Open Drone ID

Message Specification

Draft Specification Version 0.64.3

Protocol version 0

Mar 3, 2019

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).	G. Cox
0.61.0	11/9/2018	1. Major Updates to consensus agreed field list	G. Cox

		2. Header: Removed UniqueID (moved to BasicID). Unique HW address will need to be used as a "session key". Message size has gone from 21 to 25 bytes. 3. Basic ID: Removed: Flags, Weight, Reg ID, Make/Model Added: Single Byte of ID Type + Drone Type Added: General 24 Byte ID Field (of ID type) 4. Location Message: Replaced AGL with Geodetic Alt Added Age of Data (in 10ms) 5. Authentication Message: Added Auth Type/Page Number Auth Type will drive Format of Message 6. Self ID Message: Incresed to 25 bytes (see item 2) 7. Added Formation info to Operator message 8. Added Horizontal and Vertical Accuracies to Enumerated Field Definitions	
0.62.0	2/11/2019	1. Fix: Text to show header size is 1 byte 2. Change: Total Message Size reduced from 26 to 22 bytes 3. Change: Self ID Msg: Reduced Unique ID size to 20 bytes 4. Change: Location Msg; Speed multipliers to from 0 & 10 to 0.25 & 0.75 5. Change: Location Msg: Vertical speed multiplier to 0.5 6. Change: Location Msg: All Altitudes are now +1000 and *0.5 and unsigned int. 7. Add: Location Msg: AGL above takeoff point 8. Add: Location Msg: Horizontal Velocity Accuracy (both data element and enumeration) 9. Add: Self ID: Description Type	G. Cox
0.62.1	2/24/2019	1. No message format changes in this revision. 2. Used more consistent UAS term within field names (rather that mixing drone, UAV, UAS. The term Drone is still used in the doc text, but not for field names. 3. Added some clarifying text around optionality of Auth Message. 4. Fixed UAS Type enumeration to show 4 bits on the unumeration table. 5. Added some clarification text to distinguish Horizontal/Vertical (Location) Accuracy from Horizontal Velocity (speed) 6. Made JSON format more consistent with block field names.	G. Cox
0.64.3	3/10/2019	1. Improved/added sub-level numbering. 2. Inserted top level message list table. 3. Changed Message size from 22 to 25 bytes (including ODID 1 byte header). An updated technique in the Wi-WiFi spec allowed us to this. 4. Added Encoding Instructions table to clarify some of the more complex encoding techniques in the Location/Vector and System messages. 5. Changed 1 byte Data Age Field to 16bit Timestamp 6. Increased aax Authentication and Self ID field sizes utilizing change #3 7. Renamed Operator Message to System Message 8. Moved Flags in System Message to 1st field to be consistent with other messages. 9. In System Message, renamed field from "Formation" to hopefully more general "Group" term. Additionally, added "Group Altitude" 10. Updated JSON structure to align with new fields and names	

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 F38.02 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

3.1 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”. There are 5 message types identified in Table 1. Subsequent subsections further describe each message type.

#	Message Name	Purpose	Details
0	Basic ID	Provides ID for UA, characterizes the type of ID, and Identifies type of UA	3.3
1	Location/Vector	Provides location, altitude, direction and speed of UA	3.4
2	Authentication	Optional message that provides authentication data for the UA	3.5
3	Self-ID	Optional message that can be used by operators to identify themselves and the purpose of an operation	3.6
4	System	Identifies the location of the operator	3.7

Each message shall be 25 bytes in length (padded with nulls as needed). Each message shall begin with a 1 byte header followed by 24 bytes of data. Non-magnitude values, Strings, or IDs that may be or may not be numerical (such as the Unique ID) shall be expressed in Network Byte Order which reads in a common left to right byte order. Other numerical values that represent a magnitude expressed as a 16 or 32 bit integer (such as Latitude, Longitude, Weight, Altitude, etc.) shall be expressed as “little endian” (marked as “LE” in

[1] https://www.faa.gov/news/updates/?newsId=89404

the tables below) meaning that the byte order is reversed where the least significant byte (LSB) is on the left, and the most significant byte is on the right.

If opting out of an optional message, it is not necessary to send that (blank) message. For “optional” fields within messages being sent, (see Common Data Dictionary), they shall be filled in as stated in the corresponding block message format and if opting out, they shall be filled with nulls (0s) unless an alternate default is stated. This allows the block message to stay properly aligned with the field definitions.

3.2 Message Header

The message header includes the Message Type and Protocol Version and must be sent in each message.

