondary Packet as illustrated below. Therefore, the Primary packet shall be broadcast through all 3 beacon channels, followed by the Secondary packet on the remaining channels (see <reference to packet spreading algorithm).

BLE5 Long Range Advertisement Pointer Frame (LE Coded)										
Preamble (1 Byte) [Coded Phy]	Acc Addr (4 Bytes)	CI (2bits) [S=8]	TERM1 (3bits)	PDU Hdr (2 Bytes)	Ext Hdr Len (6 Bits)	Adv Mode (2 bits) non-scan undirect	Ext Header (12 Bytes)	Adv Data	CRC (3 Bytes)	Term2 (3bits)
0x3C	0x8E89BED6	00b	<xxx>b	0x700D	0x0C	00b	<12 bytes>	N/A	<calculated>	<xxx>b

Additional Primary Packet Details

<table>
<tr><th>Name</th><th>Size*</th><th>Value</th><th>Value Description</th></tr>
<tr><td>Preamble</td><td>1</td><td>0x3C</td><td>LE Coded PHY</td></tr>
<tr><td>Acc Addr</td><td>6</td><td>0x8E89BED6</td><td>Broadcast Packet</td></tr>
<tr><td>CI</td><td>2 bits</td><td>00b</td><td>Coding Indication: FEC Block 2 is coded using S=8 (longest range)</td></tr>
<tr><td>Term1</td><td>3 bits</td><td>xxxb</td><td>FEC Block 1 Termination as defined in Bluetooth Core Specification 5.0, Volume 6, Part B, Section 3.3.1</td></tr>
<tr><td>PDU Hdr</td><td>2</td><td>0x700D</td><td>
<table>
<tr><th>Field</th><th>Bits</th><th>Hex</th><th>Desc</th></tr>
<tr><td>PDU Type</td><td>0111</td><td>0x7</td><td>ADV_EXT_IND (Primary)</td></tr>
<tr><td>RFU</td><td>0</td><td>0x0</td><td>Reserved</td></tr>
<tr><td>ChSel</td><td>0</td><td></td><td>Reserved</td></tr>
<tr><td>TxAdd</td><td>0</td><td></td><td>Reserved</td></tr>
<tr><td>RxAdd</td><td>0</td><td></td><td>Reserved</td></tr>
<tr><td>Len</td><td>0010 0101</td><td>0x0D</td><td>13 Bytes</td></tr>
</table>
</td></tr>
<tr><td>Ext Hdr</td><td>12</td><td></td><td>
<table>
<tr><th>Field</th><th>Size*</th><th>Bits (binary)</th><th>Hex</th><th>Desc</th></tr>
<tr><td>Flags</td><td>1</td><td>0001 1001</td><td>0x19</td><td>Field Selection (AdvA, ADI, Aux Ptr)</td></tr>
<tr><td>AdvA</td><td>6</td><td><HW ADDR></td><td>0xXXXXXX</td><td>Adv Address (HW Addr)</td></tr>
<tr><td>ADI</td><td>2</td><td>0000 0000
0000 xxxx</td><td>0x0000 –
0x000F</td><td>Advertising Data ID (12bits) = 0

Advertising Set ID (4bits):
Increment each time data changes</td></tr>
<tr><td>Aux Ptr</td><td>3</td><td>cccc cca0
dddd dddd
dddd d010</td><td>0xXXXXXX</td><td>cccccc = Channel
a = clock accuracy
0 = 30us offset multiplier
dddddd = offset/delay
010 =LE Coded Phy
** See Aux Ptr Field Details below.</td></tr>
</table>
</td></tr>
<tr><td>CRC</td><td>3</td><td></td><td>CRC Error Correction Data as defined in Bluetooth Core Specification 5.0, Volume 6, Part B, Section 3.1.1</td></tr>
<tr><td>Term2</td><td>3bits</td><td></td><td>FEC Block 2 Termination as defined in Bluetooth Core Specification 5.0, Volume 6, Part B, Section 3.3.1</td></tr>
</table>

* Bytes

Aux Ptr Field Details

The Aux Ptr Field in the Primary Packet shall be implemented in accordance to the Bluetooth Core Specification 5.0, Volume 6, Part B, Section 2.3.4.5 with the following guidance.

