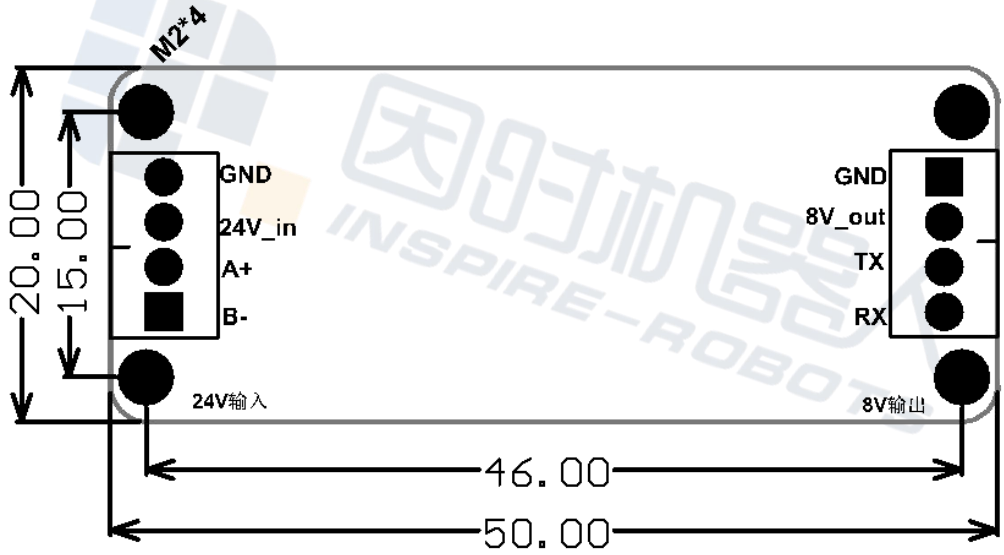
# ACTUATOR WITH PLC CONTROL INSTRUCTIONS

Beijing Inspire-Robots Technology Co., Ltd. | March 2023

ID: PRJ-01-TS-U-017 | Page Number: 10 pages | Version: V0.0.1

# Actuator with PLC Control Instructions

For the motion of the Servo Actuator with PLC Control, the RS485 converter (model: AED-LA-82-12) is required to convert its level and communication format. The RS485 converter can convert DC 24 V to DC 8 V to supply power to the servo actuator, and convert the RS485 level to the LVTTL 3.3 V level.


| Left signal of the circuit board (connected to the master control terminal) | | | Right signal of the circuit board (connected to the linear actuator) | | | |
|---|---|---|---|---|---|---|
| GND | GND | Input | GND | GND | Output | Black cable of the actuator |
| VCC | 24V_in | Input (24V1A) | VCC | 8V_out | Output (8V2A) | Red cable of the push rod |
| Positive terminal of 485 | A+ | Differential input (positive) | Positive terminal of 485 | TX | Transmission via 3.3 V serial port | Yellow cable of the push rod |
| Negative terminal of 485 | B- | Differential input (negative) | Negative terminal of 485 | RX | Receiving via 3.3 V serial port | Blue cable of the push rod |

**Note:** This is a passthrough module where the master control terminal sends the instruction data via RS485 from TX to the linear push rod; the feedback data from the linear push rod is converted to the RS485 signal via RX, and then such signal is uploaded to the master control terminal. The maximum baud rate is 115200bps. A module has one linear push rod.

