DJI R SDK

Protocol and User Interface

2021.01

V2.3

Edited by

Approved by

Edited on

Document No.

SZ DJI TECHNOLOGY CO., LTD.

Release Notes

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Version | Date | Section | Reason for Change | Description of Change |
| 1.0.0.0 | July 17, 2019 |  |  | Draft document |
| 2.0.0.0 | October 8, 2019 | 3 | 1. Deleted sample code 2. Added cyclic redundancy check (CRC) parameters description | 1. First release 2. Added CRC model parameters description |
| 2.1.0.1 | May 11, 2020 | 2.3  3.3  3.4 | Added commands and CRC pattern sample | 1. Added module version protocol 2. Added sample of command group pack 3. Added CRC sample code |
| 2.1.0.2 | June 17, 2020 | 2.3, 3.1 | Added external device control command and hardware support description | 1. Added joystick command 2. Added CAN support |
| 2.2.0.3 | June 22, 2020 | 2.3 | Added commands | 1. Added function to obtain handheld gimbal user parameters 2. Added function to set handheld gimbal user parameters 3. Added function to set gimbal operating mode 4. Added Recenter and Selfie 5. Added third-party camera motion command |
| 2.2.0.4 | July 16, 2020 | 2.3 | Added commands | 1. Added Follow Mode settings 2. Added Auto Tune settings and information push function 3. Added ActiveTrack settings 4. Added function to obtain camera status |
| 2.2.0.5 | October 30, 2020 | 2.2 | Modified reply frame data segment | Added CmdSet and CmdID to the reply frame return packet, making it consistent with the command frame |
| 2.2.0.6 | January 7, 2021 | 2.3 | Added commands | 1. Modified the command to obtain module version number 2. Added the command to   control the Focus Motor |

**Contents**

1. [DJI R SDK Protocol Introduction 2](#_bookmark0)
2. [DJI R SDK Protocol Description 2](#_bookmark0)
   1. [Data Format 2](#_bookmark0)
   2. [Field Description 2](#_bookmark0)
   3. [Detailed Descriptions 3](#_bookmark1)
      1. [Commands Set and Command ID 3](#_bookmark1)
      2. [Return Code 4](#_bookmark2)
      3. [Device ID 4](#_bookmark2)
      4. [Gimbal Command Set Data Segment Details 5](#_bookmark3)
         1. [Handheld Gimbal Position Control 5](#_bookmark3)
         2. [Handheld Gimbal Speed Control 6](#_bookmark4)
         3. [Handheld Gimbal Information Obtaining 6](#_bookmark4)
         4. [Handheld Gimbal Limit Angle Settings 7](#_bookmark5)
         5. [Obtain Handheld Gimbal Limit Angle 7](#_bookmark5)
         6. [Handheld Gimbal Motor Stiffness Settings 8](#_bookmark6)
         7. [Obtain Handheld Gimbal Motor Stiffness 8](#_bookmark6)
         8. [Handheld Gimbal Parameter Push Settings 8](#_bookmark6)
         9. [Handheld Gimbal Parameter Push 9](#_bookmark7)
         10. [Obtain Module Version Number 10](#_bookmark8)
         11. [External Device Control Command Push 10](#_bookmark8)
         12. [Obtain Handheld Gimbal User Parameters 11](#_bookmark9)
         13. [Handheld Gimbal Parameter Information Push Settings 12](#_bookmark10)
         14. [Handheld Gimbal Operating Mode Settings 13](#_bookmark11)
         15. [Handheld Gimbal Recenter, Selfie, and Follow Modes Settings 14](#_bookmark12)
         16. [Gimbal Auto Calibration Settings 14](#_bookmark12)
         17. [Gimbal Auto Calibration Status Push 15](#_bookmark13)
         18. [Gimbal ActiveTrack Settings 15](#_bookmark13)
         19. [Focus Motor Control Command 15](#_bookmark13)
      5. [Camera Command Set Data Segment Details 16](#_bookmark14)
         1. [Third-Party Camera Motion Command 16](#_bookmark14)
         2. [Third-Party Camera Status Obtain Command 17](#_bookmark15)
3. [Notices 18](#_bookmark16)
   1. [Hardware Support 18](#_bookmark16)
      1. [Device Connection Diagram 18](#_bookmark16)
      2. [Ronin Series Accessories (RSA)/NATO Ports 19](#_bookmark17)
   2. [Software Support 19](#_bookmark17)
   3. [Command Sample 19](#_bookmark17)
   4. [CRC Code Sample 20](#_bookmark18)

# DJI R SDK Protocol Introduction

The DJI R SDK protocol is a simple, easy, stable, and reliable communication protocol. A third party can control the handheld gimbal device movement and obtain its partial information via the DJI R SDK protocol. With the support of the DJI R SDK protocol, the handheld gimbal device has greater extensibility and can be applied in more scenarios.