Header (1 byte)		Message (24 bytes)
Message Type (4bits)	Protocol Version (4bits)	Message Fields based on Message Type
0x1-0xF	0x0	< Message Data >

Figure 1 - Message Format

3.3 Basic ID Message

Message Type: 0x0, Static, Mandatory

The Basic ID message includes the ID Type, UAV Type and the Unique ID. This Unique ID shall (BMG0070) default to the Manufacturer Serial number expressed in the ANSI/CTA-2063 Physical Serial Number format. Once the aircraft is provisioned, the UAS ID shall be one of the following:

(a) a Civil Aviation Authority (CAA) issued Unique ID for the UAV, or

(b) a UTM Assigned ID if operating within a UTM system formatted as a RFC4122 UUID (Binary encoded to 16 bytes).

Offset (Byte)	Len (Bytes)	Data Field	Details	Limitations	Example
1	1	ID Type, UAS Type	[0 0 0 0 0 0 0 0] ID Type 0: None 1: Serial Number (ANSI/CTA-2063) 2: CAA Assigned ID 3: UTM Assigned ID (UUID RFC4122) UAS Type VTOL, fixed wing, hybrid, etc. (See Enumerated Field Definitionsfor more details.)	Up to 16 ID types Up to 16 UAS types	
2	20	UAS ID	UAS ID within the format of ID Type	Max. 20 Bytes	
22	3	Reserved			

Figure 2 - Basic ID Message

3.4 Location/Vector Message

Message Type: 0x1, Dynamic, Mandatory

The Location/Vector message provides the location, altitude, direction and speed of the drone.
This message is **mandatory**.

Note: Speed is expressed in North(+)/South(-) and East(+)/West(-) components. This will facilitate computation of bearing and resultant speed. Several of the fields require special encodings to better pack the data and to allow for more precise values. If indicated, the transmitted data shall be encoded according to the Encoding Table (below this message specification). Additionally any fields that requires multiplier flags to be set shall be set according to the Encoding Table as well.

Encoding Logic

Some of the fields in the Location/Vector and System Messages have some encoding techniques to either compress the data or to allow for a more optimal resolutions of precision. The table below is here to clarify how the fields shall be encoding and give direction on how the fields may be decoded.

Field Type	To Encode (sender)	To Decode (receiver)

Speed (NS or EW)
(Int8)
	if abs(Value)/0.25 <= 127
	EncodedValue = Value / 0.25
	Set Multiplier Flag to 0
	
	if abs(Value)/0.25 > 127 and abs(Value) < 127
	if Value < 0 (negative)
	EncodedValue = (Value + (127*0.25)) / 0.75
	else
	EncodedValue = (Value - (127*0.25)) / 0.75
	Set Multiplier Flag to 1
	
	if (Value >= 127m/s)
	EncodedValue = 127
	
	*Note: Encoded Value must be rounded to nearest Integer
		if Multiplier Flag = 0
		Value = EncodedValue * 0.25
		else if Multiplier Flag = 1
		If EncodedValue < 0 (negative)
		Value = (EncodedValue * 0.75) - (127*0.25)
		else
		Value = (EncodedValue * 0.75) + (127*0.25)
		
		Encoding Rationale: This allows for a higher speed precision of 0.25 m/s for lower speeds (0.5kts) and 0.75 m/s (1.5kts) at higher speeds.

Lat/Lon	(Int32) EncodedValue = Value * 10^7	(Double) Value = EncodedValue / 10^7
Vertical Speed	(Int8) EncodedValue = Value / 0.5	(Float) Value = EncodedValue * 0.5
Altitude	(UInt16) EncodedValue = (Value / 0.5) + 1000	(Float) Value = (EncodedValue * 0.5) - 1000 Encoding Rationale: Eliminates unused negative integer space and increases precision to 1/2m.
Time Stamp	(UInt16) Encoded Value = Tenths of seconds since current Hour	if Encoded Value > Tenths of seconds since the current hour at time of receipt, then ValueTenths = tenths of seconds since previous hour. else ValueTenths = tenths of seconds since current hour (Date/Time) Value = Current GMT Date/Time + Value-T This is the "Time of Applicability"