Channel Index	Shall be calculated using the following formula: Channel = (Current Channel + 9) % 36 This will ensure some entropy by hopping through the channels and spreading out the beacons to minimize the effects of external interference.
Clock Accuracy (CA)	0: 51 – 500ppm 1: 0 - 50ppm
Offset Units	0: 30us
Aux Offset/Delay	This represents the time offset from when the primary packet is sent and the secondary packet. Since all 3 primary packets are sent prior to the secondary packet, the offset is different for each one. This offset should be calculated based on the Bluetooth Core Specification 5.0 specification. The following offsets may be used as guidance : Beacon 1: 166 us Beacon 2: 114 us Beacon 3: 62 us These calculations are based on a primary packet time of 1552us + a T_MAFS (minimum aux frame space) of 300us divided by the offset multiplier unit of 30us. The time of sending the current beacon + remaining beacons must be included. Thus, Beacon 1 includes the time of itself + 2 more beacons + T_MAFS.
Aux PHY	010: LE Coded Phy

Bluetooth 5 Extended Advertising Secondary Packet

The secondary packet contains the actual desired advertisement. Additionally this packet contains a 16 bit “Message Type Mask” which includes bit representing that message types included. All message types supported shall be represented within a single Secondary Packet in the order of lowest message type (index) first. Only the messages selected in the Message Type mask shall be presented within the single Secondary Advertising Packet (with a capacity of up to 255 bytes).

BLE5 Long Range Advertisement Data Frame (LE Coded)										
Preamble (1 Byte) [Coded Phy]	Acc Addr (4 Bytes)	CI (2bits) [S=8]	TERM1 (3bits)	PDU Hdr (2 Bytes)	Ext Hdr Len (6 Bits)	Adv Mode (2 bits) non-scan undirect	Ext Header (12 Bytes)	Adv Data	CRC (3 Bytes)	Term2 (3bits)
0x3C	0x8E89BED6	00b	<xxx>b	0x7025	0x09	00b	<9 Bytes>	<7 + N*21 Bytes>	<calculated>	<xxx>b

*N = Number of Message Types in Message Pack

Open Drone ID BLE5 Message Pack							
MsgType (4 bits) [BT5hd]	Version (4 bits)	UniqueID	Msg Types Mask (16 bits)	Counter	Message (Type 0)	Message (Type 1)	....
0xF	0x0-0xF	4 Bytes	0x000F	<1 Byte>	<21Bytes>	<21Bytes>	....

Name	Size*	Value	Value Description
Preamble	1	0x3C	LE Coded PHY
Acc Addr	6	0x8E89BED6	Broadcast Packet
CI	2 bits	00b	Coding Indication: FEC Block 2 is coded using S=8 (longest range)
Term1	3 bits	xxxb	FEC Block 1 Termination as defined in Bluetooth Core Specification 5.0, Volume 6, Part B, Section 3.3.1

<table>
<tr><td>PDU Hdr</td><td>2</td><td>0x70XX</td><td>
<table>
<tr><th>Field</th><th>Bits</th><th>Hex</th><th>Desc</th></tr>
<tr><td>PDU Type</td><td>0111</td><td>0x7</td><td>AUX_ADV_IND (Secondary)</td></tr>
<tr><td>RFU</td><td>0</td><td rowspan="4">0x0</td><td>Reserved</td></tr>
<tr><td>ChSel</td><td>0</td><td>Reserved</td></tr>
<tr><td>TxAdd</td><td>0</td><td>Reserved</td></tr>
<tr><td>RxAdd</td><td>0</td><td>Reserved</td></tr>
<tr><td>Length</td><td>xxxx xxxx</td><td>0xXX</td><td>18 + N*21 Bytes where N is the number of Messages in the Message Pack</td></tr>
</table>
</td></tr>
<tr><td>Ext Hdr</td><td>12</td><td></td><td>
<table>
<tr><th>Field</th><th>Size*</th><th>Bits (binary)</th><th>Hex</th><th>Desc</th></tr>
<tr><td>Flags</td><td>1</td><td>0000 1001</td><td>0x09</td><td>Field Selection (AdvA, ADI)</td></tr>
<tr><td>AdvA</td><td>6</td><td><HW ADDR></td><td>0xXXXXXX</td><td>Adv Address (HW Addr)</td></tr>
<tr><td>ADI</td><td>2</td><td>0000 0000
0000 xxxx</td><td>0x0000 –
0x000F</td><td>Advertising Data ID (12bits) = 0