# DJI R SDK Protocol Description

## 2.1 Data Format

The data packet format of the DJI R SDK protocol is shown below:

|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
| SOF | Ver/Length | CmdType | ENC | RES | SEQ | CRC-16 | DATA | CRC-32 |
| 1-byte | 2-byte | 1-byte | 1-byte | 3-byte | 2-byte | 2-byte | n-byte | 4-byte |

- Figure 1 Data Packet Format -

## 2.2 Field Description

|  |  |  |  |
| --- | --- | --- | --- |
| Domain | Offset | Size | Descriptions |
| SOF | 0 | 1 | The frame header is set as 0xAA |
| Ver/Length | 1 | 2 | [15:10] - Version number (0 by default) [9:0] - The length of the entire frame  Note: LSB first |
| CmdType | 3 | 1 | [4:0] - Reply type  0 - No reply is required after data is sent 1 - Can reply or not after data is sent  2-31 - Reply is required after data is sent  [5] - Frame type  0 - Command frame 1 - Reply frame  [7:6] - Reserve (0 by default) |
| ENC | 4 | 1 | [4:0] - The length of supplementary bytes when encrypting (16-byte alignment is required when encrypting)  [7:5] - Encryption type 0 - Unencrypted  1 - AES256 encryption |
| RES | 5 | 3 | Reserved byte segment |
| SEQ | 8 | 2 | Serial number |
| CRC-16 | 10 | 2 | Frame header check |
| DATA | 12 | n | Data segment (description is shown below) |
| CRC-32 | n+12 | 4 | Frame check (the entire frame) |

- Figure 2 Data Packet Field Description -

Below shows the data segment content:

|  |  |  |  |
| --- | --- | --- | --- |
| Domain | Offset | Size | Descriptions |
| CmdSet | 0 | 1 | Command set |
| CmdID | 1 | 1 | Command code |
| CmdData | 2 | n-2 | Data content |

- Figure 3 Data Segment Content -

## 2.3 Detailed Descriptions

### Commands Set and Command ID

The command sets and command codes used by the handheld gimbal are shown below:

|  |  |  |
| --- | --- | --- |
| CmdSet | CmdID | Descriptions |
| 0x0E | 0x00 | Control handheld gimbal position  2.3.4.1 Handheld Gimbal Position Control |
| 0x01 | Control handheld gimbal speed  2.3.4.2 Handheld Gimbal Speed Control |
| 0x02 | Obtain the angle information of handheld gimbal, including joint angle and attitude angle  2.3.4.3 Obtain Handheld Gimbal Information |
| 0x03 | Set handheld gimbal limit angle  2.3.4.4 Handheld Gimbal Limit Angle Settings |
| 0x04 | Obtain handheld gimbal limit angle  2.3.4.5 Obtain Handheld Gimbal Limit Angle |
| 0x05 | Set handheld gimbal motor stiffness  2.3.4.6 Handheld Gimbal Motor Stiffness Settings |
| 0x06 | Obtain handheld gimbal motor stiffness  2.3.4.7 Obtain Handheld Gimbal Motor Stiffness |
| 0x07 | Set information push of handheld gimbal parameters  2.3.4.8 Handheld Gimbal Parameter Information Push Settings |
| 0x08 | Push handheld gimbal parameters  2.3.4.9 Push Handheld Gimbal Parameter |
| 0x0E | 0x09 | Obtain module version number  2.3.4.10 Obtain Module Version Number |
| 0x0A | Push joystick control command  2.3.4.11 External Device Control Command Push |
| 0x0B | Obtain handheld gimbal user parameters  2.3.4.12 Obtain Handheld Gimbal User Parameter |
| 0x0C | Set handheld gimbal user parameters  2.3.4.13 Handheld Gimbal User Parameters Settings |
| 0x0D | Set handheld gimbal operating mode  2.3.4.14 Handheld Gimbal Operating Mode Settings |
| 0x0E | Set handheld gimbal Recenter, Selfie, and Follow modes  2.3.4.15 Handheld Gimbal Recenter, Selfie, and Follow Modes Settings |

|  |  |  |
| --- | --- | --- |
| 0x0E | 0x0F | Set gimbal auto calibration  2.3.4.16 Set gimbal auto calibration |
| 0x10 | Set gimbal auto calibration status push  2.3.4.17 Set gimbal auto calibration status push |
| 0x11 | Set gimbal ActiveTrack  2.3.4.18 Set gimbal ActiveTrack |
| 0x12 | Focus Motor Control Command  2.3.4.19 Focus Motor Control Command |
| 0x0D | 0x00 | Third-party camera motion command  2.3.5.1 Third-Party Camera Motion Command |
| 0x01 | Third-party camera status obtain command  2.3.5.2 Third-Party Camera Status Obtain Command |

- Figure 4 Command Set and Command -

### Return Code

Return codes currently supported by the handheld gimbal are shown below:

|  |  |
| --- | --- |
| Error Code Value | Implication |
| 0x00 | Command execution succeeds |
| 0x01 | Command parse error |
| 0x02 | Command execution fails |
| 0xFF | Undefined error |

- Figure 5 Return Code Implication -

### Device ID

The device ID is a 4-byte figure used to differentiate devices that connect to the DJI R SDK system. The ID must be submitted to DJI for approval and can only be used once approved. The device IDs currently in use are listed below:

|  |  |
| --- | --- |
| Device ID | Descriptions |
| 0x00000000 | Reserved |
| 0x00000001 | DJI R SDK |
| 0x00000002 | Remote controller |

- Figure 6 Device ID -

### Gimbal Command Set Data Segment Details

#### Handheld Gimbal Position Control

CmdSet = 0x0E CmdID = 0x00 (the data segment details are shown below):

