Enacted on December 1, 2021
Amended on November 28, 2022

Director, Unmanned Aircraft Systems Division

Requirements for remote ID devices and applications

1. Objective

In accordance with the provisions of Article 236-6, paragraph (1), item (ii) of the Aviation Law Enforcement Regulations (Ministry of Transport Ordinance No. 56 of 1952; hereinafter referred to as "Regulations") based on Article 132-5 of the Civil Aeronautics Law, registered unmanned aircraft must be equipped with a remote ID, which is a function for remotely identifying the registration identification of the unmanned aircraft.

The purpose of this requirement is to establish specific requirements that manufacturers must follow when developing and manufacturing Remote ID function installed in unmanned aircraft or external Remote ID device (hereinafter, referred to as "RID equipment"), and the application for register the registration code and other necessary information(hereinafter, referred to as “Application”), according to the obligation to install the remote ID function for displaying the registration symbol based on Article 236-6, paragraph (1), item (ii) of the Regulations.

2. Target

The targets are RID equipment and Applications listed in Article 236-6, paragraph (1), item (ii) of the Regulations.

3. Configuration of requirements

This requirement has the following structure.

(Attach) Direct Remote ID Specification
(Attach 1) Remote ID Equipments Interface Specification
(Attach 2) Application Interface Specification for Manufacturer Application
(Attach3) Notification form of Self-verification Result and Type Information
(Attach4) Notification application form for the Remote ID public key and application authentication code

Supplementary provisions

This requirement will come into effect on June 20, 2022.

This requirement will come into effect on December 5, 2022.

Direct Remote ID Specification

This document is made in Japanese and translated into English. The Japanese text is the original and the English text is for reference purposes. If there is any conflict or inconsistency between these two texts, the Japanese text shall prevail.

MM 2021

Notice:

- Used throughout the specification, Bluetooth is a registered trademark of Bluetooth SIG, Inc.
- Used throughout the specification, Wi-Fi is a registered trademark of Wi-Fi Alliance.

Questions or inquiries regarding this matter will be accepted only by e-mail or in writing.

Unmanned Aircraft Systems Division, Aviation Safety and Security Department,
Civil Aviation Bureau, Ministry of Land, Infrastructure, Transport and Tourism
2-1-3 Kasumigaseki, Chiyoda-ku, Tokyo 100-8918, Japan
hqt-jcab.remoteid@ki.mlit.go.jp

1. General

This Direct Remote ID Specification (hereinafter referred to as the "RID Specification") prescribes the specification to be complied with by manufacturers in the development and manufacture of unmanned aircraft with built-in remote ID or remote ID module (hereinafter referred to as "RID equipments") pursuant to the provisions of Article 132-5, paragraph (1) of the Civil Aeronautics Law and Article 236-6, paragraph(1), item(ii) of the Aviation Law Enforcement Regulations. The scope image is shown in Figure 1.

The manufacturers of RID equipments must develop and manufacture RID equipments in compliance with the RID Specificationso that the registration ID and the cryptographic key informationfor RID encryption, which are notified by the Drone/UAS Information Platform System 2.0 developed and managed by the Civil Aviation Bureau of the Ministry of Land, Infrastructure, Transport and Tourism (hereinafter referred to as the "Registration System") must be written via a smartphone application developed and managed by the Civil Aviation Bureau of the Ministry of Land, Infrastructure, Transport and Tourism (hereinafter referred to as the "JCAB App") or an application developed and managed by a manufacturer of RID equipments connected to the Registration System (hereinafter referred to as the "Manufacturer App").

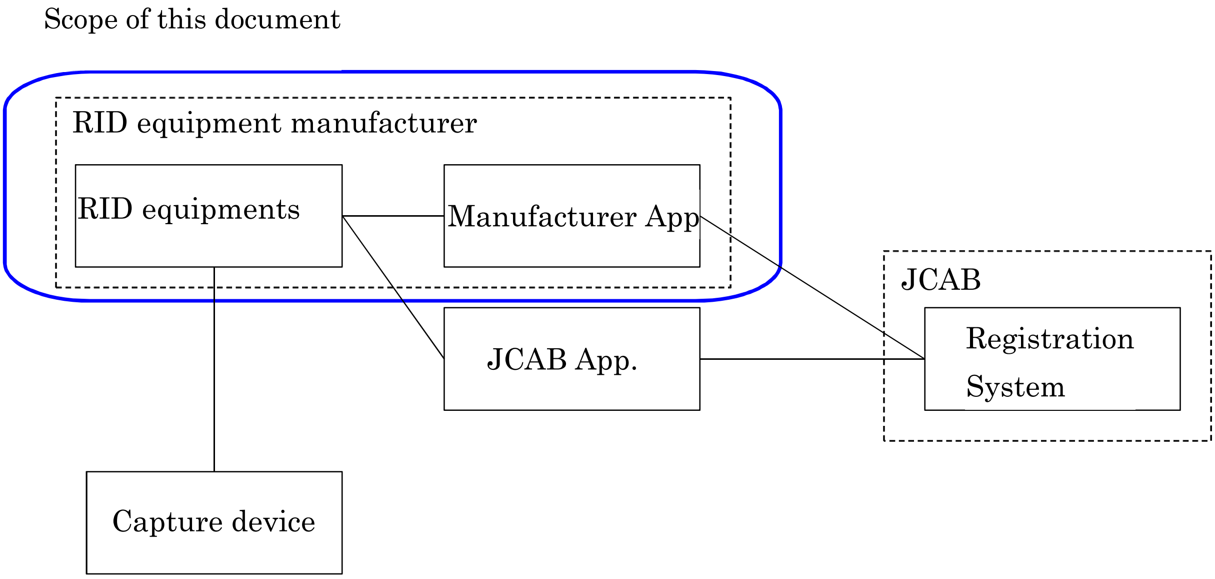


Figure 1 Scope image

2. Performance requirements for RID equipments

(1) Remote ID signal (hereinafter referred to as "RID signal") must be directly transmitted by wireless method from RID equipments by Bluetooth 5.x Long Range (hereinafter referred to as "Bluetooth 5.x"), Wi-Fi Neighbor Awareness Networking (hereinafter referred to as "Wi-Fi Aware"), or Wi-Fi Beacon.

(2) The RID signal must contain at least the following information according to "3. Data format of RID signal".

(i) Registration ID notified under the provisions of Article 132-4 paragraph (3) of the Civil Aeronautics Act

(ii) Serial number specified by the manufacturer

(iii) Location, speed, and timestamp information, etc.

(iv) Authentication information

(3) The RID signal transmission cycle must be at least once every second for all of (i) through (iv), and must be such that it is automatically transmitted continuously while the unmanned aircraft is in flight.

In addition, dynamic information such as location, speed and timestamp must be transmitted within one second after the information is acquired.

(4) The equivalent isotropic radiated power (EIRP) of the transmitted radio wave of the RID signal must satisfy the following. The RID signal should preferably be able to be received from a distance of 300 meters or more in horizontal distance under ideal conditions.

・ In case of Bluetooth 5.x, +5dbm or more

・ In case of Wi-Fi Aware or Wi-Fi Beacon, +11dbm or more

However, the EIRP must be limited to the maximum value of the technical standards of the Radio Act. (For details, refer to Article 49-20 of the Radio Equipment Regulations of the Radio Act)

(5) The accuracy of location information must be above or equal to the accuracy of GNSS stand-alone positioning (preferably within ±30 m) under ideal conditions.

(6) The RID signal transmission must not be able to be stopped by the unmanned aircraft operator or any other person during the flight of the unmanned aircraft.

3. Data format of RID signal

The RID signal must be transmitted in accordance with 5. Performance Requirements in ASTM International F3411-19 "Standard Specification for Remote ID and Tracking" (hereinafter referred to as "ASTM standard")*. In the ASTM standard, the items described as the mandate must be included in the RID signal, while the items marked Optional are voluntary to be included in the RID signal.

* Wi-Fi Beacon is not included in the ASTM F3411-19 standard, but it can be used with the data format complied with ASTM F3411-19 as the ASTM reviced standard allowes to transmit in that way .

However, the following items must be subject to the following requirements.

(1) For Basic ID Message, both of the following must be transmitted.

・The registration ID notified under the provisions of Article 132-4, paragraph (3) of the Civil Aeronautics Act. “JA.” must be added to the beginning. UAS ID type must be 2. (ex. JA.JU12345ABCDE)

- The serial number specified by the manufacturer. UAS ID type must be 1.

(2) The Authentication Message must be transmitted as a mandatory item. Authentication Type must be 3, Page Count must be 0, Length must be 17, and Timestamp must be 32 bit Unix timestamp in seconds since 00:00:00 01/01/2019 (UTC). Authentication Message Header must be 0 and the message authentication code generated according to the following must be used as Authentication Data. (cf. Table 1)

A) The target data is all message after Basic ID Message. Authentication Message must be included as data in which the value of Authentication Data is filled with 0. And it is necessary to include target data when sending Self-ID Message, System Message and Operator Message arbitrarily. Hence, the target data size must be a multiple of 25.

B) A message authentication code (12 bytes)* must be generated by performing message authentication on the target data of A) using AES-128bit-CCM (Counter with Cipher block chaining Message authentication code).

* Ciphertext generated at the same time (data size is the same as A) above) does not need to be included in the authentication data.

C) The common key (16 bytes) used for a message authentication in B) above must be the one written in the RID equipments in accordance with 4. (2). Nonce (12 bytes) must be the following presented from left to right.

(a) Anything written into RID equipments according to 4.(2) (6 bytes)

(b) Timestamp of the Location/Vector Message (2 bytes)

(c) Timestamp of Authentication Message (4 bytes)

(3) In order to satisfy (1) and (2), the RID signals must be sent together as a single Message Pack according to Figure 2.

(4) When transmitting RID signals using Bluetooth 5.x, the phrase of “If implementing this specification using Bluetooth 5 Long Range, Legacy (ADV_NONCONN_IND) advertisements must (BB50010) be sent, as described in 5.4.6, for backwards compatibility with less capable receivers.” in ASTM Standard 5.4.7.1 shall not be applied.

4. Manufacturing requirements for RID equipments

(1) RID equipments must have received technical standards conformity certification, etc. based on the Radio Act of Japan.

(2) It must be possible to write the registration ID information for RID equipments and the cryptographic key information required to

generate the message authentication code for the authentication message (common key and necessary value to generate Nonce) by either of the following methods.

(i) JCAB App method (RID equipments must comply with Attachment 1 "Remote ID Equipments Interface Specification")

(ii) Manufacturer App method (Manufacturer App must comply with Attachment 2 "Application Interface Specification for Manufacturer Application".)

(3) Cryptographic key information written in accordance with (2) above must be stored in RID equipments after taking measures such as storing it in encrypted form in order to prevent easy theft, falsification, or other attacks by third parties.

(4) The serial number specified by the manufacturer must be entered in advance by the manufacturer of the RID equipments at the time of manufacture. The serial number must be assigned in accordance with ANSI/CTA-2063-A. However, unmanned aircraft that hasn't had a remote ID function and whose serial number hasn't complied with ANSI / CTA-2063-A because the registration regulation wasn't enforced, must maintain that serial number when a remote ID function is provided via updation that firmware.

(5) RID equipments must be designed so that the operator of the unmanned aircraft can confirm that the RID equipments is operating during the pre-flight inspection. It is recommended that the design should allow the operator to know when the RID equipments is operating, and when it fails due to malfunction, etc., even while the unmanned aircraft is in flight.