Advertising Set ID (4bits):
Increment each time data changes</td></tr>
</table>
</td></tr>
<tr><td>CRC</td><td>3</td><td></td><td>CRC Error Correction Data as defined in Bluetooth Core Specification 5.0, Volume 6, Part B, Section 3.1.1</td></tr>
<tr><td>Term2</td><td>3bits</td><td></td><td>FEC Block 2 Termination as defined in Bluetooth Core Specification 5.0, Volume 6, Part B, Section 3.3.1</td></tr>
</table>

6 Message Definitions

General

Each 26 byte message shall begin with a 5 byte header followed by 21 bytes of data. Numerical (such as the Unique ID) shall be expressed in Network Byte Order (MSB First). Other numerical values that represent a magnitude (such as Lat, Lon, Weight, Alt, etc.) shall be expressed as “little endian” (LSB First).

Message Header

The message header includes the Message Type, Protocol Version and Unique ID and must be sent in each message. If the first byte of the Unique ID is 0x00, then the remaining 3 bytes shall be the ICAO 24-Bit Aircraft Address (Mode-S Code).

Header (5 bytes)			Message (21 bytes)
Message Type (4bits)	Protocol Version (4bits)	Unique ID (4 Bytes)	Message Fields based on Message Type
0x1-0xF	0x0	Ex: 0x00A79E9F	< Message Data >

Figure 6 - Message Format

Basic ID Message

Message Type: 0x0, Static

The BasicID message includes the ICAO registration (eg: N-Number) as well as a 32bit unique ID. Additionally, other static data about the aircraft (Make/Model, Type, etc). It is sent at a low frequency. This message is **mandatory**.

Offset (Byte)	Len (bits)	Data Field	Details	Limitations	Example
5	64	RegID	ICAO Registration ID/Callsign	8 Byte Reg#/N#	N590NM
13	64	Make/Model	Make/Model AC (Abbreviated) [CCC][MMMMM] Company Code Abbreviated Model	8 Bytes	DJIMP YUNTH INTF8P
21	8	Flags	[0 0 0 0 0 0 0 0] Reserved Weight Multiplier 0 = x10 1 = x1000		
22	8	Drone Type	VTOL, fixed wing, hybrid, etc. (See *Figure 11* below for more details.)	Max. 255 types	
23	8	Drone Category (cert)	More information about the drone (e.g. weight, size) (Reserved for now)	Limits depending on the category. Categories can be defined to overcome limits for today’s unknown systems	
24	16	Drone Gross Weight	Weight in Grams x10	0 - 655,350g Up to 655,000kg w/multiplier	

Figure 7 - Basic ID Message

Location Message

Message Type: 0x1, Dynamic

The Location/Vector message provides the location, altitude, direction and speed of the drone. This message is **mandatory**. Note: When using the "multiplier" fields, the values shall be expressed in the most specific value possible. Therefore, moving to the next available multiplier shall only happen after the maximum integer value has been reached for the base value field. For Example: 2047 MSL shall be expressed as Altitude=2047, Altitude Multiplier = 0 (1). For 2048 MSL, Altitude=205 (round), Altitude Multiplier=1 (10) which gives an interpreted value of 2050.

Offset (Byte)	Len (bits)	Data Field	Details	Limitations	Example
5	8	Status, Flags	Bits [7..0] [0 0 0 0] [0 0 0 0] Status: Flags: Reserved V-Speed Multiplier H-Speed Multiplier 0 = x1, 1 = x10	0..15 statuses Multiplier enables speeds up to +- 1270. Only use when speed exceeds 127 m/s.	
6	8	Speed North/South	Speed in m/s (+N, -S)	Up to +-1270 m/s	5 m/s
7	8	Speed East/West	Speed in m/s (+E,-W) *Note, Heading/Track is derived as a vector of these 2 magnitudes.	Up to +-1270 m/s	2 m/s
8	8	Speed Vertical	Vertical Speed m/s (+ up, - down)	Up to +-127 m/s (25k ft/min)	0 m/s
9	32	Latitude	Latitude of drone	Int signed deg*10^7	-48123987
13	32	Longitude	Longitude of drone	Int signed deg*10^7 (11mm precision)	11989298
17	16	Altitude (MSL)	Pressure Altitude	+-32767m (107503ft)	
19	16	Height (AGL)	Height above ground or start pt	+-32767m (107503ft)	65m
21	8	Horizontal/Vertical Precision	Bits [7..0] [0 0 0 0] [0 0 0 0] Vertical Horizontal HDOP/VDOP Values 0=Unknown If >0 and <1, round up If >1, round to nearest int	These are used to derive accuracy	
22	32	Reserved	4 bytes reserved for future use	4 bytes	