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Frame Type | Data | | | | |
| Offset | Size | Name | Type | Descriptions |
| Command frame | 0 | 2 | yaw\_angle | int16\_t | yaw angle, unit: 0.1° (range: -1800 to  +1800) |
| 2 | 2 | roll\_angle | int16\_t | roll angle, unit: 0.1° (range: -1800 to  +1800) |
| 4 | 2 | pitch\_angle | int16\_t | pitch angle, unit: 0.1° (range: -1800 to  +1800) |
| 6 | 1 | ctrl\_byte | uint8\_t | [7:4] - Reserved (must be 0)  [3] - Whether the pitch axis is valid/invalid 0: Valid  1: Invalid  [2] - Whether the roll axis is valid/invalid 0: Valid  1: Invalid  [1] - Whether the yaw axis is valid/invalid 0: Valid  1: Invalid  [0] - Control mode  0: Incremental control  1: Absolute control |
| 7 | 1 | time\_for\_action | uint8\_t | Command execution speed, unit: 0.1s This field is used to set the motion speed when the gimbal is executing this command. For example, when this field  is 20, the gimbal will rotate to the position  desired within 2s at a constant speed. |
| Reply frame | 0 | 1 | return code | uint8\_t | Refer to error return code 2.3.2 Return  Error |

* + - * + Figure 7 Position Control Command -

#### Handheld Gimbal Speed Control

CmdSet = 0x0E CmdID = 0x01 (the data segment details are shown below):

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Frame Type | Data | | | | |
| Offset | Size | Name | Type | Descriptions |
| Command frame | 0 | 2 | yaw\_speed | int16\_t | Unit: 0.1°/s (range: 0°/s to 360°/s) |
| 2 | 2 | roll\_speed | int16\_t | Unit: 0.1°/s (range: 0°/s to 360°/s) |
| 4 | 2 | pitch\_speed | int16\_t | Unit: 0.1°/s (range: 0°/s to 360°/s) |
| 6 | 1 | ctrl\_byte | uint8\_t | [7] - Control Bit  0: Release speed control 1: Take over speed control  [6:4] - Reserved, (must be 0)  [3] - Camera focal length  0: The moving speed will take the impact of camera focal length into consideration  1: The moving speed will not take the impact of camera focal length into consideration  [2:0] - Reserved (must be 0) |
| Reply frame | 0 | 1 | return code | uint8\_t | Refer to error return code 2.3.2 Return Code |

* + - * + Figure 8 Speed Control Command -

Note: This command can only control for 0.5s each time it is issued due to safety reasons. If users require continuous speed, they can send this command periodically. If users want to stop the rotation of three axes immediately, they can set the fields of yaw\_speed, pitch\_speed, and roll\_speed as 0.

#### Handheld Gimbal Information Obtaining

CmdSet = 0x0E CmdID = 0x02 (the data segment details are shown below):

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Frame Type | Data | | | | |
| Offset | Size | Name | Type | Descriptions |
| Command frame | 0 | 1 | ctrl\_byte | uint8\_t | 0x00: No operation  0x01: Obtain the attitude angle of handheld gimbal  0x02: Obtain the joint angle of  handheld gimbal |
| Reply frame | 0 | 1 | return code | uint8\_t | Refer to error return code 2.3.2 Return  Code |
| 1 | 1 | data\_type | uint8\_t | 0x00: Data is not ready  0x01: The current angle is attitude angle  0x02: The current angle is joint angle |

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Reply frame | 2 | 2 | yaw | int16\_t | yaw axis angle (unit: 0.1°) |
| 4 | 2 | roll | int16\_t | roll axis angle (unit: 0.1°) |
| 6 | 2 | pitch | int16\_t | pitch axis angle (unit: 0.1°) |

- Figure 9 Obtain Gimbal Information Command -

#### Handheld Gimbal Limit Angle Settings

CmdSet = 0x0E CmdID = 0x03 (the data segment details are shown below):

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Frame Type | Data | | | | |
| Offset | Size | Name | Type | Descriptions |
| Command frame | 0 | 1 | ctrl\_byte | uint8\_t | 0x00: No operation  0x01: Set handheld gimbal limit angle |
| 1 | 1 | pitch\_max | uint8\_t | Max. tilt axis angle (range: 0 to 179) |
| 2 | 1 | pitch\_min | uint8\_t | Min. tilt axis angle (range: 0 to 179) |
| 3 | 1 | yaw\_max | uint8\_t | Max. pan axis angle (range: 0 to 179) |
| 4 | 1 | yaw\_min | uint8\_t | Min. pan axis angle (range: 0 to 179) |
| 5 | 1 | roll\_max | uint8\_t | Max. roll axis angle (range: 0 to 179) |
| 6 | 1 | roll\_min | uint8\_t | Min. roll axis angle (range: 0 to 179) |
| Reply frame | 0 | 1 | return code | uint8\_t | Refer to error return code 2.3.2 Return  Code |

* + - * + Figure 10 Set Gimbal Limit Angle Command -

#### Obtain Handheld Gimbal Limit Angle

CmdSet = 0x0E CmdID = 0x04 (the data segment details are shown below):