5. Requirements for manufacturers of RID equipments

(1) Upon completion of the development of RID equipments or Manufacturer App, the manufacturer must notify the Civil Aviation Bureau of the manufacturer's name and the model name (or the application name in case of Manufacturer App) of the equipment in the form of Attachment 3, together with the documents confirming and verifying by oneself that the equipment or Manufacturer Application conforms to the RID Specification. (Including maximum communication distances and things to recognize complying with the Japan Radio Act.)

(2) Upon receipt of a notification under (1) above, the Civil Aviation Bureau publishes the name of the manufacturer and the model name of the RID equipments by an appropriate method.

(3) The manufacturer of RID equipments may, when the notification under (1) above has been accepted, label the RID equipments with the statement that it is RID equipments for which notification has been made to the Civil Aviation Bureau, and sell it.

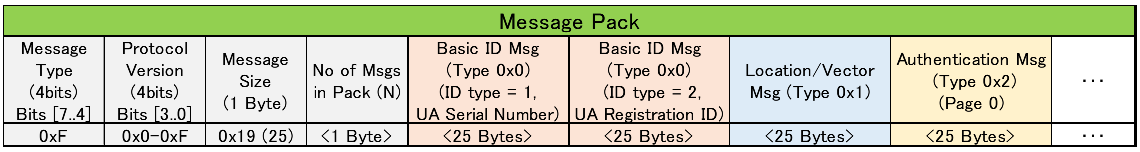
(4) Manufacturers of RID equipments must not develop, manufacture, or

indicate RID equipments that does not conform to the RID Specification.

(5) The manufacturer of RID equipments must not sell the equipment developed and manufactured based on the RID Specification until after the notification in (1) above has been accepted.

Table 1 Authentication Message Details

Offset (bytes)	Length (bytes)	Data Field	Details
1	1	Auth Type, Page Number	Bits [7..0][0000][0000] Auth Type : Bits [7..4] Default value must be 3 : Message Set Signature Page Number : Bits [3..0] Default value shall be 0
2	1	Page Count	Bits [7..0][0000][0000] Reserved : Bits [7..4] Total Page Count : Bits [3..0] Default value must be 0
3	1	Length (bytes)	Total Data Length of concatenation of all Authentication Data fields. Default value must be 17
4	4	Timestamp	32 bit Unix timestamp in seconds since 00:00:00 01/01/2019 (UTC)
8	1	Authentication Message Header	0 : AES-128bit-CCM 1-255 : Reserved Default value must be 0
9	16	Authentication Data	The message authentication code that is generated in accordance with 3.(2)
21	4	Reserved	



Message Pack								
Message Type (4bits) Bits [7..4]	Protocol Version (4bits) Bits [3..0]	Message Size (1 Byte)	No of Msgs in Pack (N)	Basic ID Msg (Type 0x0) (ID type = 1, UA Serial Number)	Basic ID Msg (Type 0x0) (ID type = 2, UA Registration ID)	Location/Vector Msg (Type 0x1)	Authentication Msg (Type 0x2) (Page 0)	・・・
0xF	0x0-0xF	0x19 (25)	<1 Byte>	<25 Bytes>	<25 Bytes>	<25 Bytes>	<25 Bytes>	・・・

Figure 2 Message Pack

Attachement 1

Remote ID Equipments
Interface Specification

This document is made in Japanese and translated into English. The Japanese text is the original and the English text is for reference purposes. If there is any conflict or inconsistency between these two texts, the Japanese text shall prevail.

MM 2021

Attachement 1

Notice:

- Used throughout the specification, Bluetooth is a registered trademark of Bluetooth SIG, Inc.
- Used throughout the specification, Wi-Fi is a registered trademark of Wi-Fi Alliance.

Attachement 1

1. General

(1) This specification specifies the interface specification to be prepared for RID equipments when writing registration ID information and the cryptographic key information (common key and initialization vector) necessary for authentication data generation for authentication message to unmanned aircraft with built-in remote ID and remote ID equipments (hereinafter referred to as “RID equipments”) via a smartphone application developed and managed by the Civil Aviation Bureau of the Ministry of Land, Infrastructure, Transport and Tourism (hereinafter referred to as “Smartphone App”) as shown in “5. Requirements for manufacturers of RID equipments, Direct Remote ID Specification”.

(2) The manufacturer of RID equipments must fill out the application form (Attachment 4) and submit it to the following application window by e-mail. After completing the confirmation of the application information, Civil Aviation Bureau notifies the applicant of the public key for digital signature and application authentication code, etc., which are required for the development and manufacture of RID equipments. When it becomes necessary to change the public key and application authentication code, etc., Civil Aviation Bureau notifies the manufacturer of the RID equipments of the reason for the change, the changed public key and application authentication code, etc., and other necessary information.

【Application window】
Unmanned Aircraft Systems Division, Aviation Safety and Security Department, Civil Aviation Bureau, Ministry of Land, Infrastructure, Transport and Tourism
E-mail：hqt-jcab.remoteid@ki.mlit.go.jp

2. Communication requirements between Smartphone App and RID equipments

(1) Communication between Smartphone App and RID equipments must be made in Bluetooth Low Energy communication mode specified in Bluetooth 4.x using the 1M PHY physical layer.

(2) RID equipments must operate as “Peripheral” specified in the Generic Access Profile (hereinafter referred to as "GAP") defined in the Bluetooth Low Energy specification and for connecting, they must send an advertising packet to the surroundings to make the existence of RID equipments known.

(3) Smartphone App operates as “Central” specified in GAP, discovering the advertising packet transmitting from RID equipments, and granting permission for connection.

(4) RID equipments and Smartphone App confirmed the connection

between them must be paired based on LE Secure Connections (using Just Works), and communication must be encrypted after pairing.

(5) RID equipments whose connection has been confirmed must operate as a Server specified in the Generic Attribute Profile (hereinafter referred to as "GATT") defined in the Bluetooth Low Energy specification, and provide services to access attribute information held by RID equipments. The definition of the services that must be provided by RID equipments is specified in “4. Service configuration of RID equipments”.

(6) Smartphone App whose connection has been confirmed operates as a Client defined in GATT, and through a series of procedures (sequence) with RID equipments achieve writing registration ID information to RID equipments. The sequence when writing information between RID equipments and Smartphone App is specified in “5. Sequence when writing to RID equipments”.

(7) Strings or IDs and binary data must be sent and received by the network byte order which is read from left to right, and in the order of most significant bite (MSB) to least significant bite (LSB), except for those whose size is indicated by the magnitude of the numerical value. “Those whose size is indicated by the magnitude of the numerical value” is a number indicated as a 16- or 32-bit integer (latitude, longitude, altitude, time stamp, etc.), with the LSB on the left and the MSB on the right and is treated as "little endian". (Hereinafter referred to as "LE" in the following description in this specification).

3. Requirements for RID equipments

For RID equipments, following requirements shall be realized.

(1) RID equipments must have a function to switch between the mode of writing registration ID specified in this specification and the mode of transmitting the remote ID signal described in the Direct Remote ID Specification.

(2) When pairing with Smartphone App, the target RID equipments must be identifiable by its serial number.

(3) If it becomes necessary to change the public key and application authentication code, etc. notified by the MLIT in 1.(2) above, the firmware must be updated to accommodate the change. It can also be done in a different way.

4. Service configuration of RID equipments

RID equipments must provide the services listed in Table 1 in accordance with the provisions of GATT. As shown in Table 1, this service provides three types of access: RID Auth (access to attributes related to application authentication), RID Command (access to attributes related to

Command instructions to RID equipments), and RID Response (access to attributes related to Responses to Command instructions to RID equipments).

Table 1:　Service configuration of RID equipments

Classification	Type	UUID	Permission	Value	Value Size (bytes)
Service declaration	Declaration	0x2800	Read	→Table 2：Remote ID Service UUID	16
Characteristic 1	Declaration	0x2803	Read	Prop=Write	1
	Value	→Table 2：Remote ID Auth UUID	Encryption Write	(Application authentication code)	32
	Description	0x2901	Read	RID Auth	–
Characteristic 2	Declaration	0x2803	Read	Prop=Write	1
	Value	→Table 2：Remote ID Command UUID	Encryption Write	→Table 3：Frame format of Command	176
	Description	0x2901	Read	RID Command	–
Characteristic 3	Declaration	0x2803	Read	Prop=Notify	1
	Value	→Table 2：Remote ID Response UUID	–	→Table 4: Frame format of Response	176
	CCCD	0x2902	Encryption Read/Write	bit0 0=Notification disabled 1=Notification enabled	2
	Description	0x2901	Read	RID Response	–

Table 2:　List of UUIDs for services by RID equipments

Type	Size	UUID
Remote ID Service UUID	128 bit	f9ed6165-faa8-4f2d-8b82-dc67d3444b0f
Remote ID Auth UUID	128 bit	aacf388f-0e69-4802-8067-3508b1b50c3a
Remote ID Command UUID	128 bit	2d67083e-5291-4dfa-a357-8ae4317413f5
Remote ID Response UUID	128 bit	d98c42d8-3013-462e-8d35-2b5b61eea94d

5. Sequence when writing to RID equipments

When processing the writing of registration ID information, etc. to RID equipments, the process must be called according to the sequence shown below.

(1) Connection process to RID equipments

① Operate RID equipments, switch it to a mode where it can be connected to Smartphone App, and start advertising.

② Scan Service UUID by Smartphone App and discover the RID equipments transmitting Remote ID Service UUID. In addition, discover the RID device with the CompleteLocalName-serial number as the filtering condition.

③ Pair the discovered RID equipments with the smartphone.

④ After pairing, start communication according to the mechanism specified in GATT.

Attachement 1

⑤ Write to RID Auth characteristic of RID equipments from Smartphone App. In that case, if a value different from the application authentication code of RID equipments, notified in advance by the MLIT in 1. (2) is written, disconnect the connection.

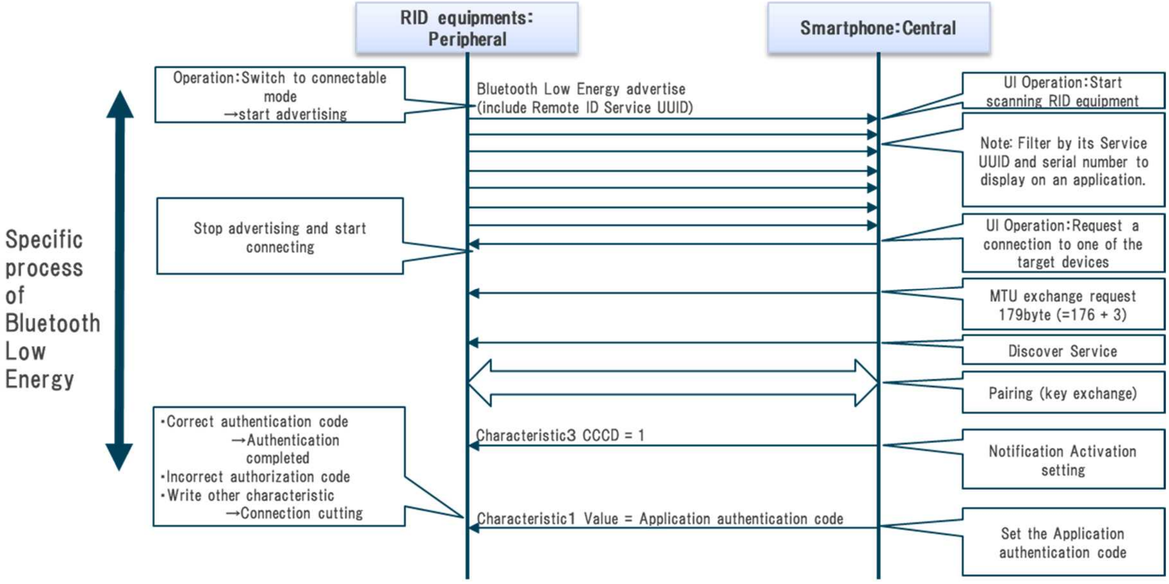

RID equipments: Peripheral
Smartphone: Central
Specific process of Bluetooth Low Energy
Operation: Switch to connectable mode →start advertising
Bluetooth Low Energy advertise (include Remote ID Service UUID)
UI Operation: Start scanning RID equipment
Note: Filter by its Service UUID and serial number to display on an application.
Stop advertising and start connecting
UI Operation: Request a connection to one of the target devices
MTU exchange request 179byte (=176 + 3)
Discover Service
Pairing (key exchange)
・Correct authentication code →Authentication completed
・Incorrect authorization code
・Write other characteristic →Connection cutting
Characteristic3 CCCD = 1
Notification Activation setting
Characteristic1 Value = Application authentication code
Set the Application authentication code

Fig.1: Connection processing sequence to RID equipments

(2) Command processing (normal system)

Assuming that the connection to RID equipments has been successfully completed and that communication between RID equipments and Smartphone App has been established, the process must be called according to the sequence shown below.

