```
    uint16_t receiver_ID;
}ext_student_interactive_header_data_t;
```

Content ID	Length (header length + content segment length)	Function Description
0x0200~0x02FF	6+n	Communication between own robots
0x0100	6+2	The client deletes a graphic.
0x0101	6+15	The client draws one graphic.
0x0102	6+30	The client draws two graphics.
0x0103	6+75	The client draws five graphics.
0x0104	6+105	The client draws seven graphics.
0x0110	6+45	The client draws a character graphic.

Ensure the bandwidth is properly configured since there are multiple content IDs and the maximum uplink frequency of the entire cmd_id is 10 Hz.

5. ID Description

Robot ID: 1. hero (red); 2. engineer (red); 3/4/5. standard (red); 6. aerial (red); 7. sentry (red); 9. radar station (red); 101. hero (blue); 102. engineer (blue); 103/104/105, standard (blue); 106. aerial (blue); 107. sentry (blue); 109. radar station (blue).

Client ID: 0x0101: hero operator client (red); 0x0102: engineer operator client (red); 0x0103/0x0104/0x0105: standard operator client (red); 0x0106: aerial operator client (red); 0x0165: hero operator client (blue); 0x0166: engineer operator client (blue); 0x0167/0x0168/0x0169: standard operator client (blue); 0x016A: aerial operator client (blue).

Communication between students' robots: cmd_id 0x0301; content ID: 0x0200~0x02FF

Table 5-1 Communication of interaction date between robots: 0x0301

Byte offset	Size	Description	Remarks
0	2	Data content ID	0x0200~0x02FF The above ID segments can be selected. The specific ID meaning shall be defined by the participating teams.

Byte offset	Size	Description	Remarks
2	2	Sender ID	The sender ID needs to be verified.
4	2	Receiver ID	The receiver ID needs to be verified. For example, data cannot be sent to an enemy robot ID.
6	n	Data segment	n needs to be less than 113.

```
typedef __pack struct
{
uint8_t data[]
} robot_interactive_data_t
```

Table 5-2 Communication between robots when client deletes a graphic: 0x0301

Byte offset	Size	Description	Remarks
0	2	Data content ID	0x0100
2	2	Sender ID	The sender ID needs to be verified.
4	2	Receiver ID	The receiver ID needs to be verified. For example, data can be sent to only a client corresponding to the robot.
6	1	Graphic operation	Including: 0: No operation 1: Delete a graphic layer 2: Delete all
7	1	Number of graphic layers	Number of graphic layers: 0~9

```
typedef __packed struct
{
uint8_t operate_tpye;
uint8_t layer;
} ext_client_custom _graphic_delete_t
```

Table 5-3 Graphic data

Byte offset	Size	Description	Remarks
0	3	Graphic name	Used as the index of the client in operations such as deleting and modifying
3	4	Graphic configuration	bit 0-2: Graphic operation: 0: No operation 1: add 2 : modify 3: delete bits 3-5: graphic type: 0: straight line 1: rectangle 2 : circle 3: ellipse 4: arc 5: floating number 6: integer 7: character bits 6-9: Number of graphic layers, 0-9 bits 10-13: color: 0: red and blue 1: yellow 2 : green 3: orange 4: purplish red 5: pink 6: cyan

Byte offset	Size	Description	Remarks
			7: black 8: white bits 14-22: Start angle (°), range [0, 360] bits 23-31: End angle (°), range [0, 360]
7	4	Graphic configuration 2	bits 0-9: Line width bits 10-20: Coordinate x of start point bits 21-31: Coordinate y of start point
11	4	Graphic configuration 3	bits 0-9: Font size or radius bits 10-20: Coordinate x of end point bits 21-31: Coordinate y of end point

```
typedef __packed struct
{
uint8_t graphic_name[3];
uint32_t operate_tpye:3;
uint32_t graphic_tpye:3;
uint32_t layer:4;
uint32_t color:4;
uint32_t start_angle:9;
uint32_t end_angle:9;
uint32_t width:10;
uint32_t start_x:11;
uint32_t start_y:11;
uint32_t radius:10;
uint32_t end_x:11;
uint32_t end_y:11;
} graphic_data_struct_t
```

For details about the graphic configuration, see the following table. Null indicates that the data of the field has no impact on the graphic. The recommended font size and line width ratio is 10:1.

