

08 Sub-controller JSON Command Set

From Waveshare Wiki

Jump to: navigation, search

In the previous chapter, we introduced a simple demo where we sent motion control commands from the host to the sub-controller. The microcontroller is capable of receiving a wide variety of commands, and in this chapter, we will introduce these commands.

Composition of JSON Commands

Taking the command `{"T":1,"L":0.2,"R":0.2}` sent in the previous chapter as an example, the **T** value in this JSON data represents the command type, while the **L** and **R** values represent the target linear velocities for the left and right wheels, respectively. The unit for linear velocity is by default meters per second (m/s). In summary, this command is a motion control command where the motion parameters are the target linear velocities for the left and right wheels. All subsequent JSON commands will include a **T** value to define the command type, but the specific command parameters will vary depending on the type of command.

JSON Command Set

You can view the definitions of these commands in the `json_cmd.h` file of our open-source microcontroller demo, or add new sub-controller functionalities yourself.

Motion Control Commands

These commands are fundamental to the mobile robot and are used for motion-related control functions. Each command below includes three parts: an example, a brief introduction, and a detailed description.

CMD_SPEED_CTRL

- `{"T":1,"L":0.5,"R":0.5}`
- Sets the target linear velocity for both wheels (velocity closed-loop control).

L and R represent the target linear velocities for the left and right wheels, respectively, in m/s. Negative values indicate reverse rotation and 0 means stop. The range of target linear velocities depends on the motors/reducers/wheel diameters used in the product, and the relevant calculation formulas can be found in the open-source microcontroller demo. It's important to note that for chassis using brushed DC motors, when the given target velocity's absolute value is very small (but not 0), the motor's poor low-speed performance may cause significant speed fluctuations during movement.

CMD_PWM_INPUT

- `{"T":11,"L":164,"R":164}`
- Sets the PWM value for both drive wheels (velocity open-loop control).

L and R represent the PWM values for the left and right wheels, respectively, with a range of -255 to 255. Negative values indicate the reverse direction and an absolute value of 255 means 100% PWM, indicating full power operation for that wheel.

CMD_ROS_CTRL

- `{"T":13,"X":0.1,"Z":0.3}`
- ROS control (velocity closed-loop control).

This command is for ROS-based host computer control of the chassis movement. X represents the linear velocity in m/s, which can be negative; Z represents the angular velocity in rad/s, which can also be negative.

CMD_SET_MOTOR_PID

- {"T":2,"P":20,"I":2000,"D":0,"L":255}
- PID controller settings.

This command is used to tune the PID controller. The PID parameters in the example above are the default parameters for this product. L represents WINDUP_LIMITS, which is an interface currently not used in the product.

OLED Display Control Commands

The product comes with an OLED display, which communicates with the ESP32 module of the microcontroller via I2C. The host computer can send JSON commands to change the content displayed on the screen.

CMD_OLED_CTRL

- {"T":3,"lineNum":0,"Text":"putYourTextHere"}
- Controls the display of custom content on the screen.

lineNum is the line number. A single JSON command can change the content of one line. Upon receiving a new command, the default OLED screen that appears at startup will disappear, replaced by your added content. For most products using a 0.91-inch OLED display, lineNum can be 0, 1, 2, or 3, totaling four lines. Text is the content you want to display on this line. If the content is too long for one line, it will automatically wrap to the next line, but this may also push off the last line of content.

CMD_OLED_DEFAULT

- {"T":-3}
- Controls the display to show the default startup screen.

Use this command to revert the OLED display to the default image shown at startup.

Module Type

The mobile chassis can be equipped with different types of modules (none/mechanical arm/gimbal). This command tells the microcontroller about the currently installed module type. This command is usually sent automatically by the host computer to the microcontroller at startup, and more details will be provided in later chapters.

CMD_MODULE_TYPE

- {"T":4,"cmd":0}
- Sets the module type.

The cmd value represents the type of module. Currently, there are three options: 0 for no module, 1 for a robotic arm, and 2 for the pan-tilt.

IMU Related Functions

The chassis is equipped with an IMU sensor. You can use the following commands to obtain data from the IMU sensor. It's important to note that after the product is powered on, the continuous feedback function for chassis information (including IMU data) is enabled by default. The IMU-related functions are only necessary when the continuous feedback function is disabled.