① RID equipments receive Write Commands from Smartphone App with RID Command characteristic.

② When written successfully, RID equipments perform processing based on the contents of the Command.

③ RID equipments write the processing result to RID Response characteristic and notify Smartphone App.

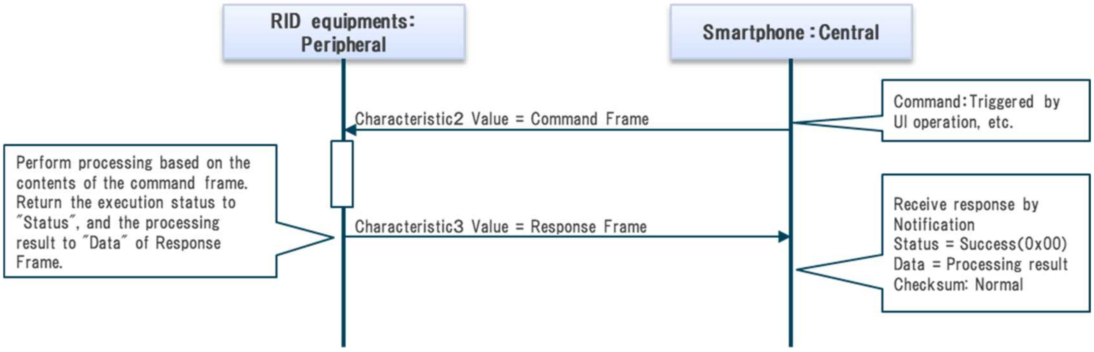

RID equipments: Peripheral
Smartphone : Central
Command: Triggered by UI operation, etc.
Characteristic2 Value = Command Frame
Perform processing based on the contents of the command frame. Return the execution status to "Status", and the processing result to "Data" of Response Frame.
Characteristic3 Value = Response Frame
Receive response by Notification
Status = Success(0x00)
Data = Processing result
Checksum: Normal

Fig.2: Sequence of Command processing (normal system)

(3) Command processing (in case of processing error)

If an abnormality occurs during the Command processing, the process must be interrupted by RID equipments according to the sequence shown below.

① RID equipments receive Write Command from Smartphone App with RID Command characteristic.

② When successfully written, RID equipments perform processing based on the contents of the Command.

③ If the processing result is an error, write the error contents to RID Response characteristic and notify Smartphone App.

④ RID equipments are disconnected after displaying the error contents on SmartPhone APP.

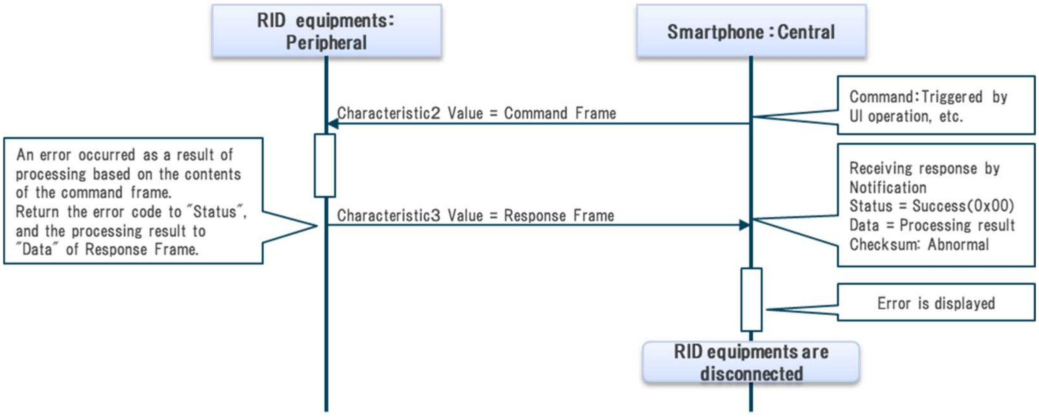

RID equipments: Peripheral
Smartphone : Central
Characteristic2 Value = Command Frame
Command: Triggered by UI operation, etc.
An error occurred as a result of processing based on the contents of the command frame.
Return the error code to "Status", and the processing result to "Data" of Response Frame.
Characteristic3 Value = Response Frame
Receiving response by Notification
Status = Success(0x00)
Data = Processing result
Checksum: Abnormal
Error is displayed
RID equipments are disconnected

Fig.3: Sequence of Command processing (in case of processing error)

The error code must return 0x01 for an error in Command arguments and 0x02 for other internal errors. For the description of the error results, the trace of the error event that was confirmed internally by RID equipments must be returned in the range of 172 bytes.

(4) Command processing (in case of Response error)

If there is an abnormality in the Response to the Command processing result, it is judged that there is a high possibility that an abnormality has occurred in the communication path, and the processing must be interrupted according to the sequence shown below.

① RID equipments receive Write Command from Smartphone App with RID Command characteristic.

② When successfully written, RID equipments perform processing based on the contents of the Command.

③ RID equipments write the processing results to the Response characteristic and notify Smartphone App.

④ If the checksum of the Response is abnormal, RID equipments are disconnected after displaying an error.

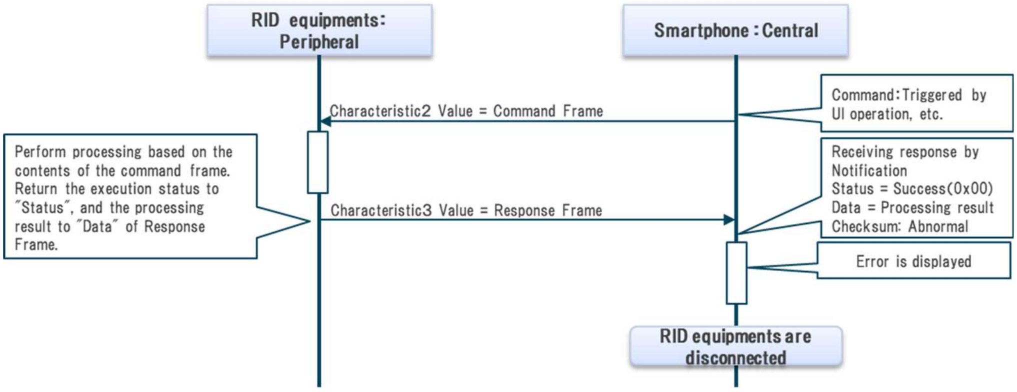

RID equipments: Peripheral
Smartphone : Central
Characteristic2 Value = Command Frame
Command: Triggered by UI operation, etc.
Perform processing based on the contents of the command frame. Return the execution status to "Status", and the processing result to "Data" of Response Frame.
Characteristic3 Value = Response Frame
Receiving response by Notification
Status = Success(0x00)
Data = Processing result
Checksum: Abnormal
Error is displayed
RID equipments are disconnected

Fig.4: Sequence of Command processing (in case of Response error)

(5) Command processing (in case of communication error)

If a Bluetooth Low Energy communication error is detected during the Command processing, a certain number of retries must be performed, and if the process is not completed even after repeated retries, the process must be interrupted according to the sequence shown below.

① Write Command from Smartphone App to RID Command characteristic of RID equipments.

② If failed due to a communication error, retry a certain number of times.

③ If the retry is not successful, the error message is displayed and RID equipments are disconnected.

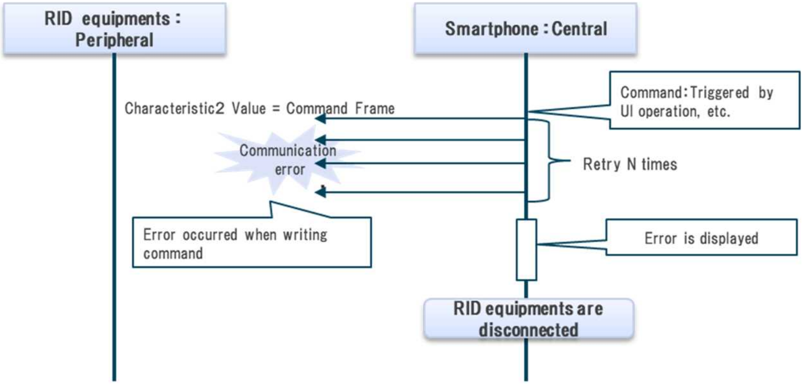

RID equipments : Peripheral
Smartphone : Central
Characteristic2 Value = Command Frame
Command: Triggered by UI operation, etc.
Communication error
Retry N times
Error occurred when writing command
Error is displayed
RID equipments are disconnected

Fig.5: Sequence of Command processing (in case of communication error)

(6) Command processing (at timeout)

If, a state in which no Response is returned for some reason is detected during the Command processing, it is judged that there is a high possibility that a failure has occurred in the communication path or RID equipments, and the processing must be interrupted according to the sequence shown below.

① RID equipments receive Write Command from Smartphone App with RID Command characteristic.

② If SmartPhone APP receive no Response within a certain period of time after the command, RID equipments is disconnected after the error content is displayed.

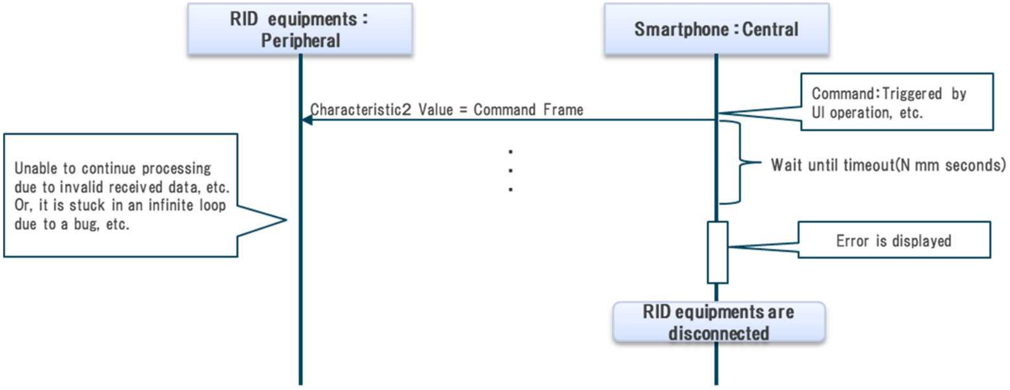

RID equipments : Peripheral
Smartphone : Central
Characteristic2 Value = Command Frame
Command:Triggered by UI operation, etc.
Wait until timeout(N mm seconds)
Unable to continue processing due to invalid received data, etc. Or, it is stuck in an infinite loop due to a bug, etc.
Error is displayed
RID equipments are disconnected

Fig.6: Sequence of Command processing (at timeout)

(7) Disconnection process with RID equipments

If it becomes necessary to disconnect from RID equipments, the process must be interrupted according to the sequence shown below.