Type	start_angle	end_angle	width	start_x	start_y	radius	end_x	end_y
Straight line	Null	Null	Line width	Coordinate x of start point	Coordinate y of start point	Null	Coordinate x of end point	Coordinate y of end point
Rectan gle	Null	Null	Line width	Coordinate x of start point	Coordinate y of start point	Null	Coordinate x of diagonal vertex	Coordinate y of diagonal vertex
Circle	Null	Null	Line width	Coordinate x of center of a circle	Coordinate y of center of a circle	Radius	Null	Null
Ellipse	Null	Null	Line width	Coordinate x of center of a circle	Coordinate y of center of a circle	Null	Length of x axis	Length of y axis
Arc	Start angle	End angle	Line width	Coordinate x of center of a circle	Coordinate y of center of a circle	Null	Length of x axis	Length of y axis
Floatin g number	Font size	Number of valid decimal places	Line width	Coordinate x of start point	Coordinate y of start point	Multiply by 1000, 32-bit integer, int32_t		
Integer	Font size	Null	Line width	Coordinate x of start point	Coordinate y of start point	32-bit integer, int32_t		
Charact er	Font size	Character length	Line width	Coordinate x of start point	Coordinate y of start point	Null	Null	Null

Table 5-4 Communication between robots when client draws one graphic: 0x0301

Byte offset	Size	Description	Remarks
0	2	Data content ID	0x0101

Byte offset	Size	Description	Remarks
2	2	Sender ID	The sender ID needs to be verified.
4	2	Receiver ID	The receiver ID needs to be verified. For example, data can be sent to only a client corresponding to the robot.
6	15	Graphic 1	See graphic data introduction for details

```
typedef __packed struct
{
  graphic_data_struct_t   grapic_data_struct;
} ext_client_custom_graphic_single_t;
```

Table 5-5 Communication between robots when client draws two graphics: 0x0301

Byte offset	Size	Description	Remarks
0	2	Data content ID	0x0102
2	2	Sender ID	The sender ID needs to be verified.
4	2	Receiver ID	The receiver ID needs to be verified. For example, data can be sent to only a client corresponding to the robot.
6	15	Graphic 1	See graphic data introduction for details
21	15	Graphic 2	See graphic data introduction for details

```
typedef __packed struct
{
   graphic_data_struct_t   grapic_data_struct[2];
} ext_client_custom_graphic_double_t;
```

Table 5-6 Communication between robots when client draws five graphics: 0x0301

Byte offset	Size	Description	Remarks
0	2	Data content ID	0x0103
2	2	Sender ID	The sender ID needs to be verified.
4	2	Receiver ID	The receiver ID needs to be verified. For example, data can be sent to only a client corresponding to the robot.

Byte offset	Size	Description	Remarks
6	15	Graphic 1	See graphic data introduction for details
21	15	Graphic 2	See graphic data introduction for details
36	15	Graphic 3	See graphic data introduction for details
51	15	Graphic 4	See graphic data introduction for details
66	15	Graphic 5	See graphic data introduction for details

```
typedef __packed struct
{
  graphic_data_struct_t   grapic_data_struct[5];
} ext_client_custom_graphic_five_t;
```

Table 5-7 Communication between robots when client draws seven graphics: 0x0301

Byte offset	Size	Description	Remarks
0	2	Data content ID	0x0104
2	2	Sender ID	The sender ID needs to be verified.
4	2	Receiver ID	The receiver ID needs to be verified. For example, data can be sent to only a client corresponding to the robot.
6	15	Graphic 1	See graphic data introduction for details
21	15	Graphic 2	See graphic data introduction for details
36	15	Graphic 3	See graphic data introduction for details
51	15	Graphic 4	See graphic data introduction for details
66	15	Graphic 5	See graphic data introduction for details
81	15	Graphic 6	See graphic data introduction for details
96	15	Graphic 7	See graphic data introduction for details

```
typedef __packed struct
{
    graphic_data_struct_t   grapic_data_struct[7];
} ext_client_custom_graphic_seven_t;
```

Table 5-8 Communication between robots when client draws a character: 0x0301

Byte offset	Size	Description	Remarks
0	2	Data content ID	0x0110
2	2	Sender ID	The sender ID needs to be verified.
4	2	Receiver ID	The receiver ID needs to be verified. For example, data can be sent to only a client corresponding to the robot.
6	15	Character configuration	See graphic data introduction for details
21	30	Character	

```
typedef __packed struct
{
    graphic_data_struct_t   grapic_data_struct;
    uint8_t data[30];
} ext_client_custom_character_t;
```

6. Custom Controller Interaction Data

Custom controller data includes a unified segment header structure. The data segment is a content data segment. The total length of the entire interaction data packet has a maximum of 39 bytes, and nine bytes of frame_header, cmd_id, and frame_tail are subtracted. Therefore, the content data segment sent has a maximum of 30 bytes. The downlink frequency of the entire interaction data packet 0x0302 is 30 Hz.

Table 6-1 Interaction data receiving information: 0x0302. Transmission frequency: Transmitted at a frequency of 30 Hz at most

Byte offset	Size	Description	Remarks
0	x	Content segment data	The maximum value of x is 30.

```
typedef __pack struct
{
uint8_t data[]
} robot_interactive_data_t
```