CMD_GET_IMU_DATA

- {"T":126}
- Retrieves IMU data.

This command allows retrieval of data from the IMU sensor upon sending.

CMD_CALI_IMU_STEP

- {"T":127}
- IMU calibration (reserved interface).

Current product programs do not require calibration; this command is a reserved interface for future use.

CMD_GET_IMU_OFFSET

- {"T":128}
- Retrieves current IMU offsets (reserved interface).

This command can provide feedback on the offset values for each axis of the current IMU.

CMD_SET_IMU_OFFSET

- {"T":129,"x":-12,"y":0,"z":0}
- Sets the IMU offsets (reserved interface).

This command allows setting the offset values for each axis of the IMU. It is a reserved command and not required for current products.

Chassis Information Feedback

CMD_BASE_FEEDBACK

- {"T":130}
- Chassis information feedback.

After the product is powered on, chassis information feedback is typically enabled by default and occurs automatically. If the continuous feedback function for chassis information is disabled, and there's a need to obtain information about the chassis at a single instance, this command can be used to acquire basic chassis data.

CMD_BASE_FEEDBACK_FLOW

- {"T":131,"cmd":1}
- Continuous chassis information feedback.

Setting the cmd value to 1 enables this function, which is by default activated and continuously provides chassis information. Setting the cmd value to 0 disables this function. Once disabled, the host computer can use the CMD_BASE_FEEDBACK command to obtain chassis information.

CMD_FEEDBACK_FLOW_INTERVAL

- {"T":142,"cmd":0}
- Sets the interval for continuous feedback.

The cmd value is the interval time to be set, in milliseconds (ms). This command allows adjusting the frequency of chassis feedback information.

CMD_UART_ECHO_MODE

- {"T":143,"cmd":0}
- Sets the command echo mode.

When the cmd value is set to 0, echo is disabled. When the cmd value is set to 1, echo is enabled, which means the sub-controller will output the commands it receives, facilitating debugging and verification processes.

WiFi Configuration

CMD_WIFI_ON_BOOT

- {"T":401,"cmd":3}
- Set WiFi Mode at Boot.

cmd value 0 turns off WiFi; 1 sets to AP mode; 2 sets to STA mode; 3 sets to AP+STA mode.

CMD_SET_AP

- {"T":402,"ssid":"UGV","password":"12345678"}
- Configure SSID and Password for AP Mode (ESP32 as a Hotspot).

CMD_SET_STA

- {"T":403,"ssid":"WIFI_NAME","password":"WIFI_PASSWORD"}
- Configure SSID and Password for STA Mode (ESP32 connects to a known hotspot).

CMD_WIFI_APSTA

- {"T":404,"ap_ssid":"UGV","ap_password":"12345678","sta_ssid":"WIFI_NAME","sta_password":"WIFI_PASSWORD"}
- Set Names and Passwords for AP and STA Modes (AP+STA Mode).

CMD_WIFI_INFO

- {"T":405}
- Get Current WiFi Information.

CMD_WIFI_CONFIG_CREATE_BY_STATUS

- {"T":406}
- Create a New WiFi Configuration File Using Current Settings.

CMD_WIFI_CONFIG_CREATE_BY_INPUT

- {"T":407,"mode":3,"ap_ssid":"UGV","ap_password":"12345678","sta_ssid":"WIFI_NAME","sta_password":"WIFI_PASSWORD"}
- Create a New WiFi Configuration File Using Input Settings.

CMD_WIFI_STOP

- {"T":408}
- Disconnect WiFi Connection.

12V Switch and Gimbal Settings

CMD_LED_CTRL

- {"T":132,"IO4":255,"IO5":255}
- 12V Switch Output Settings.

The device's sub-controller board features two 12V switch interfaces, each with two ports, totaling four ports. This command allows you to set the output voltage of these ports. When the value is set to 255, it corresponds to the voltage of a 3S battery. By default, these ports are used to control LED lights, and you can use this command to adjust the brightness of the LEDs.

CMD_GIMBAL_CTRL_SIMPLE

- {"T":133,"X":0,"Y":0,"SPD":0,"ACC":0}
- Basic Gimbal Control Command.