① RID equipments is instructed to perform the disconnection process by Smartphone App.

② Following the disconnection instruction, RID equipment is disconnected.

③ If necessary, RID equipments switch to a mode that allows Remote ID to be transmitted.

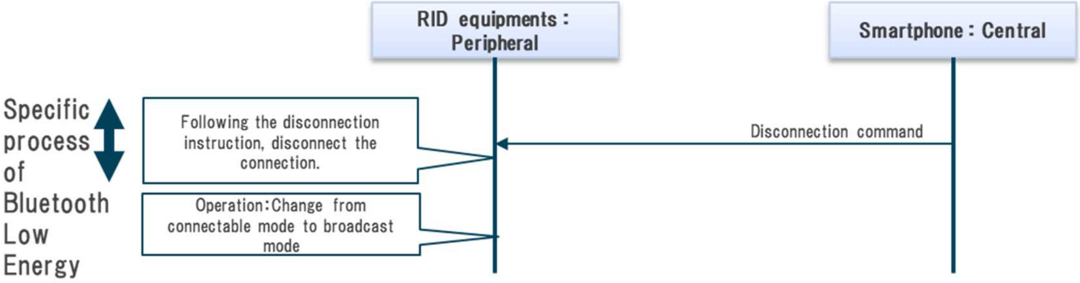

RID equipments : Peripheral
Smartphone : Central
Specific process of Bluetooth Low Energy
Following the disconnection instruction, disconnect the connection.
Disconnection command
Operation:Change from connectable mode to broadcast mode

Fig.7: Disconnection processing sequence with RID equipments

6. Frame format for communication with RID equipments

Communication between RID equipments and Smartphone App must be performed by reading and writing values in a 176-byte frame of attribute values indicated by Remote ID Command UUID/Remote ID Response UUID, as described in Table 1 Service configuration of RID equipments. Respective frame formats for Command/Response are shown below.

Attachement 1

(1) Frame format of Command

Table 3 shows the frame format of Command.

Table 3: Frame format of Command

Offset	Size(bytes)	Data	Remarks
0	1	Sequence Number	Incremented for each Command issued
1	1	Command ID	0x01 RID writing 0x02 RID inquiry
2	1	Reserved	—
3	172	（Data for each Command）	→7. Described in the data definition for each Command
175	1	Check Sum	Sum of Offset 0-174

(2) Frame format of Response

Table 4 shows frame format of Response.

Table 4: Frame format of Response

Offset	Size(bytes)	Data	Remarks
0	1	Sequence Number	Sequence Number stored in Command frame
1	1	Command ID	Command ID stored in Command frame
2	1	Status	Success：0x00 Errors in Command arguments：0x01、Other internal errors::0x02
3	172	（Data for each Response）	→7. Described in the data definition for each Command
175	1	Check Sum	Sum of Offset 0-174

Attachement 1

7. Data definition for each Command

(1) RID writing Command

① Definition of Command data

Command data definition of RID writing Command is shown in Table 5.

When erasing the RID information, information other than the serial number must be transmitted as 0x00.

Table 5: Command data definition for RID writing Command

Offset	Size (bytes)	Endian	Data	Description	Remarks
0	1	–	Key type	Type of key used for authentication code generation	0x00:Unregistered 0x01: Indicates that the key is for AES-CCM (128bit/little endian) Other values are undefined (add values when adding authentication methods in the future). Generated by the server side
1	1	–	UA type	Type of airframe Obtained from the server side	RID Specification：UA type of Basic ID Message
2	15	–	Registration ID	”JA” shall be added to the beginning of registration ID issued by the government Obtained from the server side	RID Specification：UAS ID of Basic ID message, ID type = 2
17	20	–	Serial number	Serial number of RID equipments (A writing error will occur if the serial number does not match the factory-set serial number) Obtained from the server side	RID Specification：UAS ID of Basic ID message, ID type=1 If the serial number is shorter than 20 digits, the back is filled with 0x00
37	4	LE	Start	Starting date of registration validity period	Elapsed seconds since 2019/1/1 00:00:00 (RID Specification：Authentication Message page 0：Same generation method as timestamp) Generated by the server side
41	4	LE	Expire	Expiration date of registration validity period	
45	32	–	Key information	Key information	Key information corresponding to the key type In the case of AEC-CCM, the 16 bytes after the key information are filled with 0x00 Generated by the server side
77	23	–	Reserved	–	All to be filled with 0x00
100	72	–	Signature information	Digital signature generated from the data from Offset 0-99	SHA-256 for the hash algorithm Binary data through DER-encoding the digital signature generated by ECDSA using P-256 curve at the server side If it is shorter than 72 bytes, it will be suffixed with 0x00

② Checking the writing result

It should be necessary to verify the authenticity of the data by collating the result of decrypting the signature information using the public key notified in advance by the MLIT in 1. (2) with the hash value obtained using the SHA-256 algorithm from the portion of the Command data excluding the signature information.

③ Definition of Response data

When the process is completed normally, RID equipments must return data stuffed with 0x00 to Smartphone App as the Response data.

In the event of an error, RID equipments must return data describing the error content in the range of 172 bytes to Smartphone App as the Response data.

Attachement 1

(2) RID inquiry Command

① Definition of Command data

RID equipments must return data stuffed with 0x00 to Smartphone App.

② Definition of Response data

When the process is completed normally, RID equipments must return the Response data shown in Table 6: Response data definition for RID inquiry Command to Smartphone App.

Table 6: Response data definition for RID inquiry Command

Offset	Size (bytes)	Endian	Data	Description	Remarks
0	1	–	Key type	Type of key	0x00:Unregistered 0x01: Indicates that the key is for AES-CMAC (128bit/little endian) Other values are undefined (add values when adding authentication methods in the future).
1	1	–	UA type	Type of airframe	RID Specification：UA type of Basic ID Message
2	15	–	Registration ID	"JA" shall be added to the beginning of registration ID issued by the government	RID Specification：UAS ID of Basic ID message, ID type = 2
17	20	–	Serial number	The serial number of the RID equipments issued by the manufacturer	RID Specification：UAS ID of Basic ID message, ID type=1 If the serial number is shorter than 20 digits, the back is filled with 0x00
37	4	LE	Start	Starting date of registration validity period	Elapsed seconds since 2019/1/1 00:00:00 （RID Specification：Authentication Message page 0：Same data format as timestamp）
41	4	LE	Expire	Expiration date of registration validity period	
45	127	–	Reserved		ALL 0x00

In the event of an error, RID equipments must return data describing the error content in the range of 172 bytes to Smartphone App as the Response data.

Attachment 2

Application Interface Specification for Manufacturer Application

This document is made in Japanese and translated into English. The Japanese text is the original and the English text is for reference purposes. If there is any conflict or inconsistency between these two texts, the Japanese text shall prevail.

MM 2021

Attachment 2

Notice:

- Used throughout the specification, Bluetooth is a registered trademark of Bluetooth SIG, Inc.
- Used throughout the specification, Wi-Fi is a registered trademark of Wi-Fi Alliance.

Attachment 2

1. General

(1) This specification specifies the requirements for applications developed and managed by manufacturers of RID equipments (hereinafter referred to as "Manufacturer App") that are connected to the unmanned aircraft registration system developed and managed by the Civil Aviation Bureau of the Ministry of Land, Infrastructure, Transport and Tourism as shown in "5. Requirements for manufacturers of RID equipments, Direct Remote ID Specification", and the interface specification which is required when obtaining information necessary for writing the Registration ID to RID equipments from the registration system and when storing the writing result of the registration ID to the registration system.

(2) The manufacturer of Manufacturer App must fill out the application form (Attachment 4) and submit it to the following application window by e-mail. After completing the confirmation of the application information, Civil Aviation Bureau will notify the applicant of the public key for digital signature and application authentication code, etc., which are required for the development and manufacture of Manufacturer App. When it becomes necessary to change the public key and application authentication code, etc., Civil Aviation Bureau will notify the manufacturer of the Manufacturer App of the reason for the change, the changed public key and application authentication code, etc., and other necessary information.

【Application window】

Unmanned Aircraft Systems Division, Aviation Safety and Security Department, Civil Aviation Bureau, Ministry of Land, Infrastructure, Transport and Tourism
E-mail：hqt-jcab.remoteid@ki.mlit.go.jp

2. Communication requirements between Manufacturer application and the registration system

(1) Manufacture App and the registration system must communicate via the internet, encrypted by the https protocol.

(2) When Manufacturer App connects to the registration system, user authentication must be performed on the authentication infrastructure provided by the registration system using the user ID/PW of the registration system. Open ID Connect must be used as the authentication method.

(3) The API provided by the registration system is a Web API in REST format, and the specification is described and published in Open API format.

Attachment 2

3. Requirements for Manufacturer application

(1) The registration ID and authentication information must be accessible only by the user or the application, and must not be accessible by third parties. In the case of Android, the registration ID and authentication information must not be placed in the /sdcard area.

(2) If it becomes necessary to change the public key and application authentication code, etc. notified by the MLIT in 1.(2) above, the firmware must be updated to accommodate the change. It can also be done in a different way.

4. Remote ID registration sequence

Figure 1 shows the series of processing flow that Manufacturer App obtains the registration ID information and cryptographic key information from the registration system, writes the information to RID equipments, and stores the writing result in the registration system.

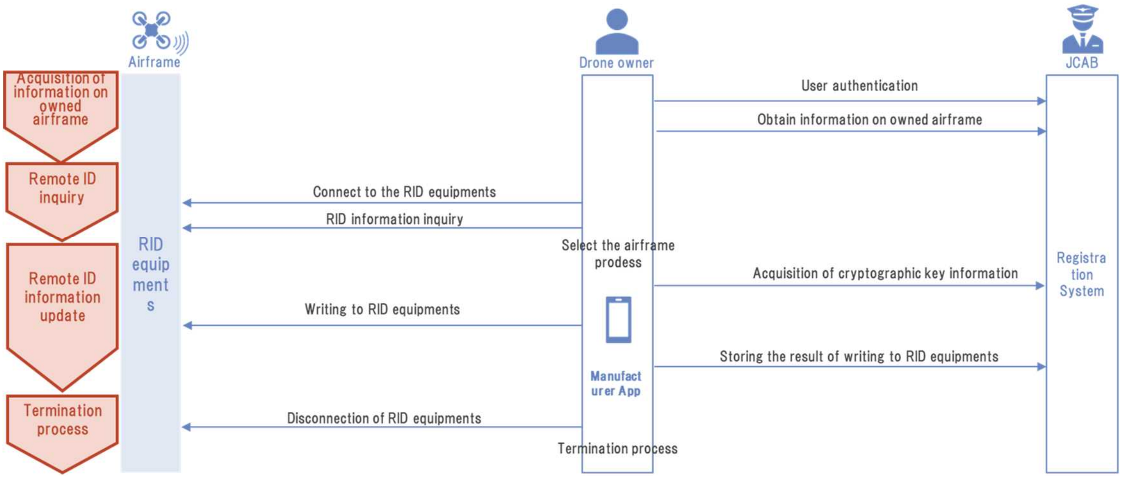

Airframe
Drone owner
JCAB
Acquisition of information on owned airframe
Remote ID inquiry
Remote ID information update
Termination process
RID equipments
User authentication
Obtain information on owned airframe
Connect to the RID equipments
RID information inquiry
Select the airframe prodess
Acquisition of cryptographic key information
Writing to RID equipments
Storing the result of writing to RID equipments
Manufacturer App
Disconnection of RID equipments
Termination process
Registration System

Fig.1: Remote ID registration sequence

(1) Acquisition of information on owned airframe

Connect to the registration system to obtain a list of information about the unmanned aircraft you own.

