



M5Stack Unit Roller485 Protocol(Readback)												All returns start with 0xAA 0x55										V1 (FW Version) 2025/6/3	
bytes	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	note				
command	Cmd ID	Motor ID	Data1 Byte0	Data1 Byte1	Data1 Byte2	Data1 Byte3	Data2 Byte0	Data2 Byte1	Data2 Byte2	Data2 Byte3	Data3 Byte0	Data3 Byte1	Data3 Byte2	Data3 Byte3	Data4	Data5	Data6	CRC8					
Readback 0	0x40	ID	0	CRC8																			
Return	0x50	ID	Speed Readback k-byte0	Speed Readback k-byte1	Speed Readback k-byte2	Speed Readback k-byte3	Position Readback k-byte0	Position Readback k-byte1	Position Readback k-byte2	Position Readback k-byte3	Current Readback k-byte0	Current Readback k-byte1	Current Readback k-byte2	Current Readback k-byte3	Mode	Status	Error	CRC8	Speed Readback = Speed Readback-byte0 + Speed Readback-byte1 * 256 + Speed Readback-byte2 * 65536 + Speed Readback-byte3 * 16777216 Actual Speed Readback = Speed Readback/100	Position Readback = Position Readback-byte0 + Position Readback-byte1 * 256 + Position Readback-byte2 * 65536 + Position Readback-byte3 * 16777216 Actual Position Readback = Position Readback/100	Current Readback = Current Readback-byte0 + Current Readback-byte1 * 256 + Current Readback-byte2 * 65536 + Current Readback-byte3 * 16777216 Actual Current Readback = Current Readback/100	Mode 1: Speed Mode 2: Position Mode 3: Current Mode 4: Encoder Mode  Status: 0: Standby; 1: Running; 2: Error  Error: 1: Overvoltage; 2: Stalled; 4: Over Range	
Readback 1	0x41	ID	0	CRC8																VIN X100 = VIN X100-byte0 + VIN X100-byte1 * 256 + VIN X100-byte2 * 65536 + VIN X100-byte3 * 16777216 Actual VIN = VIN X100/100	Temp = Temp-byte0 + Temp-byte1 * 256 + Temp-byte2 * 65536 + Temp-byte3 * 16777216	Encoder Counter = Encoder Counter-byte0 + Encoder Counter-byte1 * 256 + Encoder Counter-byte2 * 65536 + Encoder Counter-byte3 * 16777216 Can only use in Encoder mode	RGB Mode: 0, Sys-default 1, User-define  RGB Brightness: 0~100
Return	0x51	ID	VIN X100-byte0	VIN X100-byte1	VIN X100-byte2	VIN X100-byte3	Temp-byte0	Temp-byte1	Temp-byte2	Temp-byte3	Encoder Counter-byte0	Encoder Counter-byte1	Encoder Counter-byte2	Encoder Counter-byte3	RGB Mode	RGB Brightness	0	CRC8	P/I/D(uint32_t): PID = PID-byte0 + PID-byte1 * 256 + PID-byte2 * 65536 + PID-byte3 * 16777216 For example: P=0.00001, P setting value=0.00001*100000=1, P-byte0=1, P-byte1=0, P-byte2=0, P-byte3=0	P/I/D(uint32_t): PID = PID-byte0 + PID-byte1 * 256 + PID-byte2 * 65536 + PID-byte3 * 16777216 For example: P=0.00001, P setting value=0.00001*100000=1, P-byte0=1, P-byte1=0, P-byte2=0, P-byte3=0	P/I/D(uint32_t): PID = PID-byte0 + PID-byte1 * 256 + PID-byte2 * 65536 + PID-byte3 * 16777216 For example: P=0.00001, P setting value=0.00001*100000=1, P-byte0=1, P-byte1=0, P-byte2=0, P-byte3=0	P/I/D(uint32_t): PID = PID-byte0 + PID-byte1 * 256 + PID-byte2 * 65536 + PID-byte3 * 16777216 For example: P=0.00001, P setting value=0.00001*100000=1, P-byte0=1, P-byte1=0, P-byte2=0, P-byte3=0	
Readback 2	0x42	ID	0	CRC8																P/I/D(uint32_t): PID = PID-byte0 + PID-byte1 * 256 + PID-byte2 * 65536 + PID-byte3 * 16777216 For example: P=0.00001, P setting value=0.00001*100000=1, P-byte0=1, P-byte1=0, P-byte2=0, P-byte3=0	P/I/D(uint32_t): PID = PID-byte0 + PID-byte1 * 256 + PID-byte2 * 65536 + PID-byte3 * 16777216 For example: P=0.00001, P setting value=0.00001*100000=1, P-byte0=1, P-byte1=0, P-byte2=0, P-byte3=0	P/I/D(uint32_t): PID = PID-byte0 + PID-byte1 * 256 + PID-byte2 * 65536 + PID-byte3 * 16777216 For example: P=0.00001, P setting value=0.00001*100000=1, P-byte0=1, P-byte1=0, P-byte2=0, P-byte3=0	P/I/D(uint32_t): PID = PID-byte0 + PID-byte1 * 256 + PID-byte2 * 65536 + PID-byte3 * 16777216 For example: P=0.00001, P setting value=0.00001*100000=1, P-byte0=1, P-byte1=0, P-byte2=0, P-byte3=0
Return	0x52	ID	Speed-P-X100000-byte0	Speed-P-X100000-byte1	Speed-P-X100000-byte2	Speed-P-X100000-byte3	Speed-I-X100000-00-byte0	Speed-I-X100000-00-byte1	Speed-I-X100000-00-byte2	Speed-I-X100000-00-byte3	Speed-D-X100000-byte0	Speed-D-X100000-byte1	Speed-D-X100000-byte2	Speed-D-X100000-byte3	RGB-B	RGB-G	RGB-R	CRC8	(1) CRC8: uint8_t crc8(uint8_t *data, uint8_t len) { uint8_t i, crc = 0x00;  while(len--) { crc ^= *data++; for(i = 0; i < 8; i++) { if(crc & 0x01) { crc = (crc >> 1) ^ 0x8c; } else crc >>= 1; } } return crc; }	(2) 所有返回的消息以0xAA 0x55开头, 0x5A 0x55 不参与CRC校验			