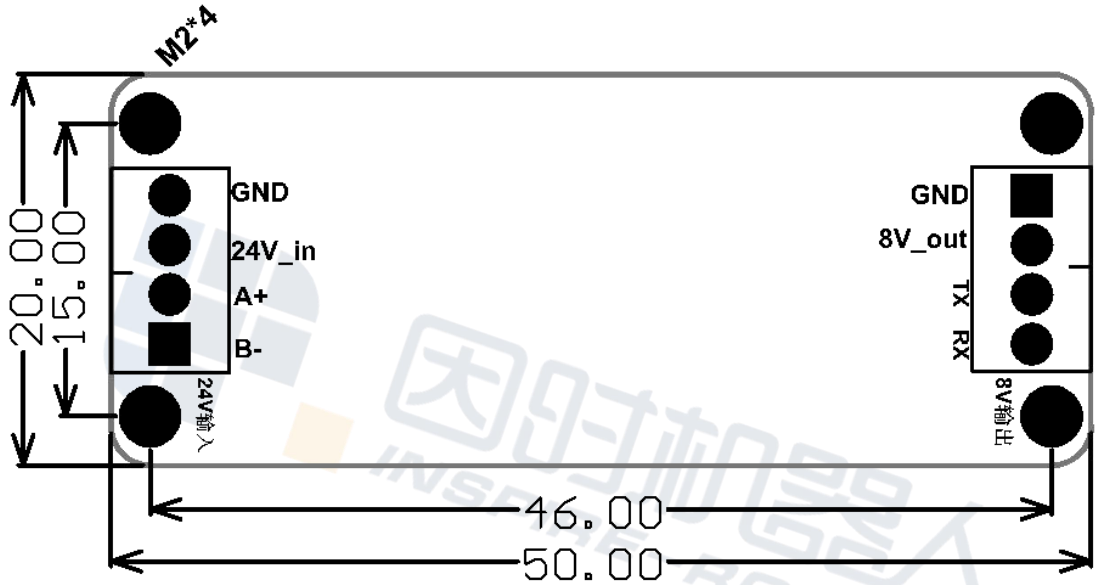
## Product: Modbus RTU Converter

## Model: AED-LA-92-12

**Function description:**

1. The built-in MCU can convert the control instruction of RS485 in the Modbus RTU format to the instruction for motion control with LVTTL 3.3 V level that is suitable for the Micro Linear Servo Actuator;

2. 24 V to 8 V step-down module




| Left signal of the circuit board (connected to the master control terminal) | | | Right signal of the circuit board (connected to the linear push rod) | | | |
|---|---|---|---|---|---|---|
| GND | GND | Input | GND | GND | Output | Black cable of the push rod |
| VCC | 24V_in | Input (24V1A) | VCC | 8V_out | Output (8V2A) | Red cable of the push rod |
| Positive terminal of 485 | A+ | Differential input (positive) | Positive terminal of 485 | TX | Transmission via 3.3 V serial port | Yellow cable of the push rod |
| Negative terminal of 485 | B- | Differential input (negative) | Negative terminal of 485 | RX | Receiving via 3.3 V serial port | Blue cable of the push rod |

**Note:** This is a protocol conversion module where the master control terminal converts the Modbus protocol command via the RS485 interface to the instruction that

meets the motion control requirement of the linear push rod, and sends such instruction to the linear push rod via TX; the feedback data from the linear push rod is converted to the Modbus protocol after it is received via RX, and then is uploaded to the master control terminal via the RS485 interface. The maximum baud rate at the Modbus terminal is 115200bps. A module has one linear push rod. Modbus instructions are listed below:

Modbus RTU interface protocol

Communication data structure

9600 (baud rate); 8 (data bit); Even (parity bit); 1 (start bit); 1 (stop bit)

| | |
|---|---|
| Start | Holding the state without input data for ≥10 ms |
| Slave station address | Slave station address: 8 bits binary address |
| Command code | Command code: 8 bits binary address |
| Data (n-1) | Data content<br>n × 8 bits binary number, n≤202 |
| ... | |
| Data (0) | |
| Low-order byte in CRC checksum | CRC checksum<br>CRC checksum consists of two 8 bits binary numbers. |
| High-order byte in CRC checksum | |
| End | Holding the state without input data for ≥10 ms |