① User authentication by the registration system

User authentication must be performed using the authentication function provided by the authentication infrastructure of the registration system. Based on the access right obtained here, access to the following registration system.

② Obtain information on owned airframe

Obtain information about the airframe owned by the user (the registration ID, manufacturer, model and serial number of the airframe, and serial number of RID equipments)

(2) Remote ID inquiry

Connect to the RID equipments to be written and inquire the information of the remote ID before writing.

① Connection to RID equipments
Connect Manufacturer App to RID equipments.

② RID information inquiry
Inquire about the Registration ID written on RID equipments and the serial number of RID equipments.

(3) Remote ID information update

Update remote ID information by linking the airframe information on the registration system with RID equipments connected to Manufacturer App.

① Select the registered airframe
Select the airframe on the registration system to be updated and RID equipments on a one-to-one basis. When selecting, the serial number of RID equipments must be checked to prevent writing the registration ID information to wrong RID equipments.

② Acquisition of cryptographic key information
Acquire cryptographic key information from the registration system. Cryptographic key information must be acquired only when writing to RID equipments is required.

③ Writing to RID equipments
Update the registration ID information of RID equipments.

④ Storing the result of writing to RID equipments
The result of writing registration ID information to RID equipments must be stored in the registration system.

(4) Termination process

After the registration is completed, the termination process must be executed. This process must be executed even when a series of processing procedures are terminated in the middle due to errors.

① Disconnection of RID equipments
If necessary, disconnect from RID equipments and return the RID equipments to the unconnected state.

② Termination process
The deletion process must be performed so that no cryptographic key information must remain inside Manufacturer App.

5. Definition of manufacturer application interface with the registration system

Manufacturer App and the registration system must use the application interface shown below to process the writing of the registration ID information.

(1) Entire sequence

Manufacturer App and the registration system must conduct an Open ID Connect-compliant authentication process on the authentication infrastructure provided by the registration system, and must request various APIs according to the authority of the authenticated user by using the access token obtained there. Entire sequence is shown in Figure 2.

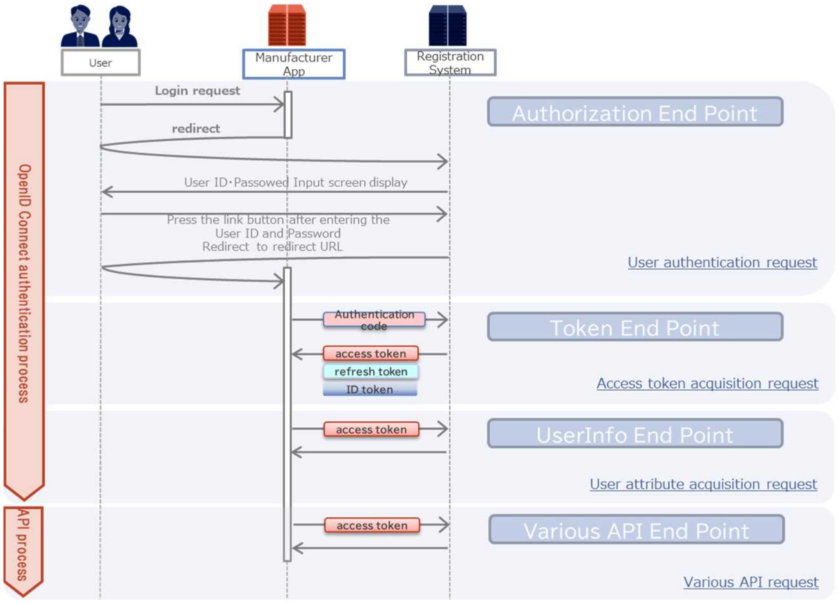


Fig.2: Entire sequence

(2) User authentication

For user authentication, the authentication infrastructure provided by the registration system must be used.

The authentication process must comply with Open ID Connect and applies an extension (RFC7636: Proof Key for Code Exchange by OAuth Public Clients) to prevent leakage of authentication information within the same device.

① Sequence of the authentication process

The sequence of the authentication process is described in Figure 3.

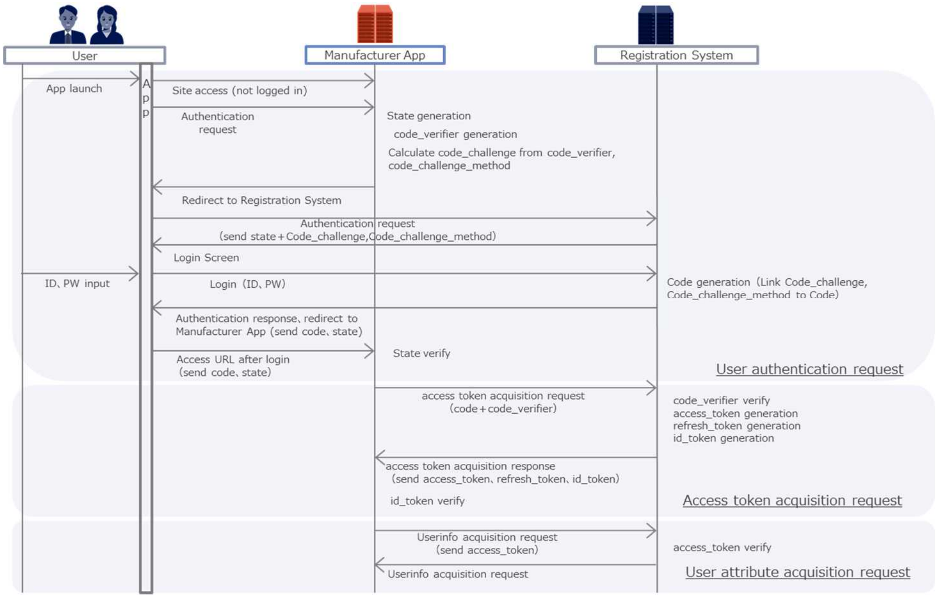

User
Manufacturer App
Registration System
App launch
App
Site access (not logged in)
Authentication request
State generation
code_verifier generation
Calculate code_challenge from code_verifier, code_challenge_method
Redirect to Registration System
Authentication request
(send state＋Code_challenge,Code_challenge_method)
Login Screen
ID、PW input
Login（ID、PW）
Code generation（Link Code_challenge, Code_challenge_method to Code）
Authentication response、redirect to Manufacturer App (send code、state)
Access URL after login (send code、state)
State verify
User authentication request
access token acquisition request (code＋code_verifier)
code_verifier verify
access_token generation
refresh_token generation
id_token generation
access token acquisition response (send access_token、refresh_token、id_token)
id_token verify
Access token acquisition request
Userinfo acquisition request (send access_token)
access_token verify
Userinfo acquisition request
User attribute acquisition request

Fig.3: Authentication sequence with Open ID Connect (PKCE extension)

Attachment 2

② Request processing for authentication

Request processing for authentication (user authorization, access token acquisition and user attribute acquisition processing) must be performed in accordance with the provisions of Open ID Connect.

The request URL in the operational environment will depend on the specification of the registration system.

(3) APIs to be provided for writing registration ID information

APIs to be provided for writing registration ID information are shown in Table 1.

Table 1: APIs to be provided for writing registration ID information

Name of API	Type	API path	Contents
Acquisition of list of owned airframe information	GET	/rid/aircrafts	Acquire a list of airframe information owned by the user. Based on the acquired airframe information, confirm the airframe specifications required for RID writing.
Acquisition of information on owned airframe	GET	/rid/aircrafts/{registration_code}	Acquire information on the user’ s airframe, using the registration ID as a key.
Acquisition of cryptographic key information	GET	/rid/remoted	Acquire the cryptographic key information required to generate the authentication data for the authentication message. (Cryptographic key can be recreated by specifying parameters)
Store remote ID writing result	POST	/rid/remoted	Store the result of writing to RID equipments in the registration system.

In the future, if API changes that are not backward compatible occur, as version control, insert "v2", "v3" ... into the API path. In this specification omitted as "v1".
ex) In the case of the acquisition of list of owned airframe information AIP “/rid/v2/aircrafts”

① Request

The request parameters for each API are described in the individual API definition.

When requesting each API, the access token obtained from (2) User Authentication must be given to the Authorization: Bearer header in order to restrict the API by user authority.

In addition, the request URLs in the operational environment depend on the specification of the registration system.

② Response

When the request is successful, the response described in the individual API definition is returned. When an error occurs in the registration system, the response code shown in Table 2 and the following response body shown in Table 3 must be returned.

Attachment 2

Table 2: Response Codes for Errors

HTTP status	Meaning	Contents
400	Request parameter error	Processing failure (invalid parameter)
500	System error in API	Unexpected system error

Table 3: Response body in case of error

Item name	Parameter name	Data type	Required	Contents
Error code	error Code	Character string	○	Error code
Error message	error Message	Character string	○	Detailed description of the error

(4) Definition of the API for obtaining a list of owned airframe information

Obtain a list of drone airframe information owned by the applicant.

① Request parameter

None

② Response body

Table 4 shows the definition of the response body when a request to the API for obtaining a list of owned airframe information is successful.

Attachment 2

Table 4: Response body when the API for obtaining a list of owned airframe information is successful

Item name	Parameter name	Data type	Required	Contents
Airframe information		Array	○	Array of airframe information 0～N
Registration ID	registration_code	Character string	○	Registration ID issued by the government
Manufacturing category	manufacturing category	Character string	○	Either of the following values 1 ：Manufacturer's airframe/modified airframe 2 ： Home-built airframe
Manufacturer in Japanese	manufacturer_jpn	Character string	○	Name of the manufacturer of the unmanned aircraft (Japanese)
Model in Japanese	model_jpn	Character string	○	Model name of unmanned aircraft (Japanese)
Manufacturer in English	manufacturer_eng	Character string	○	Name of the manufacturer of the unmanned aircraft (English)
Model in English	model_eng	Character string	○	Model name of unmanned aircraft (English)
Serial number	manufacturing_number	Character string	○	Serial number of unmanned aircraft
Remodeling or not	remodeling_type	Character string		1 ： With modifications 2 ： No modifications
Type	aircraft_type	Character string	○	One of the following values 1 ： Airplane 2 ： Rotary wing aircraft (helicopters) 3 ： Rotary wing aircraft (multirotor) 4 ： Rotary wing aircraft (other) 5 ： Glider 6 ： Airship
With or without RID	rid_type	Character string	○	Does the airframe have RID? 0 ： No 1 ： Yes (built-in type) 2 ： Yes (external type)
External type RID equipments manufacturer in Japanese	rid_manufacturer_jpn	Character string		Manufacturer's name of external type RID equipments (in Japanese). In the case of built-in type RID, the same as the manufacturer of the unmanned aircraft
Model of external type RID equipments in Japanese	rid_model_jpn	Character string		Model name of external type RID equipments (in Japanese) In the case of built-in type RID, the same as the model of unmanned aircraft
External type RID equipments manufacture in English	rid_manufacturer_eng	Character string		Manufacturer's name of external type RID equipments (in English) In the case of built-in type RID, the same as the manufacturer of the unmanned aircraft
Model of external type RID equipments in English	rid_model_eng	Character string		Model name of external type RID equipments (in English) In the case of built-in type RID, the same as the model of unmanned aircraft
Serial number of external type RID equipments	rid_manufacturing_number	Character string		Serial number of external type RID equipments In the case of built-in type RID, the same as the serial number of the unmanned aircraft
Date and time of update	modified_date	Character string	○	Date and time of update(UTC) Return an empty string when initially registered. YYYY-MM-DDThh:mm:ssZ format.
Writing flag	write_status	Character string	○	Status of writing registration ID information to RID equipments 0 ： Unwritten 1 ： Written

(5) Definition of the API for obtaining information on owned airframe
Using the registration ID as a key, obtain information on one drone airframe owned by the applicant.