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Frame Type | Data | | | | |
| Offset | Size | Name | Type | Descriptions |
| Command frame | 0 | 1 | ctrl\_byte | uint8\_t | 0x00: No operation  0x01: Obtain handheld gimbal limit angle |
| Reply frame | 0 | 1 | return code | uint8\_t | Refer to error return code 2.3.2 Return  Code |
| 1 | 1 | pitch\_max | uint8\_t | Max. tilt axis angle (range: 0 to 179) |
| 2 | 1 | pitch\_min | uint8\_t | Min. tilt axis angle (range: 0 to 179) |
| 3 | 1 | yaw\_max | uint8\_t | Max. pan axis angle (range: 0 to 179) |
| 4 | 1 | yaw\_min | uint8\_t | Min. pan axis angle (range: 0 to 179) |
| 5 | 1 | roll\_max | uint8\_t | Max. roll axis angle (range: 0 to 179) |
| 6 | 1 | roll\_min | uint8\_t | Min. roll axis angle (range: 0 to 179) |

* + - * + Figure 11 Obtain Gimbal Limit Angle Command -

#### Handheld Gimbal Motor Stiffness Settings

CmdSet = 0x0E CmdID = 0x05 (the data segment details are shown below):

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Frame Type | Data | | | | |
| Offset | Size | Name | Type | Descriptions |
| Command frame | 0 | 1 | ctrl\_byte | uint8\_t | 0x00: No operation  0x01: Set handheld gimbal motor stiffness |
| 1 | 1 | pitch\_stiffness | uint8\_t | VALUE : 0 ~ 100 |
| 2 | 1 | roll\_stiffness | uint8\_t | VALUE : 0 ~ 100 |
| 3 | 1 | yaw\_stiffness | uint8\_t | VALUE : 0 ~ 100 |
| Reply frame | 0 | 1 | return code | uint8\_t | Refer to errtor return code 2.3.2  Return Code |

* + - * + Figure 12 Set Motor Stiffness Command -

#### Obtain Handheld Gimbal Motor Stiffness

CmdSet = 0x0E CmdID = 0x06 (the data segment details are shown below):

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Frame Type | Data | | | | |
| Offset | Size | Name | Type | Descriptions |
| Command frame | 0 | 1 | ctrl\_byte | uint8\_t | 0x00: No operation  0x01: Obtain handheld gimbal motor stiffness |
| Reply frame | 0 | 1 | return code | uint8\_t | Refer to error return code 2.3.2  Return Code |
| 1 | 1 | pitch\_ stiffness | uint8\_t | VALUE : 0 ~ 100 |
| 2 | 1 | yaw\_ stiffness | uint8\_t | VALUE : 0 ~ 100 |
| 3 | 1 | roll\_ stiffness | uint8\_t | VALUE : 0 ~ 100 |

* + - * + Figure 13 Obtain Motor Stiffness Command -

#### Handheld Gimbal Parameter Push Settings

CmdSet = 0x0E CmdID = 0x07 (the data segment details are shown below)

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Frame Type | Data | | | | |
| Offset | Size | Name | Type | Descriptions |
| Command frame | 0 | 1 | ctrl\_byte | uint8\_t | 0x00: No operation  0x01: Enable handheld gimbal parameter push  0x02: Disable handheld gimbal  parameter push |
| Reply frame | 0 | 1 | return code | uint8\_t | Refer to error return code 2.3.2 Return  Code |

* + - * + Figure 14 Gimbal Push Setting Command -

#### Handheld Gimbal Parameter Push

CmdSet = 0x0E CmdID = 0x08 (the data segment details are shown below):

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Frame Type | Data | | | | |
| Offset | Size | Name | Type | Descriptions |
| Command frame | 0 | 1 | ctrl\_byte | uint8\_t | 1. : Angle information valid symbol 0: Angle information currently pushed is invalid (attitude angle, joint angle)   1: Angle information currently pushed is valid (attitude angle, joint angle)   1. : Valid symbol of angle limit information   0: Angle limit information currently pushed is invalid  1: Angle limit information currently pushed is valid   1. : Valid symbol of motor stiffness information   0: Motor stiffness information currently pushed is invalid  1: Motor stiffness information  currently pushed is valid |
| 1 | 2 | yaw\_angle | int16\_t | Unit: 0.1° |
| 3 | 2 | roll\_ angle | int16\_t | Unit: 0.1° |
| 5 | 2 | pitch\_ angle | int16\_t | Unit: 0.1° |
| 7 | 2 | yaw\_joint\_agnle | int16\_t | Unit: 0.1° |
| 9 | 2 | roll\_ joint\_agnle | int16\_t | Unit: 0.1° |
| 11 | 2 | pitch\_ joint\_agnle | int16\_t | Unit: 0.1° |
| 13 | 1 | pitch\_max | uint8\_t | Max. tilt axis angle (range: 0 to 179) |
| 14 | 1 | pitch\_min | uint8\_t | Min. tilt axis angle (range: 0 to 179) |
| 15 | 1 | yaw\_max | uint8\_t | Max. pan axis angle (range: 0 to  179) |
| 16 | 1 | yaw\_min | uint8\_t | Min. pan axis angle (range: 0 to 179) |
| 17 | 1 | roll\_max | uint8\_t | Max. roll axis angle (range: 0 to 179) |
| 18 | 1 | roll\_min | uint8\_t | Min. roll axis angle (range: 0 to 179) |
| 19 | 1 | pitch\_stiffness | uint8\_t | VALUE : 0 ~ 100 |
| 20 | 1 | yaw\_ stiffness | uint8\_t | VALUE : 0 ~ 100 |
| 21 | 1 | roll\_ stiffness | uint8\_t | VALUE : 0 ~ 100 |