Offset (Byte)	Len (Bytes)	Data Field	Details	Limitations	Example
1	1	Status, Flags	Bits [7..0] [0 0 0 0] [0 0 0 0] Status: Optional: 0 = Undeclared Flags: Reserved NS-Speed Multiplier EW-Speed Multiplier 0 = x0.25, 1 = x0.75	0..15 statuses See Enumerated Field Definitions Multiplier enables speeds up to +- 127 m/s. Only use 1 when speed exceeds 31.75 m/s and add 31.75.	
2	1	Speed North/South	Ground Speed in m/s (+N, -S)	Up to +-95.25 m/s	6 (encoded) =1.5 m/s
3	1	Speed East/West	Ground Speed in m/s (+E,-W) *Note, Heading/Track is derived as a vector of these 2 magnitudes.	Up to +-95.25 m/s	20(enc)= 5 m/s
4	1	Vertical Speed	Vertical Speed m/s from GNSS (+ up, - down) Multiplier = 0.5	Up to +-63 m/s (12.5k ft/min)	15(enc)= 7.5m/s
5	4	Latitude	Latitude of UAV deg *10^7	Int32 signed (LE) (11mm precision)	-48123987
9	4	Longitude	Longitude of UAV deg * 10^7	Int32 signed (LE)	11989298
13	2	Altitude (MSL)	Pressure Altitude (Ref 29.92inhg) Required only if Baro-equipped Altitude/0.5 + 1000m (LE)	-1000-32767m (107503ft) 16bit Signed Int (LE)	1021 (enc) =10.5m
15	2	Geodetic Altitude	WGS84-HAE Altitude/0.5 + 1000m (LE)	-1000-32767m 16bit Signed Int (LE)	1021 (enc) =10.5m
17	2	Height above takeoff	Height above takeoff location Altitude/0.5 + 1000m (LE)	-1000-32767m 16bit Signed Int (LE)	1021 (enc) =10.5m
19	1	Horizontal/Vertical Location Accuracy	Bits [7..0] [0 0 0 0] [0 0 0 0] Vertical (GNSS) Horizontal Vertical: Extended ADS-B GVA Horizontal: Extended ADS-B NACp	See Enumerated Field Definitions	
20	1	Timestamp/ Speed Accuracy	Bits [7..0] [0 0 0 0] [0 0 0 0] Timestamp (* 0.01s) Speed: Based on Extended ADS-B NACv	See Enumerated Field Definitions	
21	2	Timestamp	Time of applicability expressed in 1/10ths of seconds since the last hour.	0-36000: 16 Bit UInt (LE)	3611 = 6mins,1.1s after the hour.
23	2	Reserved			

Figure 3 - Location Message

3.5 Authentication Message

Message Type: 0x2, Static, Optional

The Authentication Message defines a field that can provide a means of authenticity for the identity of the UA sending the message. Depending on the applicable requirements, an Authentication Message may need to be sent. The Auth Type sets the data format and the authentication protocol of the Authentication Data. The Data Page field allows for Authentication Data sizes that may exceed the 24 bytes available per message. The Data Page must be incremented (starting from 0) for each additional message required to complete the oversized message.

Offset (Byte)	Length (Bytes)	Data Field	Details	Limitations	Example
1	1	AuthType, Data Page	Bits [7..0] [0 0 0 0] [0 0 0 0] Auth Type 0: None 1: MPUID 2-9: Reserved for Spec A-F: Available for Private Data Page Start at Page 0.	Up to 16 Types, Up to 16 Pages	
2	23 Bytes	Authentication Data	Opaque Authentication Data Type 1: MPUID	Up to 16 Pages * 23 Bytes	

Figure 4 - Authentication Message

3.6 Self ID Message

Message Type: 0x3, Static

The Self-ID Message is an opportunity for the Remote Pilot to (optionally) declare their identity and purpose of the flight. This message can provide additional information that could reduce the threat profile of a UA flying in a particular area or manner. 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 is a free-form text field. This message is optional.

This message is **optional**

Offset (Byte)	Length (Bytes)	Data Field	Details	Limitations	Example
1	1	Description Type	0: Text Description 1-255: Reserved	0 Only	0
2	23	Operation Description	Text of Operator and Purpose	23 Bytes (UTF-8)	DronesRus: Survey

Figure 5 - Self ID Message

3.7 System Message

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

The System Message general system information including information about the ground control station (GCS). If the GCS can obtain valid location (e.g., GNSS data), the remote ID must broadcast the current location information of the GCS. If the GCS cannot obtain valid location data, the remote ID must broadcast the location of the aircraft's takeoff location. Since this value generally does not change at the same rate of a UA location, the update frequency shall be the same rate as static messages. If a group of aircraft is being represented, the number of aircraft and radius of flight area shall be expressed in this message.