Attachment 2

① Request parameters

None

② Response body

Table 5 shows the definition of the response body when a request for the API for obtaining a list of owned airframe information is successful.

Table 5: Response body when the API for obtaining information on owned airframe is successful.

Item name	Parameter name	Data type	Required	Contents
Registration ID	registration_code	Character string	○	Registration ID issued by the government
Manufacturing category	manufacturing_category	Character string	○	Either of the following values 1 :Manufacturer's airframe/modified airframe 2 : Home-built airframe
Manufacturer in Japanese	manufacturer_jpn	Character string	○	Name of the manufacturer of the unmanned aircraft (Japanese)
Model in Japanese	model_jpn	Character string	○	Model name of unmanned aircraft (Japanese)
Manufacturer in English	manufacturer_eng	Character string	○	Name of the manufacturer of the unmanned aircraft (English)
Model in English	model_eng	Character string	○	Model name of unmanned aircraft (English)
Serial number	manufacturing_number	Character string	○	Serial number of unmanned aircraft
Remodeling or not	remodeling_type	Character string		1 : With modifications 2 : No modification
Type	aircraft_type	Character string	○	One of the following values 1 : Airplane 2 : Rotary wing aircraft (helicopters) 3 : Rotary wing aircraft (multirotor) 4 : Rotary wing aircraft (other) 5 : Glider 6 : Airship
With or without RID	rid_type	Character string	○	Does the airframe have RID? 0 : No 1 : Yes (built-in type) 2 : Yes (external type)
External type RID equipments manufacturer in Japanese	rid_manufacturer_jpn	Character string		Manufacturer's name of external type RID equipments (in Japanese). In the case of built-in type RID, the same as the manufacturer of the unmanned aircraft
Model of external type RID equipments in Japanese	rid_model_jpn	Character string		Model name of external type RID equipments (in Japanese) In the case of built-in type RID, the same as the model of unmanned aircraft
External type RID equipments manufacture in English	rid_manufacturer_eng	Character string		Manufacturer's name of external type RID equipments (in English) In the case of built-in type RID, the same as the manufacturer of the unmanned aircraft
Model of external type RID equipments in English	rid_model_eng	Character string		Model name of external type RID equipments (in English) In the case of built-in type RID, the same as the model of unmanned aircraft
Serial number of external type RID equipments	rid_manufacturing_number	Character string		Serial number of external type RID equipments In the case of built-in type RID, the same as the serial number of the unmanned aircraft
Date and time of update	modified_date	Character string	○	Date and time of update(UTC) Return an empty string when initially registered. YYYY-MM-DDThh:mm:ssZ format.
Writing flag	write_status	Character string	○	Status of writing registration ID information to RID equipments 0 : Unwritten 1 : Written

Attachment 2

(6) Definition of API for obtaining cryptographic key information

Information such as the valid cryptographic key of the airframe identified by the registration ID is returned in the form of a data block when writing to RID equipments. In order to prevent the same external module from being written to by multiple unmanned aircraft, after the drone registration system receives an API request to acquire the cryptographic key information, check all owners and airframe using the serial number of the external type RID equipments as the key to see if any airframe has already had the written flag. If there is an airframe that has already been written with the same module, the data block cannot be acquired. (Return an empty string.)

① Request parameter

Table 6 shows the definitions of the request parameters of the API for obtaining cryptographic key information.

Table 6: Request parameters of API for obtaining cryptographic key information

Item name	Parameter name	Data type	Required	Contents
Registration ID	registration_code	Character string	○	Registration ID for the selected airframe
Cryptographic key remake flag	key_remake	Character string	○	Parameter specifying the re-creation of the cryptographic key 0 : No remake of cryptographic key 1 : With cryptographic key remake

The cryptographic key remake flag in the request parameter must be set to 0, since it is not necessary to remake the cryptographic key when writing to the remote ID equipments for the first time. For the second and subsequent writing, remake of the cryptographic key must be required for security reasons, and the request must be made with parameter 1. Judgement after the first and second time is made by the presence or absence of the update time and date which is a response item of the API for obtaining owner information and the API for obtaining a list of owner information.

② Response body

Table 7 shows the definition of the response body when a request for the API for obtaining cryptographic key information is successful.

Attachment 2

Table 7: Response body when successfully obtaining cryptographic key information through API

Item name	Parameter name	Data type	Required	Contents
Registration ID	registration_code	Character string	○	Registration ID issued by the government
Writing data block	data block	Character string	○	Information (binary) to be written to RID equipments, which is encoded by base64. The data definitions are shown in Table 8.
Writing flag	write_status	Character string	○	Status of writing registration ID information to RID equipments 0 ： Unwritten 1 ： Written
Date and time of update	modified_date	Character string	○	Date and time of update(UTC) Return an empty string when initially registered. YYYY-MM-DDThh:mm:ssZ format.
Transmission method	broadcast_methoed	Character string	○	Remote ID transmission method. One of the following. 0: Transmission method not set 1: Bluetooth 5.0 transmission method as described in the RID Specification 2: Wi-Fi Aware transmission method as described in the RID Specification 3: Wi-Fi Beacon transmission method as described in the RID Specification

By decoding the writing data block in Base64 format, binary data with the following structure is obtained.

As the structure of the data obtained here is identical to the data described in "Attachment 1 Remote ID Equipments Interface Specification Table 5: Command data definition for RID writing Command ", it can be written to the RID equipments as it is.

Definitions of the data are shown in Table 8. The item whose byte order is "little endian" is Write "LE" in the Endian column.

Table 8: Definitions of writing data blocks

Offset	Size (bytes)	Data	Endian	Description	Remarks
0	1	Key type	–	Type of key used for authentication code generation	0x00: Unregistered 0x01: Indicates that the key is for AES-CCM (128bit) All others are undefined (add values when adding authentication methods in the future). Registration system generates this value.
1	1	UA type	–	Airframe type Obtained from the registration system	Remote ID Specification: UA Type of Basic ID Messages
2	15	Registration ID	–	Add ″JA.″ to the beginning of the registration ID issued by the government. Obtained from the registration system	Remote ID Specification：UAS ID of Basic ID Message, ID type=2
17	20	Serial number	–	Serial number of RID equipments (If the serial number does not match the factory-set serial number, Manufacturer App must handle as writing error) Obtained from the registration system	Remote ID Specification：UAS ID of Basic ID Message, ID type=1 If the serial number is shorter than 20 digits, the back is filled with 0x00
37	4	Start	LE	Effective date of registration	Number of seconds elapsed since 1/1/2019 00:00:00 (Remote ID Specification: Authentication Message Page 0: Same generation method as Timestamp) Registration system generates this value.
41	4	Expire	LE	Expiration date of registration	
45	32	Key information	–	Key information	Key information corresponding to the key type In the case of AEC-CCM, the first 16 bytes are the key information, the following 6 bytes are Nonce information, and the last 10 bytes are values filled with 0x00 Registration system generates this value.
77	23	Reserved	–	–	All to be filled with 0x00.
100	72	Signature Information	–	Digital signature generated from data ranging from Offset 0-99	SHA-256 for the hash algorithm Binary data DER encoding digital signature generated by ECDSA using P-256 curve at the registration system If it is shorter than 72 bytes, it is suffixed with 0x00. Registration system generates this value.

Attachment 2

As shown in Table 8, the information obtained from the registration system and its signature information are stored in the writing data block. When a response is received, the authenticity of the data must be checked by comparing the result of decrypting the signature information using the public key notified in advance by the MLIT in 1. (2) with the hash value obtained using the SHA-256 algorithm from the portion of the command data excluding the signature information.

(7) Defining the API for storing remote ID registration results

After the completion of remote ID registration, the registration results must be stored in the registration system.

① Request body

The definition of the request body for storing the remote ID registration results through API is shown in Table 9.

Table 9: Request body for storing the remote ID registration results through API

Item name	Parameter name	Data type	Required	Contents
Registration ID	registration_code	Character string	○	Registration ID for the selected airframe
Writing flag	write_status	Character string	○	Status of writing registration ID information to RID equipments 0 : Unwritten 1 : Written

② Response body

Table 10 shows the definition of the response body when a request for storing the remote ID registration results through API is successful.

Table 10: Response body when a request for storing the remote ID registration results through API is successful

Item name	Parameter name	Data type	Required	Contents
Registration ID	registration_code	Character string	○	Registration ID issued by the government
Writing flag	write_status	Character string	○	Status of writing registration ID information to RID equipments 0 : Unwritten 1 : Written
Date and time of update	modified_date	Character string	○	Date and time of update(UTC) YYYY-MM-DDThh:mm:ssZ format.
Transmission method	broadcast_method	Character string	○	Remote ID transmission method. One of the following. 0: Transmission method not set 1: Bluetooth 5.0 transmission method as described in the RID Specification 2: Wi-Fi Aware transmission method as described in the RID Specification 3: Wi-Fi Beacon transmission method as described in the RID Specification

Attachment 2

6. OpenAPI

```
openapi: 3.0.0
info:
  title: RemoteID
  version: '1.0'
  description: |-
    API for writing remote ID to drone
  contact:
    name: Ministry of Land, Infrastructure, Transport and Tourism
  license:
    name: MLIT
tags:
  - name: Owner
paths:
  /rid/aircrafts:
    get:
      summary: Acquisition of list of owned airframe information
      tags:
        - Owner
      parameters: []
      responses:
        '200':
          description: OK
          content:
            application/json:
              schema:
                type: array
                items:
                  $ref: '#/components/schemas/Aircraft'
        '400':
          description: Bad Request
        '500':
          description: Internal Server Error
      operationId: get-rid-aircrafts
      description: |-
        Acquire a list of airframe information owned by the user

  /rid/aircrafts/{registration_code}:
    get:
      summary: Acquisition of list of owned airframe information
      tags:
        - Owner
      parameters:
        - name: registration_code
          description: Registration ID
          in: path
          required: true
          schema:
            $ref: '#/components/schemas/Registration_Code'
      responses:
        '200':
          description: OK
          content:
            application/json:
              schema:

                $ref: '#/components/schemas/Aircraft'
        '400':
          description: Bad Request
        '500':
          description: Internal Server Error
      operationId: get-rid-aircrafts-by-registration_code
      description: |-
        Acquire the owned airframe information using Registration ID as a key.
  /rid/remoteid:
    get:
      summary: Acquire cryptographic key information
      tags:
        - Owner
      parameters:
      - name: registration_code
        description: Registration ID
        schema:
          $ref: '#/components/schemas/Registration_Code'
        in: query
        required: true
      - name: key_remake
        description: Cryptographic key re-generate flag
        schema:
          $ref: '#/components/schemas/Key_Remake'
        in: query
        required: true
      responses:
        '200':
          description: OK
          content:
            application/json:
               schema:
                type: object
                properties:
                  registration_code:
                    $ref: '#/components/schemas/Registration_Code'
                  datablock:
                    format: byte
                    type: string
                    description: Base64-encoded information to be written to the RID
equipments
                  write_status:
                    $ref: '#/components/schemas/Write_Status'
                  modified_date:
                    format: date-time
                    type: string
                    description: Updated date and time
                  broadcast_methoed:
                    $ref: '#/components/schemas/Broadcast_Methoed'
        '400':
          description: Bad Request
        '500':
          description: Internal Server Error
      operationId: get-rid-remoteid
      description: Acquire the encryption key information required to generate the
authentication data of the authentication message.
    post:
      summary: Storing Remote ID registration result
      tags:
        - Owner
      parameters: []
      requestBody:
```