* + - * + Figure 15 Gimbal Parameter Push Command -

#### Obtain Module Version Number

CmdSet = 0x0E CmdID = 0x09 (the data segment details are shown below):

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Frame Type | Data | | | | |
| Offset | Size | Name | Type | Descriptions |
| Command frame | 0 | 4 | Device ID | uint32\_t | Refer to 2.3.3 Device ID Number for  specific device IDs. |
| Reply frame | 0 | 1 | Return code | uint8\_t | Refer to 2.3.2 Return Code for return  codes. |
| 1 | 4 | Device ID | uint32\_t | Refer to 2.3.3 Device ID Number for  specific device IDs. |
| 5 | 4 | Version  Number | uint32\_t | 0xAABBCCDD means that the version  is: AA.BB.CC.DD |

* + - * + Figure 16 Command Format of Obtaining the SDK Version Number -

The push frequency is 1 Hz when the device pushes the version number to DJI R SDK for displaying external device version number:

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Frame Type | Data | | | | |
| Offset | Size | Name | Type | Descriptions |
| Command frame | 0 | 4 | Device ID | uint32\_t | Refer to 2.3.3 Device ID |
| 4 | 4 | Version  Number | uint8\_t | 0xAABBCCDD means that the version  is: AA.BB.CC.DD |
| Reply frame |  |  |  |  | This command has no reply frame |

- Figure 17 Push Format of the External Device Version Number -

#### External Device Control Command Push

CmdSet = 0x0E CmdID = 0x0A (this command is used by external devices to control the gimbal. For example, the joystick or dial can use this command to control the gimbal to rotate.)

The controllers currently supported are shown below:

|  |  |
| --- | --- |
| Controller Type | Descriptions |
| 0x00 | Unknown controller |
| 0x01 | Joystick controller |
| 0x02 | Dial controller |

* + - * + Figure 18 External Controller Type -

When the gimbal uses the joystick to control, the Y and X directions of the joystick map to the pitch and yaw axes by default.

|  |  |
| --- | --- |
| Gimbal Angular Speed | Joystick Speed |
| pitch\_speed | Y\_speed |
| roll\_speed | 0 |
| yaw\_speed | X\_speed |

- Figure 19 Joystick Controller Default Mapping Relationship -

Users can use this command to change the mapping relationship when necessary. For example, the joystick can be mapped to pitch and roll axes.

|  |  |
| --- | --- |
| Gimbal Angular Speed | Joystick Speed |
| pitch\_speed | Y\_speed |
| roll\_speed | X\_speed |
| yaw\_speed | 0 |

- Figure 20 Joystick Controller Changing Mapping Relationship - The data segment details sent by the joystick controller are shown below:

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Frame Type | Data | | | | |
| Offset | Size | Name | Type | Descriptions |
| Command frame | 0 | 1 | device\_type | uint8\_t | 0x01: Joystick controller |
| 1 | 2 | pitch\_speed | int16\_t | VALUE : -15000 ~ 15000 |
| 3 | 2 | roll\_speed | int16\_t | VALUE : -15000 ~ 15000 |
| 5 | 2 | yaw\_speed | int16\_t | VALUE : -15000 ~ 15000 |
| Reply frame |  |  |  |  | This command has no reply frame |

- Figure 21 Joystick Controller Data Segment -

*adc\_value-middle\_value*

Notes: VALUE in the previous table means value= \*15000

*adc\_range*

adc\_value: ADC sample value of the current joystick middle\_value: joystick median

adc\_range: sampling precision of ADC

Users can use an external dial to control parameters such as the focus and exposure of the gimbal or the camera via the gimbal settings.

The data segment details sent by the dial controller are shown below:

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Frame Type | Data | | | | |
| Offset | Size | Name | Type | Descriptions |
| Command frame | 0 | 1 | device\_type | uint8\_t | 0x02: Dial controller |
| 1 | 2 | dial\_speed | int16\_t | VALUE : -2048 ~ 2048 |
| Reply frame |  |  |  |  | This command has no reply frame |

- Figure 22 Dial Controller Data Segment -

#### Obtain Handheld Gimbal User Parameters

CmdSet = 0x0E CmdID = 0x0B) (the gimbal user parameters can be obtained in TLV format, separately, or in combination.) TLV means ID+LENGTH+VALUE; ID refers to the command type; LENGTH refers to the VALUE length; and VALUE refers to the control status. The data type of VALUE depends on the ID. The data segment details are shown below:

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Name | Type | Length | Type | Value |
| Parameter table number selection | 0x00 | 1 | uint8\_t | 0x00: Parameter table 0 0x01: Parameter table 1  0x02: Parameter table 2 |
| Special functions under Follow mode | 0x22 | 1 | uint8\_t | [6-7] uint8\_t: 2 (Reserved bit)  [3-5] uint8\_t: 3  roll 360 mode settings 0 = normal 3-axis mode 1 = 2-axis mode  2 = ROLL 360 mode  3 = 3D\_ROLL360 mode  [2] uint8\_t: 1 Reserved bit  [1] uint8\_t: 1 Reserved bit  [0] uint8\_t: 1 Reserved bit |
| Motor special function | 0x23 | 1 | uint8\_t | VALUE:  [0] whether to power off the motor [1-7] Reserved |