This command is used to control the orientation of the gimbal. X represents the horizontal orientation in degrees, with positive values turning right and negative values turning left, ranging from -180 to 180 degrees. Y represents the vertical orientation in degrees, with positive values tilting up and negative values tilting down, ranging from -30 to 90 degrees. SPD stands for speed, and ACC for acceleration; when set to 0, they indicate the fastest speed/acceleration.

CMD_GIMBAL_CTRL_MOVE

- {"T":134,"X":45,"Y":45,"SX":300,"SY":300}
- Continuous Gimbal Control Command.

This command is for continuous control over the pan-tilt's orientation. X and Y function similarly to the basic control command, specifying the desired horizontal and vertical orientations, respectively. SX and SY represent the speeds for the X and Y axes, respectively.

CMD_GIMBAL_CTRL_STOP

- {"T":135}
- Pan-tilt Stop Command.

This command can be used to immediately stop the pan-tilt's movement initiated by the previous commands.

CMD_GIMBAL_STEADY

- {"T":137,"s":0,"y":0}
- Pan-tilt Stabilization Feature.

Setting s to 0 turns off this feature, and setting it to 1 enables it. When enabled, the pan-tilt automatically adjusts its vertical angle using IMU data to maintain stability. The y parameter sets the target angle between the pan-tilt and the ground, allowing the camera to look up and down even when the stabilization feature is active.

CMD_GIMBAL_USER_CTRL

- {"T":141,"X":0,"Y":0,"SPD":300}
- Pan-tilt UI Control.

This command is intended for pan-tilt control via a UI interface. The X value can be -1, 0, or 1, where -1 rotates left, 0 stops, and 1 rotates right. The Y value can also be -1, 0, or 1, where -1 tilts down, 0 stops, and 1 tilts up. SPD specifies the speed of the operation.

Robotic Arm Control

CMD_MOVE_INIT

- {"T":100}
- Moves the Robotic Arm to Its Initial Position.

Normally, the robotic arm automatically moves to its initial position upon powering up. This command may cause process blocking.

CMD_SINGLE_JOINT_CTRL

- {"T":101,"joint":0,"rad":0,"spd":0,"acc":10}
- Single Joint Motion Control.
- joint: Joint number.
 - 1: for BASE_JOINT
 - 2: for SHOULDER_JOINT
 - 3: for ELBOW_JOINT
 - 4: for EOAT_JOINT (wrist/claw joint)
- rad: Angle to rotate to (in radians), based on the initial position of each joint. Default angles and rotation directions for each joint are provided.

- The initial position default angle for BASE_JOINT is 0, with a rotation range of -3.14 to 3.14. When the angle increases, the base joint rotates to the left; when the angle decreases, the base joint rotates to the right.
- The initial position default angle for SHOULDER_JOINT is 0, with a rotation range of -1.57 to 1.57. When the angle increases, the shoulder joint rotates forward; when the angle decreases, the shoulder joint rotates backward.
- The initial position default angle for ELBOW_JOINT is 1.570796, with a rotation range of -1.11 to 3.14. When the angle increases, the elbow joint rotates downward; when the angle decreases, the elbow joint rotates in the opposite direction.
- The initial position default angle for EOAT_JOINT is 3.141593. For the default gripper joint, the rotation range is 1.08 to 3.14, and when the angle decreases, the gripper joint opens. If changed to a wrist joint, the rotation range is 1.08 to 5.20, and when the angle increases, the wrist joint rotates downward; when the angle decreases, the wrist joint rotates upward.
- spd: Rotation speed in steps per second, with one full rotation equaling 4096 steps. A higher value indicates faster rotation; a value of 0 rotates at maximum speed.
- acc: Acceleration at the start and end of the rotation, smoother with lower values, ranging from 0-254 in units of 100 steps/sec². An acc value of 0 signifies maximum acceleration.