Attachment 2

```
        content:
          application/json:
            schema:
              type: object
              properties:
                  registration_code:
                    $ref: '#/components/schemas/Registration_Code’

                  write_status:
                    $ref: '#/components/schemas/Write_Status’

        description: Result of writing to RemoteID
      responses:
        '200’:
          description: OK
          content:
            application/json:
              schema:
                type: object
                properties:
                  registration_code:
                    $ref: '#/components/schemas/Registration_Code’
                  write_status:
                    $ref: '#/components/schemas/Write_Status’
                  modified_date:
                    format: date-time
                    type: string
                    description: Updated date and time
                  broadcast_methoed:
                    $ref: '#/components/schemas/Broadcast_Methoed’
        '400’:
          description: Bad Request
        '500’:
          description: Internal Server Error
      operationId: post-rid-remoteid
      description: Store the results of writing to RID equipments in the registration
system
components:
  schemas:
    Registration_Code:
      type: string
      minLength: 12
      maxLength: 12
      description: Registration ID
    Write_Status:
      enum:
      - ″0″
      - ″1″
      type: string
      description: |-
        Write status of RID
        ″0″ - unwritten
        ″1″ - Written
    Broadcast_Methoed:
      enum:
      - ″0″
      - ″1″
      - ″2″
      - ″3″
      type: string
```

Attachment 2

```
      description: |-
        Transmission method
        "0" - Transmission method not set
        "1" - Bluetooth 5.0 transmission method as described in the RID Specification
        "2" - Wi-Fi Aware transmission method as described in the RID Specification
        "3" - Wi-Fi Beacon transmission method as described in the RID Specification
    Manufacturing_Category:
      enum:
      - "1"
      - "2"
      type: string
      description: |-
        manufacturing_category
        "1" - Manufactured UA /Altered UA
        "2" - Amateur-Built UA / Others
    Manufacturing_Number:
      type: string
      maxLength: 20
      description: manufacturing_number
    Remodeling_Type:
      enum:
      - "1"
      - "2"
      type: string
      description: |-
        Alteration
        "1" - Altered
        "2" - Not altered /Amateur-Built
    Aircraft_Type:
      enum:
      - "1"
      - "2"
      - "3"
      - "4"
      - "5"
      - "6"
      type: string
      description: |-
        UA category
        "1" - Airplane
        "2" - Rotorcraft(Helicopter)
        "3" - Rotorcraft(Multirotor)
        "4" - Rotorcraft(Others)
        "5" - Glider
        "6" - Airship
    Rid_Type:
      enum:
      - "0"
      - "1"
      - "2"
      type: string
      description: |-
        RIDType
        "0" - No
        "1" - Yes(built-in)
        "2" - Yes(external)
    Key_Remake:
      enum:
      - "0"
      - "1"
```

Attachment 2

```
      type: string
      description: |-
        Parameters to Recreate Encryption Keys        ″0″ - No encryption rekey
        ″1″ - Encryption rekey
    Aircraft:
      type: object
      description: |-
        Airframe information
        Extract airframe-identifying information from the registration system
      properties:
        registration_code:
          $ref: '#/components/schemas/Registration_Code'
        manufacturing_category:
          $ref: '#/components/schemas/Manufacturing_Category'
        manufacturer_jpn:
          type: string
          description: UA manufacturer name(Japanese)
        model_jpn:
          type: string
          description: UA model name(Japanese)
        manufacturer_eng:
          type: string
          description: UA manufacturer name(English)
        model_eng:
          type: string
          description: UA model name(English)
        manufacturing_number:
          $ref: '#/components/schemas/Manufacturing_Number'
        remodeling_type:
          $ref: '#/components/schemas/Remodeling_Type'
        aircraft_type:
          $ref: '#/components/schemas/Aircraft_Type'
        rid_type:
          $ref: '#/components/schemas/Rid_Type'
        rid_manufacturer_jpn:
          type: string
          description: Manufacturer name of RID external equipment(Japanese)
        rid_model_jpn:
          type: string
          description: RID external equipment moden name(Japanese)
rid_manufacturer_eng:
          type: string
          description: Manufacturer name of RID external equipment(English)
        rid_model_eng:
          type: string
          description: RID external equipment moden name(English)
        rid_manufacturing_number:
          $ref: '#/components/schemas/Manufacturing_Number'
        modified_date:
          format: date-time
          type: string
          description: Updated date and time
        write_status:
          $ref: '#/components/schemas/Write_Status'
      required:
        - registration_code
        - manufacturing_category
- manufacturer_jpn
- model_jpn
- manufacturer_eng
```

Attachment 2

```
- model_eng
- manufacturing_number
- aircraft_type
- rid_type
- write_status
```

7. List of requests and responses regarding authentication with the registration system

Detailed information regarding authentication with the registration system is described below.

(1) Request processing for authentication

Request processing for authentication (user authorization, access token acquisition, and user attribute acquisition processing) must be performed in accordance with the provisions of Open ID Connect.

Table 11 shows the request for authentication.

Table 11: Request for authentication

Request name	Kinds	Request pass	Contents
User authorization	GET	/auth/realms/drs/protocol/openid-connect/auth	Judge the user's authentication status / authorization status, Registration system redirects to the appropriate page, and return the authorization code.
Access token acquisition	POST	/auth/realms/drs/protocol/openid-connect/token	Get an access token and a refresh token (for updating the access token).
attribute acquisition	GET	/auth/realms/drs/protocol/openid-connect/userinfo	Get user attribute information.

The header and parameters of each request are described in the individual request definition.

(2) Definition of user authorization request

Judge the user's authentication status / authorization status, Registration system redirects to the appropriate page, and return the authorization code.

① Request parameter

Table 12 shows the request parameters of the authorization request.

Table 12: Request parameters of the authorization request

Item name	Parameter name	Data type	Required	Contents
Response type	response_type	Character string	○	Fixed to ″code″
Client ID	client_id	Character string	○	Predefined for each Manufacturer App [Client ID]
Redirect URI	redirect_uri	Character string	○	Predefined for each Manufacturer App [URL to redirect when login is successful]
Scope	scope	Character string	○	Fixed to "openid offline_access"
Status	state	Character string	○	Parameters used to maintain the state between the request and the callback for it
Code challenge	code_challenge	Character string	○	A character string encoded in Base64URL format by hashing (encrypting) the parameter ″code_verifier″ specified at the time of access token acquisition request with SHA256.
Code challenge method	code_challenge_method	Character string	○	Fixed to "S256"
Display language	ui_locales	Character string	−	One of the following values ja en If not specified, Japanese / English display will be switched based on the Accept-Language request header.

Attachment 2

② Response(Normal)

When the request is successful, the screen for entering the login ID and password will be displayed.

When the user performs a login operation, it redirects to "[URL to redirect when login is successful] defined in advance for each maker application". Table 13 shows the redirect query parameter on successful login.

Table 13: Redirect query parameter on successful login

Item name	Parameter name	Data type	Required	Contents
Authorization code	code	Character string	○	Authorization code
Session state	session_state	Character string	○	Session state
Status	state	Character string	○	Check if the value saved at the time of request matches the value at the time of callback. If they do not match, it may be CSRF, so do not execute the access token acquisition request.

③ Response(Errors)

When any errors occur, the screen transitions to the system error screen.

For some errors, Registration system redirects to "Predefined for each maker application [URL to redirect when login is successful]". Table 14 shows the redirect query parameters on login error.

Table 14: Redirect query parameter on login error

Item name	Parameter name	Data type	Required	Contents
Error code	error	Character string	○	Error code
Error content	error_description	Character string	○	Detailed explanation of the error content
Status	state	Character string	○	Check if the value saved at the time of request matches the value at the time of callback. If they do not match, it may be CSRF.

If the registration system is during system maintenance, the response body shown in Table 15 and the error response shown in Table 16 are returned in JSON format.

Table 15: Response body during system maintenance

Item name	Parameter name	Data type	Required	Contents
Error code	errorCode	Character string	○	Error code
Error message	errorMessage	Character string	○	Detailed explanation of the error content

Table 16: Error response during system maintenance

HTTP status	Error code	Error message	Explanation
503	E5030001	none	When the API is called during maintenance of the registration system

Attachment 2

(3) Definition of the access token acquisition request

Registration system returns the access token and refresh token (for updating the access token)

① Request header

Table 17 shows the request header of access token acquisition request.

Table 17: Request header of access token acquisition request

Item name	Header name	Data type	Required	Contents
Content type	Content-Type	Character string	○	Fixed to ″application/x-www-form-urlencoded;charset=UTF-8”

② Request parameter

Table 18 shows the request parameter of access token acquisition request.

Table 18: Request parameter of access token acquisition request

Item name	Parameter name	Data type	Required	Contents
Grant type	grant_type	Character string	○	Fixed to ″authorization_code″
Authorization code	code	Character string	○	Authorization code returned in the user authorization request
Redirect URI	redirect_uri	Character string	○	Predefined for each maker application [URL to redirect when login is successful]
Client ID	client_id	Character string	○	Predefined for each maker app [Client ID]
Code verifier	code_verifier	Character string	○	A random character string consisting of 43 to 128 characters ″A-Z″, ″a-z″, ″0-9″, ″-″, ″.″, ″_″, ″~″.

③ Response body(Normal)

Table 19 shows the response body when the access token acquisition request is normal.

Table 19: Rresponse body when the access token acquisition request is normal

Item name	Parameter name	Data type	Required	Contents
Access token	access_token	Character string	○	Token required when issuing a userinfo request or API
Expir time	expires_in	Numeric	○	Expire time of access_token (seconds)
Refresh token expire time	refresh_expires_in	Numeric	○	Expire time of refresh_token (seconds)
Refresh token	refresh_token	Character string	○	Token required when updating access_token
Token type	token_type	Character string	○	Fixed to “bearer”
ID token	id_token	Character string	○	ID token(JWT(JSON Web Token))
Not before policy	net-before-policy	Numeric	○	Value for verifying the validity of the access token
Session state	session_state	Character string	○	Session state
Scope	scope	Character string	○	Fixed to “openid profile offline_access rid”

④ Response body(Errors)

Table 20 shows the response code for errors in the access token acquisition request.

Table 20: Response body for errors in the access token acquisition request

Item name	Parameter name	Data type	Required	Contents
Error code	error	Character string	○	Error code
Error content	error_description	Character string	○	Detailed explanation of the error content

Table 21 shows the typical error code for access token acquisition request.

Table 21: Typical error code for access token acquisition request

HTTP status	Error code	Explanation
400	unauthorized_client	Invalid parameter "client_id"
400	invalid_request	Invalid parameter "grant_type"
400	invalid_grant	Authorization code is invalid, expired, invalid, parameter "redirect_uri" is invalid

(4) Definition of the Attribute acquisition request

Registration system returns user attribute information.

① Request header

Table 22 shows the request header of attribute acquisition request.