- Figure 23 User Parameters Data Segment -

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Frame Type | Data | | | | |
| Offset | Size | Name | Type | Descriptions |
| Command  frame | 0 | 1-N | read\_ids | uint8\_t[1] | Read id |
| Reply frame | 0 | 1 | return code | uint8\_t | Refer to error return code 2.3.2  Return Code |
| 1 | 1~N-1 | tlv\_buffer | uint8\_t[1] | Refer to the previous table for the  TLV format |

* + - * + Figure 24 Obtain Handheld Gimbal User Parameters -

#### Handheld Gimbal Parameter Information Push Settings

CmdSet = 0x0E CmdID = 0x0C (the gimbal user parameters can be obtained in TLV format, separately, or in combination. TLV means ID+LENGTH+VALUE; ID refers to the command type; LENGTH refers to the VALUE length; VALUE refers to the control status. The data type of VALUE depends on the ID. For data segment details, refer to the user parameter data segment of 2.3.4.12 Obtaining Handheld Gimbal User Parameters

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Frame Type | Data | | | | |
| Offset | Size | Name | Type | Descriptions |
| Command  frame | 0 | 1 | tlv\_id | int8\_t | TLV id (refer to the previous table for  definition of TLV.) |

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Command frame | 1 | 2 | tlv\_length | uint8\_t | TLV data length (the data length is decided by the corresponding ID data length shown in the previous  table.) |
| 2 | 3-4 | tlv\_data | uint8\_t[2] | TLV data segment (the data length is decided by the corresponding ID data length shown in the previous  table.) |
| Reply frame | 0 | 1 | return code | uint8\_t | Refer to error return code 2.3.2 Return  Code |
| 1 | 1~N-1 | tlv\_buffer | uint8\_t[1] | For TLV format, refer to user parameter data segment of 2.3.4.12 Obtaining Handheld Gimbal User  Parameters |

- Figure 25 Set Handheld Gimbal User Parameters -

#### Handheld Gimbal Operating Mode Settings

CmdSet = 0x0E CmdID = 0x0D (the data segment details are shown below):

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Frame Type | Data | | | | |
| Offset | Size | Name | Type | Descriptions |
| Command frame | 0 | 1 | Operating  Mode | uint8\_t | 0xFE: Mode remains unchanged |
| 1 | 1 | Landscape and portrait mode | uint8\_t | 0x00: Do not switch landscape and portrait mode  0x01: Switch to landscape mode, with a 0° rotation around the X axis.  0x02: Switch to landscape mode, with a 180° rotation around the X axis.  0x03: Switch to portrait mode, with a 90° rotation around the X axis.  0x04: Switch to portrait mode, with a  -90° rotation around the X axis.  0x05: Switch between landscape and portrait mode  (the gimbal will automatically adapt to the most appropriate angle)  0xFF: Restore to default mode  (the gimbal will automatically adapt to the most appropriate angle) |
| Reply frame | 0 | 1 | return code | uint8\_t | Refer to error return code 2.3.2 Return  Code |

* + - * + Figure 26 Handheld Gimbal Operating Mode Settings -

#### Handheld Gimbal Recenter, Selfie, and Follow Modes Settings

CmdSet = 0x0E CmdID = 0x0E (the data segment details are shown below).

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Frame Type | Data | | | | |
| Offset | Size | Name | Type | Descriptions |
| Command frame | 0 | 1 | Operating Mode | uint8\_t | VALUE : 0xFE |
| 1 | 1 | Recenter and  Selfie command | uint8\_t | 0x01: execute Recenter once  0x02: execute Selfie once |
| Reply frame | 0 | 1 | return code | uint8\_t | Refer to error return code 2.3.2  Return Code |

- Figure 27 Set Handheld Gimbal Recenter and Selfie -

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Frame Type | Data | | | | |
| Offset | Size | Name | Type | Descriptions |
| Command frame | 0 | 1 | Operating Mode | uint8\_t | VALUE:  0x00: Gimbal Lock mode  0x02: Gimbal Yaw Follow mode 0x03: Sport mode |
| 1 | 1 | Recenter and Selfie  command | uint8\_t | 0x00: unchanged  Notes: When choosing the above modes, this field must be set as 0 |
| Reply frame | 0 | 1 | return code | uint8\_t | Refer to error return code 2.3.2  Return Code |

- Figure 28 Set Handheld Gimbal Follow Mode -

#### Gimbal Auto Calibration Settings

CmdSet = 0x0E CmdID = 0x0F (realize relevant functions of gimbal auto calibration such as gimbal stiffness auto calibration.) This command uses TLV format, which is ID+LENGTH+VALUE. ID refers to the command type; LENGTH refers to the VALUE length; VALUE refers to the control status. The data type of VALUE depends on the ID. This command issues multiple TLV combinations each time, realizing combined commands control. Data segment details are shown below:

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Name | Type | Length | Type | Value |
|  |  |  |  | [0]: Symbol of enabling |
|  |  |  |  | 0: Stop self-tuning |
| Control parameters self-tuning | 0x00 | 1 | uint8\_t | 1: Start self-tuning  [7:1]: Self-tuning type  0: default mode |
|  |  |  |  | 1: single attitude mode |
|  |  |  |  | Notes: Self-tuning type selects 1 |

* + - * + Figure 29 Set Gimbal Auto Calibration -

#### Gimbal Auto Calibration Status Push

CmdSet = 0x0E CmdID =0x10, (realize the progress and status push of gimbal control parameter auto calibration.) This command uses TLV format, which is ID+LENGTH+VALUE. ID refers to the command type; LENGTH refers to the VALUE length; VALUE refers to the control status. The data type of VALUE depends on the ID. The data segment details are shown below:

