Open Drone ID

Message Specification

Draft Specification Version 0.60.0

Protocol version 0

August 23, 2018

Contents

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
0.57	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
0.59.0	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
0.60.0	8/23/2018	1. Extracted Data to a stand-alone document 2. Corrected Location Flags to be NS, EW Multipliers (removed Vertical Mult).	

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.

This specification is designed to meet such needs expressed in the ARC Report.

This document is part of a group of documents to meet the total solution requirements of Remote ID.

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   Related Documents

Open Drone ID - Bluetooth Broadcast Specification

Open Drone ID - WiFi Broadcast Specification

Open Drone ID - Network Access Specification

3   Block Message Definitions

General

The "Block" messages are intended to be packed into lightweight direct broadcast packets that are designed to fit into WiFi or Bluetooth "Beacon Advertisements".

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 1 - 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 6* 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 2 - 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 ——— NS-Speed Multiplier ——— EW-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 3 - 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 4 - 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 5 - 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 6 - Enumerated Field Definitions

4 JSON Representation of Messages

When transmitting Open Drone ID messages over a network to a Web Service, a JSON representation shall be required. Below are the JSON example representations of the above messages. *Note that multipliers are not used in these messages.

Basic ID (Message Type 0) JSON Representation

```
{
    "MessageType": 0,
    "Version": <UInt8>,
    "UniqueID": "hex string(4)",
    "RegistrationID": "string(8)",
    "MakeModel": "string(8)",
    "Type": <Uint8>,
    "Category": <Uint8>,
    "GrossWeight": <Uint32 (grams)>
}
```

Location (Message Type 1) JSON Representation

```
{
    "MessageType": 1,
    "Version": <UInt8>,
    "UniqueID": "hex string(8)",
    "Status": <Uint8>,
    "SpeedNS": <Int32 (+-m/s)>,
    "SpeedEW": <Int32 (+-m/s)>,
    "SpeedVertical": <Int32 (+-m/s)>,
    "Latitude": <float>,
    "Longitude": <float>,
    "AltitudeMSL": <Int32 (+-meters)>,
    "HeightAGL": <Int32 (+-meters)>,
    "hDOP": <Int8>,
    "vDOP": <Int8>
}
```

Authentication (Message Type 2) JSON Representation

```
{
    "MessageType": 2,
    "Version": <UInt8>,
    "UniqueID": "hex string(8)",
    "AuthToken": "string(16)"
}
```

Self ID (Message Type 3) JSON Representation

```
{
    "MessageType": 2,
    "Version": UInt8,
    "UniqueID": "hex string(8)",
    "Description": string(21)
}
```

Operator (Message Type 4) JSON Representation

```
{
    "MessageType": 4,
    "Version": UInt8,
    "UniqueID": "hex string(8)",
    "Latitude": <float>,
    "Longitude": <float>
}
```

5 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".