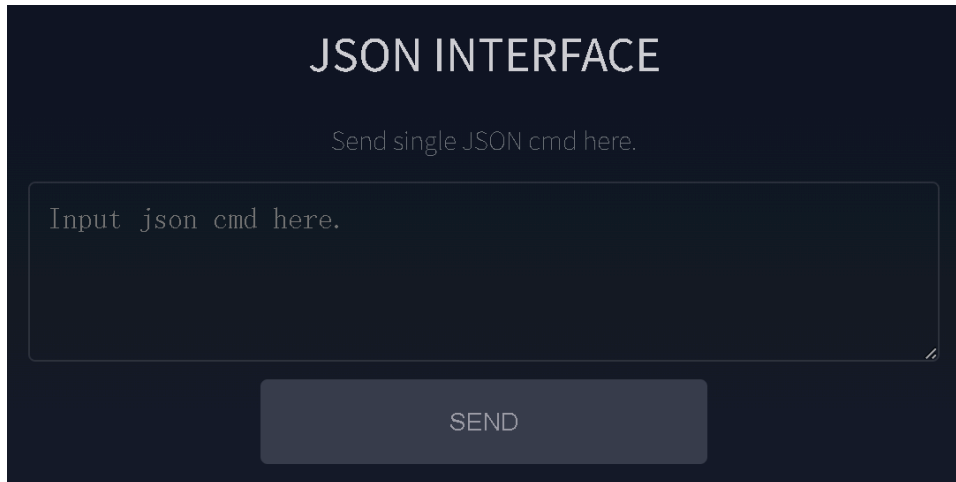


# JSON Command Interaction — Web Application

The image shows a web application interface titled "JSON INTERFACE" in white text on a dark blue background. Below the title, there is a subtitle "Send single JSON cmd here." in a lighter blue font. A large, dark blue rectangular input box with a thin white border contains the placeholder text "Input json cmd here." in a light blue font. At the bottom center of the interface is a dark blue button with the word "SEND" in white capital letters.

The input box in this section looks similar to the one above, but it serves a different purpose.

This field is designed specifically for sending **single JSON commands**, and the command entered here can be sent directly to the driver board by clicking the **SEND** button.

There is **no need** to upload the command to the device before execution — it enables **real-time JSON command interaction** through the web interface.

Below this component is the **JSON Command Library**.

Users can click on any command in the list, and that command will be automatically filled into the **JSON Interface** input box.

You may then edit the command as needed and press **SEND** to transmit it immediately to the driver board.

JSON commands form the **core communication mechanism** between the user and the driver board.

Almost all features of the system are controlled through JSON commands — not only servo control, but also Wi-Fi, ESP-NOW, OLED display, buzzer output, file system operations, and various other device settings.

We chose this interaction method not only for its flexibility and scalability, but also because JSON commands are easy for users to generate programmatically on external devices.

They are human-readable, highly debuggable, and well-suited for script-based automation.

Refer to the [JSON Command Communication Guide](#) section for details on each available JSON command and their functions.

## JSON LIST

CMD\_BREAK\_LOOP  
{ "T":0 }

INPUT

## LinkArm Ctrl

CMD\_LINK\_ARM\_SC\_JOINTS\_CTRL\_ANGLE  
{ "T":134, "ang":[0,0,0,0] }

INPUT

CMD\_LINK\_ARM\_SC\_JOINTS\_CTRL\_RAD  
{ "T":135, "rad":[0,0,0,0] }

INPUT

CMD\_XYZG\_CTRL  
{ "T":136, "xyzg":[260.5,0,122.38,50] }

INPUT

CMD\_FPV\_ABS\_CTRL  
{ "T":137, "rbzg":[260.5,0,122.38,50] }

INPUT

CMD\_SMOOTH\_XYZG\_CTRL  
{ "T":138, "xyzg":[236.5,0,122.38,0], "spd":0.4 }

INPUT

CMD\_SMOOTH\_FPV\_ABS\_CTRL  
{ "T":139, "rbzg":[236.5,0,122.38,0], "spd":0.4, "br":200 }

INPUT

CMD\_SET\_MAX\_JOINTS\_SPEED  
{ "T":140, "spd":1.2 }

INPUT

CMD\_SET\_LINK\_ARM\_FEEDBACK\_FLAG  
{ "T":145, "flag":0, "hz":10 }

INPUT

CMD\_CONST\_CTRL  
{ "T":146, "flag":0, "axis":0, "delta":0.01 }

INPUT

## SC Servo

CMD\_SC\_CTRL  
{ "T":31, "id":1, "pos":511, "time":0, "spd":0 }

INPUT

CMD\_SC\_CHANGE\_ID  
{ "T":33, "old\_id":1, "new\_id":2 }

INPUT

CMD\_SC\_TORQUE\_LOCK