The format of data characters depends on the command code. The valid description of the command code is shown below:

| Command code (Hex) | Description | Actionable unit |
|---|---|---|
| 01 (01H) | Reading node state (the input node state cannot be read) | S, Y, M, T, C |
| 02 (02H) | Reading node state (the input node state can be read) | S, X, Y, M, T, C |
| 03 (03H) | Reading the content of register | T, C, D |
| 05 (05H) | Forced separate node state On/Off | S, Y, M, T, C |
| 06 (06H) | Preset value of separate register | T, C, D |
| 15 (0FH) | Forced multiple nodes state On/Off | S, Y, M, T, C |
| 16 (10H) | Preset values of multiple registers | T, C, D |
| 17 (11H) | Reporting the slave station address | None |
| 23 (17H) | PLC LINK executes the read and write functions simultaneously in a | None |

| | | |
|---|---|---|
| | polling period. | |

Cyclic redundancy check (CRC) starts from the "Slave station address" until "the last data content" ends. The CRC calculation method is described below:

**Step 1:** Load a 16 bits register ("CRC Register") with the content of FFFF (hexadecimal system).

**Step 2:** Execute the XOR operation of the 8 bits data for the first byte in the instruction information and the 8 bits data for the low-order bytes in the CRC register.

The operation result is stored in the CRC register.

**Step 3:** Shift the content of the CRC register rightward by 1 bit, and enter 0 in its highest bit.

**Step 4:** Check the value of the lowest bit in the CRC register. If it is 0, repeat Step 3; if it is 1, execute the XOR operation of the content of the CRC register and A001 (hexadecimal system), and store the operation result in the CRC register.

**Step 5:** Repeat Steps 3 and 4 until the content of the CRC register is shifted rightward by 8 bits. At that time, the first byte in the instruction information has been processed completely.

**Step 6:** For the next byte in the instruction information, repeat the actions in Steps 2 to 5 until all bytes in the instruction information are processed completely. The last content of the CRC register is the cyclic redundancy check (CRC) value. When the CRC value is transmitted in the instruction information, the high-order and low-order bytes in the calculated CRC value must be interchanged, i.e., low-order bytes in the CRC value should be transmitted first.

Example: Read the data of 8 consecutive character sets with the PLC number of 01 and the address of H0614-H61B (T20-T27). Read the value of the slave station equipment (communication address: 1).

PC→PLC

"01 03 06 14 00 08 04 80"

Information transmitted:

| Field | Data (hexadecimal number) |
|---|---|
| Start | Holding the state without input data for ≥10 ms |
| Slave station address | 01 |
| Command code | 03 |
| Data start address | 06 |
| | 14 |

| c | Data (hexadecimal number) |
|---|---|
| Data count (unit: byte) | 00 |
| | 08 |
| Low-order byte in CRC checksum | 04 |
| High-order byte in CRC checksum | 80 |
| End | Holding the state without input data for ≥10 ms |

Response information:

| Field | Data (hexadecimal number) |
|---|---|
| Start | Holding the state without input data for ≥10 ms |
| Slave station address | 01 |
| Command code | 03 |
| Data count (unit: byte) | 10 |
| High-order byte in data (T20) | 00 |
| Low-order byte in data (T20) | 01 |
| High-order byte in data (T21) | 00 |
| Low-order byte in data (T21) | 02 |
| High-order byte in data (T22) | 00 |
| Low-order byte in data (T22) | 03 |
| High-order byte in data (T23) | 00 |
| Low-order byte in data (T23) | 04 |
| High-order byte in data (T24) | 00 |
| Low-order byte in data (T24) | 05 |
| High-order byte in data (T25) | 00 |

| | |
|---|---|
| Low-order byte in data (T25) | 06 |
| High-order byte in data (T26) | 00 |
| Low-order byte in data (T26) | 07 |
| High-order byte in data (T27) | 00 |
| Low-order byte in data (T27) | 08 |
| Low-order byte in CRC checksum | 72 |
| High-order byte in CRC checksum | 98 |
| End | Holding the state without input data for ≥10 ms |

Read and write the register values through Command Code 03 (read) and Command Code 06 (write) of Modbus RTU to execute function commands.

