

M5Stack Chain Angle Protocol								All packets start with 0xAA 0x55 and end with 0x55 0xAA				V1 (Version)
												2025/10/23
Instruction set	Byte Instruction format	0	1	2	3	4	5	6	7	8	9	10
		Length_low	Length_high	Index	Cmd	Data1	Data2	Data3	Data4	Data5	Data6	Data7
Set RGB values	Command packet format	0x05+Num*3	0x00	Index_id	0x20	Index	Num	R	G	B	...	CRC
	Response packet format	0x04	0x00	Index_id	0x20	Operation_status	CRC					
	Command Details	(1) Function Description: Sets RGB values. (2) Input Parameters: Index_id (Device ID), Index (RGB light start coordinates), Num (Number of RGB lights to be set), RGB value (3) Return Parameter: Operation_status (4) Instruction Code: 0x20 Note 1: Operation_status Operation status 0: Operation failed 1: Operation successful Note 2: Index is the RGB light start coordinate, starting from 0. Angle devices have only one RGB light, so Index = 0, Num = 1										
Get RGB values	Command packet format	0x05	0x00	Index_id	0x21	Index	Num	CRC				
	Response packet format	0x04 + Num * 3 / 0x04	0x00	Index_id	0x21	Operation_status	R	G	B	...	CRC	
	Command Details	(1) Function Description: Get RGB values. (2) Input Parameters: Index_id (Device ID), Index (RGB light start coordinates), Num (Number of RGB lights to be read) (3) Return Parameters: RGB values (4) Instruction Code: 0x21 Note 1: Operation_status Operation status 0: Operation failed 1: Operation successful Note 2: Index is the RGB light start coordinate, starting from 0. Angle device has only one RGB light, so Index = 0, Num = 1 Note 3: Operation failure returns a data packet that does not contain RGB values.										
Set RGB light brightness	Command packet format	0x05	0x00	Index_id	0x22	Light	Save_to_flash	CRC				
	Response packet format	0x04	0x00	Index_id	0x22	Operation_status	CRC					
	Command Details	(1) Function description: Set the brightness of RGB lights (2) Input parameters: Index_id (device index ID), Light (RGB brightness, 0~100, default 40), Save_to_flash (whether to save to internal Flash) (3) Return parameters: Operation_status (4) Instruction code: 0x22 Note 1: Operation_status Operation status 0: Operation failed 1: Operation successful Note 2: Save_to_flash Whether to save to internal Flash 0: Do not save 1: Save Note 3: The setting takes effect immediately upon success Note 4: Saving to internal Flash requires erasing the page, which takes a relatively long time (about 20ms). During this process, the serial port interrupt will be turned off, and frequent operation will affect the lifespan of the device. If you need to frequently adjust the brightness, please avoid saving to internal Flash every time.										
Get RGB light brightness	Command packet format	0x03	0x00	Index_id	0x23	CRC						
	Response packet format	0x04	0x00	Index_id	0x23	Light	CRC					
	Command Details	(1) Function Description: Get the brightness of the RGB light. (2) Input Parameter: Index_id (Device Index ID) (3) Return Parameter: Light (RGB brightness) (4) Instruction Code: 0x23										
Get 12-bit ADC	Command packet format	0x03	0x00	Index_id	0x30	CRC						
	Response packet format	0x05	0x00	Index_id	0x30	ADC_low	ADC_high	CRC				
	Command Details	(1) Function description: Read the 12-bit ADC of Angle. (2) Input parameter: Index_id (device index ID) (3) Return parameter: ADC value (4) Instruction code: 0x30 Note 1: Angle_ADC = (uint16_t)((ADC_high < < 8) ADC_low) Note 2: The theoretical value range is 0 ~ 4095, but due to the difference of the device, there is an error offset of 0 to 10 in both the upper and lower limits, so the actual usable range will be reduced accordingly.										
Get 8-bit ADC	Command packet format	0x03	0x00	Index_id	0x31	CRC						
	Response packet format	0x04	0x00	Index_id	0x31	ADC	CRC					
	Command Details	(1) Function description: Read the 8-bit ADC of Angle. (2) Input parameter: Index_id (device index ID) (3) Return parameter: ADC value (4) Instruction code: 0x31 Note 1: Angle_ADC = ADC Note 2: The theoretical value range is 0 ~ 255, but due to the difference of the device, there is an error offset of 0 ~ 3 in the upper and lower limits, so the actual usable range will be reduced accordingly.										
Set rotation control status	Command packet format	0x05	0x00	Index_id	0x32	Clockwise_direct	Save_to_flash	CRC				
	Response packet format	0x04	0x00	Index_id	0x32	Operation_status	CRC					
	Command Details	(1) Function description: Sets the rotation control state, determining whether the encoder value increases or decreases when rotating clockwise. (2) Input parameters: Index_id (device subscript ID), Clockwise_direct (clockwise state direction), Save_to_flash (whether to save the configuration to internal Flash) (3) Return parameter: Operation_status (4) Instruction code: 0x32 Note 1: Clockwise_direct Encoder direction 0: Clockwise decrease 1: Clockwise increase (default state) Note 2: Operation_status Operation status 0: Operation failed 1: Operation successful Note 3: Save_to_flash Whether to save to internal Flash 0: Do not save 1: Save Note 4: Setting takes effect immediately upon success Note 5: Saving to internal Flash requires erasing the page, which takes a relatively long time (about 20ms). During this process, the serial port interrupt will be turned off, and frequent operations will affect the lifespan of the device.										
Obtain rotation control status	Command packet format	0x03	0x00	Index_id	0x33	CRC						
	Response packet format	0x04	0x00	Index_id	0x33	Clockwise_direct	CRC					
	Command Details	(1) Function description: Get the rotation control status. (2) Input parameter: Index_id (device index ID) (3) Return parameter: Clockwise_direct value (4) Instruction code: 0x33 Note 1: Clockwise_direct encoder direction 0: decrease clockwise 1: increase clockwise (default state)										

Note 1: The maximum data packet length is 256 bytes.

Note 2: Data length is from Index_id to CRC, including Index_id and CRC, but excluding the data length itself.

Note 3: When calculating CRC, the packet header, packet trailer, length, and the CRC field itself need to be excluded; only the remaining data is summed.

Note 4: Serial communication baud rate is 115200, 8 data bits, 1 stop bit, no parity bit.

```
uint8_t calculateCRC(const uint8_t *buffer, uint16_t size)
{
    uint8_t crc8 = 0;
    for (uint8_t i = 4; i < (size - 3); i++) {
        crc8 += buffer[i];
    }
    return crc8;
}
```