Offset (Byte)	Length (Bytes)	Data Field	Details	Limitations	Example
1	1	Flags	Bits [7..0] [0 0 0 0 0 0 0 0] Reserved Location Source: 0 = Take Off, 1 = Live GNSS		
2	4	Latitude	Latitude deg of operator *10^7	Int32 signed (LE)	-48123987
6	4	Longitude	Longitude deg of operator *10^7	Int32 signed (LE) (11mm precision)	11989298
10	2	Group Count	Number of Aircraft in Group or Formation (default 0)	Up to 65,000 (LE)	
12	1	Group Radius	Radius of Cylindrical area of Group or Formation * 10m	Up to 2.5km	
13	2	Group Ceiling	Group Operations Ceiling WGS84-HAE (MSL) Altitude/0.5 + 1000m	-1000-31767m (107503ft) 16bit UInt (LE)	1021 (enc) =10.5m
15	10	Reserved	Reserved for future use	12 bytes	

3.8 Enumerated Field Definitions

In the data structures above, some fields are enumerated values. The table below shall be used to encode to those enumerations.

Field Name	Details	Length (bits)	Notes
UAS Type	0: None 1: Fixed Wing Powered 2: Rotorcraft/Multirotor 3: LTA (Lighter than Air) Powered 4: LTA Unpowered (Balloon) 5: VTOL (Fixed wing aircraft that can take off vertically) 6: Free Fall/Parachute 7: Rocket 8: Glider 9: Other 10-15: Reserved	4	Up to 16 Types
Status	0: Undeclared 1: Ground, 2: Airborne 3-15: Reserved	4	Up to 16 Statuses
Horizontal Accuracy	0: ≥ 18.52 km (10NM) or Unknown 1: < 18.52 km (10NM) 2: < 7.408 km (4NM) 3: < 3.704 km (2NM) 4: < 1 852 m (1NM) 5: < 926 m (0.5NM) 6: < 555.6 m (0.3NM 7: < 185.2 m (0.1NM 8: < 92.6 m (0.05NM) 9: < 30 m 10: < 10 m 11: < 3 m 12: < 1m 13-15: Reserved	4	This is the same NACp enumeration from ADS-B. Value 12 was added for a more complete range for UAS.
Vertical Accuracy	0: >150m or Unknown 1: <150m 2: <45m 3: <25m 4: <10m 5: <3m 6: <1m 7: Reserved	4	This is the same GVA enumeration from ADS-B. Values 4-6 were added for UAS.
Speed Accuracy	0: >10m/s or Unknown 1: <10m/s 2: <3m/s 3: <1m/s 4: <0.3m/s	4	This is the same enumeration scale and values as ADS-B NACv.

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 and bit packing techniques are not used in these messages to be aligned with what is customary for web service protocols.

```
{
    "Version": "x.x",
    "Response": {
        "BasicID": {
            “IDType”: <UInt8>, // 1 = Serial, 2 = CAA ID, 3 = UTM GUID
            “UASType: <UInt8>,
            “UASID”: “string”
        },
        "Location": {
            “Status”: <Uint8>, // Optional, Default 0
            “SpeedNS”: <Int32 (+-m/s)>,
            “SpeedEW”: <Int32 (+-m/s)>,
            “SpeedVertical”: <Int32 (+-m/s)>,
            “Latitude”: <float>,
            “Longitude”: <float>,
            “AltitudeBaro”: <Int32 (+-meters)>,
            “AltitudeGeo”: <Int32 (+-meters)>,
             "HeightAboveTakeoff": <Int32 (+-meters)>, // Height over TO
            “HAccuracy”: <Int8>,
            “VAccuracy”: <Int8>,
             “SpAccuracy”: <Int8>, // Speed Accuracy
             “TSAccuracy”: <Int8>, // Timestamp accuracy
             “TimeStamp”: <String (RFC3339>
        },
        "Authentication": {
            “AuthType”: <UInt8>, // Default 0 (none)
            “AuthToken”: “string” // Default empty string when AuthType=0
        },
        "SelfID": {
             "Name": "string",
            "Description": "string(25)"
         },
        "Operator": {
            “LocationSource: <Uint8>,
            “Latitude”: <float>,
            “Longitude”: <float>,
            “GroupCount”: <UInt16>,
            “GroupRadius”: <UInt16>,
             “GroupAltitude”: <UInt16>
        }
    }
}
```

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. Optional messages and fields are only optional unless otherwise required by local laws.
3. Interoperability shall be verified against “known working” clients for both BLE 4 and Bluetooth 5 Extended Advertising receivers.
4. Hardware/RF/Signal compliance TBD.
5. If a system is not compliant with this spec, then it may not claim, advertise or display references to “Open Drone ID”.