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Name | Type | Length | Type | Value |
| Control parameters self-tuning | 0x00 | 6 | uint8\_t | VALUE:  Byte0: Auto calibration status and result 0 = No auto calibration  0x01: auto calibration is running 0x02: auto calibration completed 0x03: auto calibration error Byte1: Auto calibration progress (Range: 0 to 100)  Byte2-5: Auto calibration error status preserved |

* + - * + Figure 30 Gimbal Auto Calibration Status Push -

#### Gimbal ActiveTrack Settings

CmdSet = 0x0E CmdID = 0x11 (the data segment details are shown below):

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Frame Type | Data | | | | |
| Offset | Size | Name | Type | Descriptions |
| Command frame | 0 | 1 | Enable ActiveTrack | uint8\_t | VALUE:  0x03: switch the start or stop status of tracking |

* + - * + Figure 31 Gimbal Auto Calibration Status Push -

#### Focus Motor Control Command

CmdSet = 0x0E CmdID = 0x12 (the details of the data segment are shown below):

The command is as follows when the command sub ID is set to focus position control and the push frequency to 100 Hz.

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Frame Type | Data | | | | |
| Offset | Size | Name | Type | Descriptions |
| Command frame | 0 | 1 | Command sub ID | uint8\_t | VALUE:  0x00: reserved  0x01: Focus Motor position control |
| 1 | 1 | Control type | uint8\_t | VALUE:  0x00: Focus control |
| 2 | 1 | Data length | uint8\_t | VALUE:  0x02: Two-byte length |
| 3 | 2 | Absolute position | uint16\_t | VALUE:  0-4095 |
| Return code |  |  |  |  | This command has no reply frame |

* + - * + Figure 32 Focus Motor Position Control Command -

Below shows the commands when the command sub ID is Focus Motor calibration:

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Frame Type | Data | | | | |
| Offset | Size | Name | Type | Descriptions |
| Command frame | 0 | 1 | Command sub ID | uint8\_t | VALUE:  0x02: Focus Motor calibration |
| 1 | 1 | Motor type | uint8\_t | VALUE:  0x00: Focus Motor |
| 2 | 1 | Calibrating Vision System Cameras | uint8\_t | VALUE:  0x00: No control  0x01: Enable auto calibration 0x02: Enable manual calibration 0x03: Reserved  0x04: Set the minimum calibration range 0x05: Set the maximum calibration range 0x06: Stop calibration  Other: Reserved |
| Return code | 0 | 1 | Return code | uint8\_t | Refer to error return code 2.3.2 Return  Error |
| 1 | 1 | Command sub ID | uint8\_t | Consistent with command frame |
| 2 | 1 | Motor type | uint8\_t | Consistent with command frame |
| 3 | 1 | Execution result | uint8\_t | VALUE:  0x00: Execution successful 0x01: Execution failed |

- Figure 33 Focus Motor Calibration Command -

Note: To set the calibration range manually, rotate the Focus Motor to a fixed position, send the calibration command 0x04 to set the minimum calibration range, rotate the motor to the next position, send the calibration command 0x05 to set the maximum calibration range, and then send the calibration command 0x02 to enable manual calibration.

### Camera Command Set Data Segment Details

#### Third-Party Camera Motion Command

CmdSet = 0x0D CmdID =0x00 (the data segment details are shown below):

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Frame Type | Data | | | | |
| Offset | Size | Name | Type | Descriptions |
|  |  |  |  |  | 0x0001: shutter |
|  |  |  |  |  | 0x0002: stop shuttering |
| Command  frame | 0 | 2 | Camera control  command | Uint16\_t | 0x0003: start recording  0x0004: stop recording |
|  |  |  |  |  | 0x0005: center focus |
|  |  |  |  |  | 0x000B: end center focus |
| Reply frame | 0 | 1 | return code | uint8\_t | Refer to error return code |

* + - * + Figure 34 Third-Party Camera Motion Command -

#### Third-Party Camera Status Obtain Command

CmdSet = 0x0D CmdID = 0x01 (the data segment details are shown below):

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Frame Type | Data | | | | |
| Offset | Size | Name | Type | Descriptions |
| Command  frame | 0 | 1 | Camera status  obtain | uint8\_t | 0x01: query recording status |
| Reply frame | 0 | 1 | return code | uint8\_t | Refer to error return code |
| 1 | 1 | Camera status | uint8\_t | VALUE:  0x00: not recording 0x02: recording |

* + - * + Figure 35 Third-Party Camera Status Obtain Command -

# Notices

## 3.1 Hardware Support

The communication interface for DJI RS 2 is CAN and its parameters are shown below:

|  |  |  |  |
| --- | --- | --- | --- |
| Baud rate | Frame type | CAN Tx | CAN Rx |
| 1M | Standard frame | 0x222 | 0x223 |

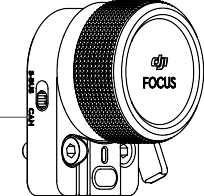
- Figure 36 CAN Communication Parameters -

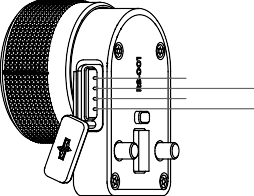
Below shows the parameters when PC communication interface is used to configure CAN:

|  |  |  |  |
| --- | --- | --- | --- |
| Baud Rate | Frame Type | CAN Tx | CAN Rx |
| 1M | Standard frame | 0x223 | 0x222 |