Figure 8 - Location Message

Authentication Message

Message Type: 0x2, Static

An Authentication message can provide an authentication token to prove the authenticity of these messages. For now, this is an optional placeholder to accommodate future authentication schemes and requirements.

Offset (Byte)	Length (bits)	Data Field	Details	Limitations	Example
5	128	Authentication Data	Opaque Authentication Data	16 Bytes	SignedHash(SMAC+ID)
21	40	Reserved	Reserved for future use	5 Bytes	

Figure 9 - Authentication Message

Self ID Message

Message Type: 0x3, Static

The Self-ID message is an opportunity for the Drone Pilot to (**optionally**) declare their identity and purpose of the flight. This can serve the purpose of putting people at ease if concerns exist as to why a drone is flying in a particular area. For Example: A Realtor may want to declare that they are taking photos of a client’s house to put the neighbors at ease.
This message is **optional**

Offset (Byte)	Length (bits)	Data Field	Details	Limitations	Example
5	168	Operation Description	Text of Operator and Purpose	21 Bytes	DronesRus: Survey

Figure 10 - Self ID Message

Operator Message

Message Type: 0x4, Static (slow update, but does change)

The Operator Message represents information about the ground control station (GCS). At this revision, it only contains the location. If the mechanism used for determining the location of the GCS is “takeoff location”, then the operator must remain near (within 20m) the takeoff location and may use the Location Source of 0. Otherwise, if the operator is “roving”, then Live GPS (or other live update mechanism) must be used to ensure the operator location remains accurate. Since this value generally does not change at the same rate of a drone, the update frequency shall be the same rate as static messages.

Offset (Byte)	Length (bits)	Data Field	Details	Limitations	Example
5	32	Latitude	Latitude of operator	Int signed deg*10^7	-48123987
9	32	Longitude	Longitude of operator	Int signed deg*10^7 (11mm precision)	11989298
13	8	Flags	Bits [7..0] [0 0 0 0 0 0 0 0] Reserved Location Source: 0 = Take Off, 1 = Live GPS		
14	96	Reserved	Reserved for future use	12 bytes	

Enumerated Field Definitions

In the data structures above, some fields are enumerated values. The table below assigns meaning to those enumerations. This can be updated as needed.

Field Name	Details	Length (bits)	Notes
Drone Type	0: None 1: Fixed Wing Powered 2: Rotorcraft/Multirotor 3: LTA (Lighter than Air) Powered 4: LTA Unpowered (Balloon) 5: VTOL 6: Free Fall/Parachute 7: Rocket 8: Glider 9: Other 10-255: Reserved	8	Up to 255 Types
Drone Category	(Placeholder Examples) 0: None 1: OOP Compliant 2: Night Compliant 3: Other types that may fit into a regulatory category that permit certain types of operations.	8	Up to 255 Categories
Status	0: Ground, 1: Airborne (manual control) 2: Returning Home 5: Automated Mission 8: Emergency Landing 9: Control Loss 10-15: Reserved	4	Up to 16 Statuses

Figure 11 - Enumerated Field Definitions

7 Compliance and Interoperability

As of this version, compliance can be "self-certified" using the following means:

1. Every "shall", "must" and any other logical directive in this document must be implemented. (See IETF RFC2119 for adopted definitions of imperatives: https://www.ietf.org/rfc/rfc2119.txt)
2. Interoperability shall be verified against "known working" clients for both BLE 4 and Bluetooth 5 Extended Advertising receivers.
3. Hardware/RF/Signal compliance TBD.
4. If a system is not compliant with this spec, then it may not claim, advertise or display references to "Open Drone ID".