Table 22: Request header of attribute acquisition request

Item name	Header name	Data type	Required	Contents
Authorization	Authorization	Character string	○	Bearer [access_token obtained by access token acquisition request]

② Request parameter

None.

③ Response body(Normal)

Table 23 shows the response body when the attribute acquisition request is normal.

Table 23: Response body when the attribute acquisition request is normal

Item name	Parameter name	Data type	Required	Contents
Account management number	sub	Character string	○	Account management number (return internal ID)
User name	preferred_username	Character string	○	User name

④ Response body(Errors)

Table 24 shows the response body for errors in the attribute acquisition request.

Table 24: Response body for errors in the attribute acquisition request

Item name	Parameter name	Data type	Required	Contents
Error code	error	Character string	○	Error code
Error content	error_description	Character string	○	Detailed explanation of the error content

Table 25 shows the typical error code for attribute acquisition request.

Table 25: Typical error code for attribute acquisition request

HTTP status	Error code	Explanation
400	invalid_request	No access token
401	invalid_token	Unauthorized access token, expired, invalid
403	insufficient scop	Insufficient access rights

8. Response list in the event of an API error

The details of the error response to be returned when an error occurs in the API provided for writing registration ID information etc. are described below. Those error messages are wrritten in both English and Japanese.

(1) Error response common to each API

The details of the errors that commonly occur in each API and their responses are shown below.

① At the time of access token verification error

Table 26 shows the response body at the time of access token verification error.

Table 26: Response body at the time of access token verification error

Item name	Parameter name	Data type	Required	Contents
Error message	message	Character string	○	Error contents

Table 27 shows the error response body at the time of access token verification error.

Table 27: Error response at the time of access token verification error

HTTP status	Error code	Explanation
401	Unauthorized	No access token
403	Forbidden	Access token verification NG
500	Internal Server Error	Access token verification failure (abnormal termination)

② At the time of processing logic error

If any errors occur in the processing logic, the response code shown in Table 28 and the response body shown in Table 29 are returned. Table 30 shows the error response common to each API. The error code that occurs for each API is described in the individual API definition.

Table 28: Response code at the time of processing logic error

HTTP status	Error code	Explanation
400	Request parameter error	Processing failure (parameter invalid)
500	In-API system error	Unexpected system error

Table 29: Response body at the time of processing logic error

Item name	Parameter name	Data type	Required	Contents
Error code	errorCode	Character string	○	Error code
Error message	errorMessage	Character string	○	Detailed explanation of the error content

Table 30: API common error code at the time of processing logic error

HTTP status	Error code	Error message	Explanation
400	E4000001	A system error has occurred. システムエラーが発生しました。	Parameter error (json parsing error)
500	E5000002	A system error has occurred. システムエラーが発生しました。	System error inside API processing (DB connection error, etc.)

③ During maintenance of the registration system

During the maintenance of the registration system, the response body shown in Table 31 and the error response shown in Table 32 are returned.

Table 31: Response body during system maintenance

Item name	Parameter name	Data type	Required	Contents
Error code	errorCode	Character string	○	Error code
Error message	errorMessage	Character string	○	Detailed explanation of the error content

Table 32: API common error code at the time of processing logic error

HTTP status	Error code	Error message	Explanation
503	E5030001	None	When the API is called during registration system maintenance.

(2) API individual error response

The details of the errors that occur individually in each API and their responses are shown below.

① API for obtaining information on the list of owned airframe

Table 33 shows the API individual error code that occur in the processing logic of API for obtaining information on the list of owned airframe.

Table 33: Individual error code of the API for obtaining information on the list of owned airframe

HTTP status	Error code	Error message	Explanation
400	E4000101	A system error has occurred. システムエラーが発生しました。	Failed to get user ID from access token

② API for obtaining information on the owned airframe

Table 34 shows the API individual error code that occur in the processing logic of API for obtaining information on the owned airframe.

Table 34: Individual error code of the API for obtaining information on the owned airframe

HTTP status	Error code	Error message	Explanation
400	E4000201	A system error has occurred. システムエラーが発生しました。	Failed to get user ID from access token
400	E4000202	A system error has occurred. システムエラーが発生しました。	Parameter check error (registration symbol (required, 12 digits))

③ API for acquisition of cryptographic key information

Table 35 shows the API individual error code that occur in the processing logic of API for acquisition of cryptographic key information.

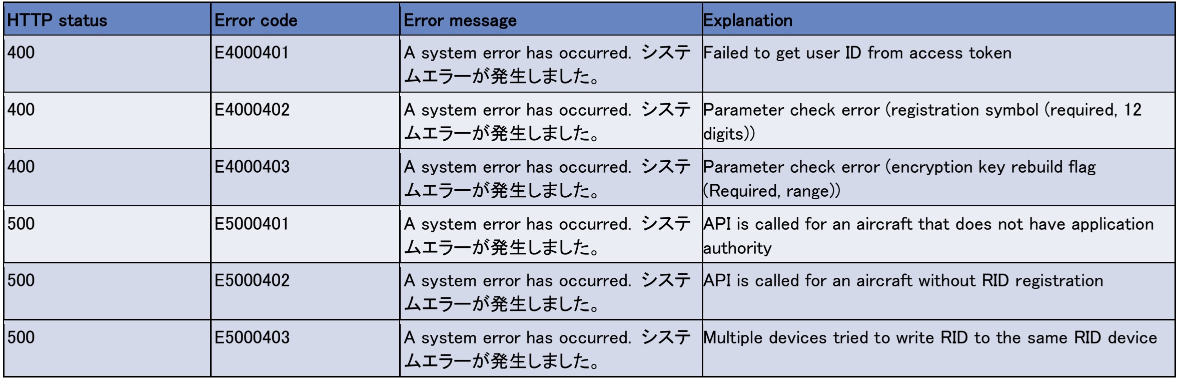
Table 35: Individual error code of the API for acquisition of cryptographic key information

HTTP status	Error code	Error message	Explanation
400	E4000301	A system error has occurred. システムエラーが発生しました。	Failed to get user ID from access token
400	E4000302	A system error has occurred. システムエラーが発生しました。	Parameter check error (registration symbol (required, 12 digits))
400	E4000303	A system error has occurred. システムエラーが発生しました。	Parameter check error (encryption key rebuild flag (Required, range))
400	E5000301	A system error has occurred. システムエラーが発生しました。	API is called for an aircraft that does not have application authority
400	E5000302	A system error has occurred. システムエラーが発生しました。	API is called for an aircraft without RID registration

④ API for storing the remote ID registration results

Table 36 shows the API individual error code that occur in the processing logic of API for storing the remote ID registration results.

Table 36: Individual error code for storing the remote ID registration results through API



HTTP status	Error code	Error message	Explanation
400	E4000401	A system error has occurred. システムエラーが発生しました。	Failed to get user ID from access token
400	E4000402	A system error has occurred. システムエラーが発生しました。	Parameter check error (registration symbol (required, 12 digits))
400	E4000403	A system error has occurred. システムエラーが発生しました。	Parameter check error (encryption key rebuild flag (Required, range))
500	E5000401	A system error has occurred. システムエラーが発生しました。	API is called for an aircraft that does not have application authority
500	E5000402	A system error has occurred. システムエラーが発生しました。	API is called for an aircraft without RID registration
500	E5000403	A system error has occurred. システムエラーが発生しました。	Multiple devices tried to write RID to the same RID device

Attachment 2

9. Notes on validating authentication requests

(1) State verification

Table 37 shows the method of state verification

Table 37: Method of state verification

No	Method of verification
1	Make sure that the value of state acquired in the user authorization response is the same as the value sent in the request.

(2) ID token verification

“id_token” is in “JSON Web Token (JWT)” format, and is divided into a header part, a payload part and a signature part by separating them with ".". The nonce is included in the payload part.

The header part and payload part are encoded in Base64, and the values shown in Table 38 are set.

※ Describes the main ones used in “id_token” validation. So that other values are also included.

Table 38: Key parameters used for ID token verification

Classification	Parameter name	Method of verification
Header part	alg	Hash algorithm used for sign id_token
Payload part	iss	Issuer of id_token. “https:// [FQDN of registration system] /auth/realms/drs”
	aud	The recipient of id_token. The client_id of the RP is set.
	exp	The expiration time of id_token. UNIX time (seconds elapsed since UTC 1970/1/1 00:00:00).
	iat	The expiration time of id_token. UNIX time (seconds elapsed since UTC 1970/1/1 00:00:00).
	auth_time	The time when the user was authenticated. UNIX time (seconds elapsed since UTC 1970/1/1 00:00:00).

Verification of ID token must be carried out as shown in Table 39.

Table 39: Method of ID token verification

No	Method of verification
1	Make sure that the value of iss (issuer of id_token) matches ″https: // [FQDN of registration system] / auth / realms / drs″.
2	Make sure that the value of aud (recipient of id_token) matches the client_id sent in the authentication request.
3	Make sure exp (id_token expiration time) is after the current time.
4	Make sure that iat (id_token issuance time) is before the current time and is not too old. * It is up to the RR side to decide how old id_token is allowed.
5	Make sure that auth_time (the time the user was authenticated) is before the current time and is not too old. * It is up to the RP side to decide how old the user's authentication time is allowed.

Attachment 3

DD MM YY

Notification form of Self-verification Result and Type Information

To Director, Unmanned Aircraft Systems Division, Civil Aviation Bureau,
Ministry of Land, Infrastructure, Transport and Tourism

As a result of self-verification of the RID equipments, it has been confirmed that it conforms to the "Requiremtnes for remote ID devices and applications", and we hereby submit the following notification with the accompanying document.

Notifier (Corporate Name) :
Name and title of representative :
Name and title of person in charge :
Address:
Phone number :
Email address :

Manufacturer name	
Model name	
Built-in type or External type	□ Built-in type (S/N is as same as an unmanned aircraft' one) □ External type (S/N is different to an unmanned aircraft's one)
Communication method	□ Bluetooth 5.x Long Range □ Wi-Fi Aware (Neighbor Awareness Networking) □ Wi-Fi Beacon
Size（L×W×H）	
Weight	
Exterior photo	

1. In the case of developping and manufacturring a manufacture application, enter its name in the Model name column. Size, weight and exterior photo may be not required.
2. For a built-in type, enter the type, size and weight of RID equipments of unmanned aircraft.
3. Submit with documents verifying compliance with the Remote ID Specification. Those must include a number of compling with the Japan Radio Law and a test result of its reachable distance.

Attachment 4

DD MM YY

Notification application form for the Remote ID public key and application authentication code

To: Director, Unmanned Aircraft Systems Division,Civil Aviation Bureau,
Ministry of Land, Infrastructure, Transport and Tourism

To comply with the "Requirements for remote ID devices and applications", we apply for the notification of public key for digital signature and application authentication code for the development and manufacture of Remote ID equipments and manufacturer applications as follows.

Notifier (Corporate Name) :
Name and title of representative :
Name and title of person in charge :
Address:
Phone number :
E-mail address :

Items to be developed and manufactured	□ Remote ID equipments □ Built-in type (S/N is as same as an unmanned aircraft's one) □ External type (S/N is different to an unmanned aircraft's one □ Manufacturer application
Communication method of Remote ID equipments	□ Bluetooth 5.x Long Range □ Wi-Fi Aware (Neighbor Awareness Networking) □ Wi-Fi Beacon
Scheduled start of development	Year & Month:
Scheduled completion date of development	Year & Month
（For manufacturer applications, include the following information about OpenID Connect※）	
Redirect URL after login	
Accessing IP address	

※The manufacturer of the manufacturer application will be notified of the client ID as well, which will be required for OpenID Connect authentication when connecting to the registration system.