- Figure 37 PC CAN Configuration Parameters -

### Device Connection Diagram

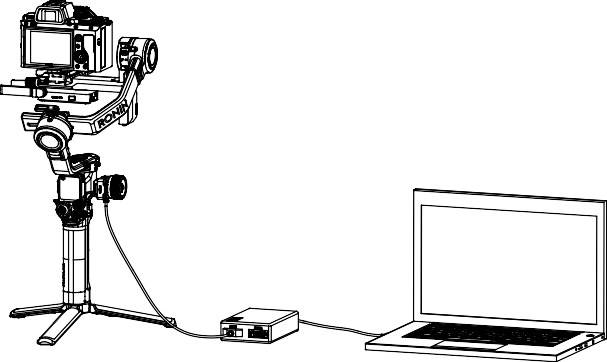
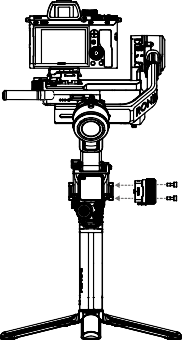
Below shows how DJI RS 2 connects to a PC via the CAN converter:



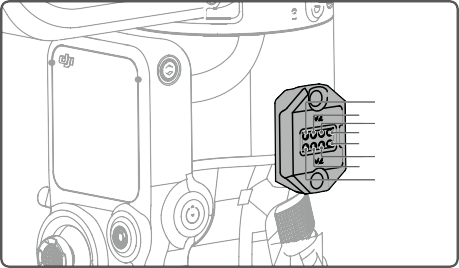
VCC\_5V CANH

GND

CAN CANL



### Ronin Series Accessories (RSA)/NATO Ports



6

4

2

1

1

3

5

6

|  |  |  |  |
| --- | --- | --- | --- |
| Pin | Signal | Description | Notes |
| 1 | VCC | Power output | Supply voltage range is 8 V ± 0.4 V, rated output current is  0.8 A, and the peak value is 1.2 A |
| 2 | CANL | CANL | / |
| 3 | SBUS\_RX | SBUS input | / |
| 4 | CANH | CANH | / |
| 5 | AD\_COM | Accessory detect port | DJI RS 2 has a built-in pull-up resistor and it is recommended to use an accessory with a 10-100k pull- down resistor. The NATO port will not output power unless  an accessory is mounted |
| 6 | GND | GND | / |

- Figure 38 RSA/NATO Ports Signal Description -

## 3.2 Software Support

The CRC16 and CRC32 parameters used in the data packet is shown below:

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| Name | Width | Poly | Init | RefIn | RefOut | XorOut |
| CRC16 | 16 | 0x8005 | 0xc55c | True | True | 0x0000 |
| CRC32 | 32 | 0x04c11db7 | 0xc55c0000 | True | True | 0x00000000 |

- Figure 39 CRC Parameters Description -

## 3.3 Command Sample

Below is a simple example of gimbal position control command to introduce how to use CRC16 and CRC32 group pack test.

The gimbal will move to a certain position once the following command is sent:

AA 1A 00 03 00 00 00 00 22 11 ***A2 42*** 0E 00 20 00 30 00 40 00 01 14 ***7B 40 97 BE***

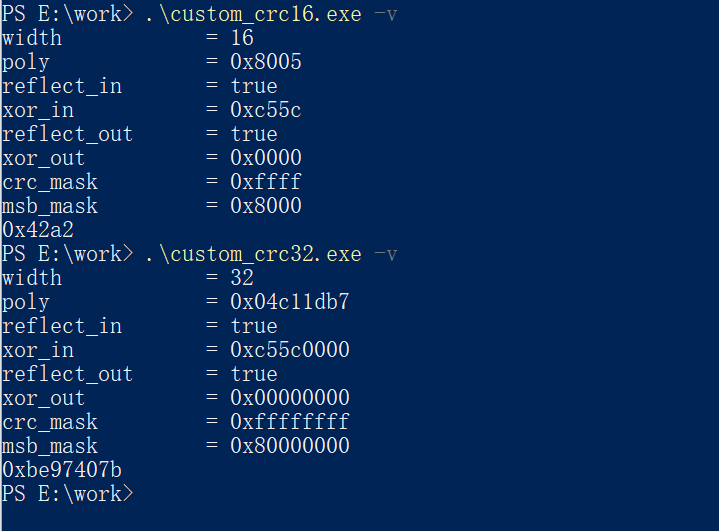
- Figure 40 CRC Parameters -

**DJI R SDK** Protocol and User Interface

## 3.4 CRC Code Sample

The CRC16 used in this protocol can refer to custom\_crc16.c, custom\_crc16.h. The CRC32 used in this protocol can refer to custom\_crc32.c, custom\_crc32.h.

Notes: An executable file that is compiled with this code can use a -v parameter to produce the corresponding CRC pattern and the CRC16 and CRC32 values generated by the command shown in Section 3.3.



- Figure 41 CRC Code Sample -

This content is subject to change.

If you have any questions about this document, please contact DJI by sending a message to [Ronin.SDK@dji.com.](mailto:Ronin.SDK@dji.com)

Copyright © 2021 DJI All Rights Reserved