CMD_JOINTS_RAD_CTRL

- {"T":102,"base":0,"shoulder":0,"elbow":1.57,"hand":1.57,"spd":0,"acc":10}
- Full Joint Rotation Control in Radians.
- base: The angle of the base joint, with rotation range as described in the "CMD_SINGLE_JOINT_CTRL" command above in the "rad" key.
- shoulder: The angle of the shoulder joint.
- elbow: The angle of the elbow joint.
- hand: The angle of the gripper/wrist joint.
- spd: The speed of rotation, measured in steps per second. A servo motor completes one full rotation in 4096 steps. A higher numerical value corresponds to a faster speed. When the speed value is 0, rotation occurs at maximum speed.
- acc: The acceleration at the start and end of rotation. A smaller value results in smoother acceleration and deceleration. The value can range from 0 to 254, measured in 100 steps per second squared. For example, setting it to 10 would mean acceleration and deceleration occur at 1000 steps per second squared. When the acceleration value is 0, the maximum acceleration is used.

CMD_SINGLE_AXIS_CTRL

- {"T":103,"axis":2,"pos":0,"spd":0.25}
- Single Axis Coordinate Control.
- axis specifies the axis: 1-x, 2-y, 3-z, 4-t, with units in mm for all but the T axis, which is in radians. spd is the speed coefficient, with higher values indicating faster movement.

CMD_XYZT_GOAL_CTRL

- {"T":104,"x":235,"y":0,"z":234,"t":3.14,"spd":0.25}
- Robotic Arm Coordinate Motion Control (Inverse Kinematics).

This function causes blocking.

CMD_XYZT_DIRECT_CTRL

- {"T":1041,"x":235,"y":0,"z":234,"t":3.14}
- Robotic Arm Coordinate Motion Control (Inverse Kinematics).

This function does not cause blocking.

CMD_SERVO_RAD_FEEDBACK

- {"T":105}
- Provides feedback on the robotic arm's coordinates.

CMD_EOAT_HAND_CTRL

- {"T":106,"cmd":1.57,"spd":0,"acc":0}
- End-of-Arm Tool Control in Radians.
- cmd: Angle to rotate to (in radians). The initial position default angle for EOAT_JOINT is 3.141593.

- For the default clamp joint, the rotation range is from 1.08 to 3.14, and when the angle decreases, the clamp joint opens.
- If changed to a wrist joint, the rotation range is from 1.08 to 5.20, and when the angle increases, the wrist joint rotates downward; when the angle decreases, the wrist joint rotates upward.
- spd: The speed of rotation, measured in steps per second. A servo motor completes one full rotation in 4096 steps. A higher numerical value corresponds to a faster speed. When the speed value is 0, rotation occurs at maximum speed.
- acc: The acceleration at the start and end of rotation. A smaller numerical value results in smoother acceleration and deceleration. The value can range from 0 to 254, measured in 100 steps per second squared. For example, setting it to 10 would mean acceleration and deceleration occur at 1000 steps per second squared. When the acceleration value is 0, the maximum acceleration is used.

CMD_EOAT_GRAB_TORQUE

- {"T":107,"tor":200}
- Clamp Force Control.

The tor value can go up to 1000, representing 100% force.

CMD_SET_JOINT_PID

- {"T":108,"joint":3,"p":16,"i":0}
- Joint PID Settings.

CMD_RESET_PID

- {"T":109}
- Resets Joint PID Settings.

CMD_SET_NEW_X

- {"T":110,"xAxisAngle":0}
- Sets a New Direction for the X Axis.

CMD_DYNAMIC_ADAPTATION

- {"T":112,"mode":0,"b":1000,"s":1000,"e":1000,"h":1000}
- Dynamic External Force Adaptation Control.

Other Settings

CMD_HEART_BEAT_SET

- {"T":136,"cmd":3000}
- Sets the Heartbeat Function Interval.

The cmd unit is milliseconds. This command sets the interval for the heartbeat function. If the sub-controller does not receive a new motion command within this time, it will automatically stop movement. This feature helps prevent continuous movement in case the host crashes.

CMD_SET_SPD_RATE

- {"T":138,"L":1,"R":1}
- Sets the Speed Ratio for Left and Right.

Due to potential errors in encoders or tire traction, the device might not move straight even when both wheels are set to the same speed. This command allows for fine-tuning the speed of the left and right wheels to correct this issue. For example, if the left wheel needs to rotate slower, you can change the value of L to 0.98. Try not to set the values of L and R greater than one.

CMD_GET_SPD_RATE

- {"T":139}
- Retrieves the Current Speed Ratio.