The communication data content is listed below.

| Modbus address | Name | Range | Description | | |
|---|---|---|---|---|---|
| H0001 | CMD-ID | H0001-H00FE | Linear servo equipment address (ID) | Read and write | To become effective immediately |
| H0002 | CMD-BAUD | H0001--1200<br>H0002--2400<br>H0003--4800<br>H0004-9600 (def)<br>H0005--14400<br>H0006--19200<br>H0007--38400<br>H0008--56000<br>H0009--57600<br>H000A—115200 | Communication baud rate setting | Read and write | To become effective after re-power on |
| H0003 | CMD-SAVE | H0000<br>H0001 (effective) | Parameter saving | Read and write | To become effective after re-power on |
| H0010 | CMD-STOP | H0000<br>H0001 (effective) | Emergency stop | Read and write | To become effective immediately |
| H0011 | CMD-RESTART | H0000<br>H0001 (effective) | Recovery | Read and write | To become effective immediately |
| H0012 | CMD-FAULTACK | H0000 | Fault clearance | Read and | To become effective |

| | | | | | |
|---|---|---|---|---|---|
| | | H0001 (effective) | | write | immediately |
| H0014 | CMD-SETSPEED | Numerical range: 1-Vmax or 5000 (unit: step/s)<br>Vmax is the no-load speed of the actuator.<br>For example, the maximum speed of LA10-02 is<br>17 mm/s, so Vmax=17*2000/10=3400<br>steps/s;<br>When this register is set to 1-Vmax,<br>the push rod will move to the target position at the set speed.<br>When this register is set to 5000,<br>the push rod will move to the target position with its maximum capacity (motor operating under the full voltage). | Speed regulation parameter setting | Read and write | To become effective immediately |
| H001F | CMD-SETFORCE | -15000 to 15000 (decimal system)<br>HC568~H3A98<br>Unit: g | Force control parameter setting (for products with closed-loop regulation of force control) | Read and write | To become effective immediately |
| H0020 | CMD-SETPOS | 0 to 2000 (decimal system)<br>H0000~H07D0 | Setting the motion position of the servo actuator | Read and write | To become effective immediately |
| H0021 | CMD_CURPOS | -100 to 2100 (decimal system)<br>HFF9C~H0834 | Current position of the servo actuator | Read | To become effective immediately |
| H0022 | CMD_CURTEMP | -20 to 100 (decimal system)<br>HFFEC-H0064 | Current temperature of the servo actuator (℃) | Read | To become effective immediately |
| H0023 | CMD_CURCUR | 0 to 2000 (decimal system)<br>H0000~H07D0 | Current of the servo actuator (mA) | Read | To become effective immediately |
| H0024 | CMD_CURERR | H0001: protection from | Error code of the servo | Read | To become effective |

| | | | | | |
|---|---|---|---|---|---|
| | | locked-rotor<br>H0002: over temperature protection<br>H0004: overcurrent protection<br>H0008: abnormal operation of the motor | actuator | | immediately |
| H0028 | CMD_SETFORCE | -15000 to +15000 (decimal system)<br>HC568-H3A98, unit: g | Set value for force control<br>(for speed and force control mode) | Read<br>Write | To become effective immediately |
| H0029 | CMD_SETSPEED | 0 to 10000 steps/s (decimal system)<br>H0000~H2710 | Set value for speed<br>(for speed and force control mode ) | Read<br>Write | To become effective immediately |
| H002A | CMD_SETPOS | 0-2000 steps (2000 steps corresponding to the full stroke)<br>(decimal system)<br>H0000~H07D0 | Set value for position<br>(for speed and force control mode) | Read<br>Write | To become effective immediately |