This command fetches the current speed ratio settings.

ESP-NOW Related Settings

CMD_BROADCAST_FOLLOWER

- {"T":300,"mode":1}
- {"T":300,"mode":0,"mac":"CC:DB:A7:5B:E4:1C"}
- Sets the mode for ESP-NOW broadcast control.

When mode is 1, other devices can control it via broadcast commands; when mode is 0, only devices with the specified MAC address can control it.

CMD_GET_MAC_ADDRESS

- {"T":302}
- Retrieves the current device's MAC address.

CMD_ESP_NOW_ADD_FOLLOWER

- {"T":303,"mac":"FF:FF:FF:FF:FF:FF"}
- Adds a MAC address to the controlled device (PEER).

CMD_ESP_NOW_REMOVE_FOLLOWER

- {"T":304,"mac":"FF:FF:FF:FF:FF:FF"}
- Removes a MAC address from the PEER.

CMD_ESP_NOW_GROUP_CTRL

- {"T":305,"dev":0,"b":0,"s":0,"e":1.57,"h":1.57,"cmd":0,"megs":"hello!"}
- ESP-NOW group control.

CMD_ESP_NOW_SINGLE

- {"T":306,"mac":"FF:FF:FF:FF:FF:FF","dev":0,"b":0,"s":0,"e":1.57,"h":1.57,"cmd":0,"megs":"hello!"}
- ESP-NOW unicast/group control.

Task File Related Functions

This functionality belongs to the advanced features of the microcontroller and is usually not required when using the host controller.

CMD_SCAN_FILES

- {"T":200}
- Scans the current task files.

CMD_CREATE_FILE

- {"T":201,"name":"file.txt","content":"inputContentHere."}
- Creates a new task file.

CMD_READ_FILE

- {"T":202,"name":"file.txt"}
- Reads a task file.

CMD_DELETE_FILE

- {"T":203,"name":"file.txt"}

- Deletes a task file.

CMD_APPEND_LINE

- {"T":204,"name":"file.txt","content":"inputContentHere."}
- Adds a new instruction at the end of a task file.

CMD_INSERT_LINE

- {"T":205,"name":"file.txt","lineNum":3,"content":"content"}
- Insert a new instruction in the middle of a task file.

CMD_REPLACE_LINE

- {"T":206,"name":"file.txt","lineNum":3,"content":"Content"}
- Replaces an instruction in a task file.

Servo Settings

CMD_SET_SERVO_ID

- {"T":501,"raw":1,"new":11}
- Changes the servo ID.

raw is the servo's original ID (new servos are all set to 1), and the new one is the ID to be changed to, which must not exceed 254, cannot be negative, and 255 is reserved as the broadcast ID.

CMD_SET_MIDDLE

- {"T":502,"id":11}
- Sets the current position of the servo as the middle position (only valid for ST series servos).

CMD_SET_SERVO_PID

- {"T":503,"id":14,"p":16}
- Sets the P value of the servo's PID.

ESP32 Related Features

CMD_REBOOT

- {"T":600}
- Reboot the ESP32.

CMD_FREE_FLASH_SPACE

- {"T":601}
- Retrieves the remaining space size in the FLASH memory.

CMD_BOOT_MISSION_INFO

- {"T":602}
- Outputs the current boot mission file.

CMD_RESET_BOOT_MISSION

- {"T":603}
- Resets the boot mission file to its default or a predetermined state.

CMD_NVS_CLEAR

- {"T":604}
- Clears the ESP32's Non-Volatile Storage (NVS) area. This command can be useful if there are issues with establishing a WiFi connection. It's recommended to reboot the ESP32 after executing this command.

CMD_INFO_PRINT

- {"T":605,"cmd":1}
- Sets the mode for information feedback.

- When cmd is set to 1, it enables the printing of debug information. Setting cmd to 2 enables continuous feedback of chassis information. Setting cmd to 0 turns off feedback, meaning no information will be provided.

*Retrieved from "https://www.waveshare.com/w/index.php?title=08_Sub-controller_JSON_Command_Set&oldid=109835
(https://www.waveshare.com/w/index.php?title=08_Sub-controller_JSON_Command_Set&oldid=109835)"*
