1. Principle of Device Master-Slave Communication

1. Preface

This chapter aims to introduce the detailed information about the master-slave relationship during the communication between MaxArm and various devices such as STM32, 51 microcontroller, Arduino, and Raspberry Pi, which can help users to master how MaxArm functions as a slave device to communicate with other devices and how other devices act as master controllers to control MaxArm.

Throughout this chapter, MaxArm serves as a slave device, communicating with other devices via UART serial communication for data transmission.

2. Master-Slave Relationship

In a master-slave control system, MaxArm serves as a slave device, while other microcontrollers and devices act as the master.

2.1 The Functions of MaxArm as the Slave Device

1) Receive and parse the received signal from the master:

Waiting for serial port signals, if the data received from the serial port is not none, the serial data is parsed according to the communication protocol, and the corresponding functions can be invoked based on the data information.

2) Call the functionalities of MaxArm according to the received data.

When the signals are parsed, the corresponding MaxArm requires to be invoked. For example, the motion control of bus servo, the motion control of

the PWM servo at the end-effector, the working mode of the suction cup at the end-effector.

3) Data encapsulation and feedback:

When a read command is received, the corresponding read function needs to be invoked. After reading the data, it is encapsulated into a data packet according to the communication protocol and send to the master device.

2.2 Other Devices as the Masters

1) Command Encapsulation and Transmission:

The master needs to encapsulate control commands and data into data packets according to the communication protocol and send them to the device.

2) Control Coordination :

The master needs to manage the overall system's collaborative operation, ensuring the smooth communication and operation between MaxArm and other devices, maintaining a goods working state.

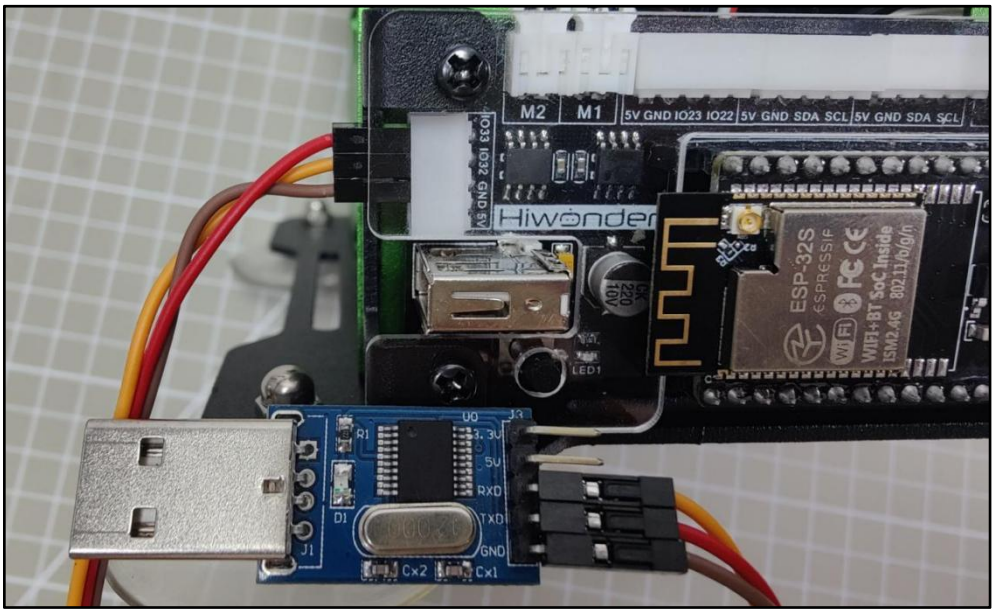
3) Receive Data:

When the master reads the servo status,it sends a read command and then receives status information send by MaxArm to ensure the integrity and correctness of the data. Consequently, it parses the data packet to extract the useful information.

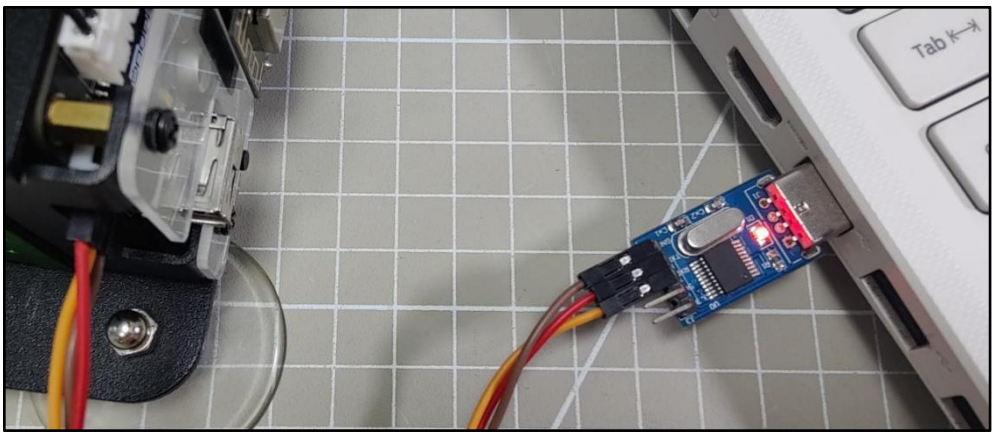
3. Hardware Wiring

Here will use an example of MaxArm connected to a PC:

1) Connect the RXD, TXD, GND of USB adapter to IO32, IO33, GND ports of ESP32 expansion board respectively using Dupont wires.



2) Connect the USB adapter into your computer.



4. Data Transmission Format

The default UART serial port data transmission format for MaxArm is as follow:

Baud rate	9600
Data bits	8

Parity bit	None
Stop bit	1

5. Communication Protocol

Frame header	**Function code**	**Data length**	**Data information**	**Check bit**
0xAA 0x55	func	len	data	check

Frame header: when the frame header is received, it indicates data is transmitting. The fixed frame header length is 2 bytes.

Function code: Used to indicate the purpose of an information frame.

Data length: Indicate the amount of the subsequent data.

Data information: Represents the transmitted data.

Parity bit: (Function Code + Data Length + Data) is negated and the low byte is taken.

6. Notice

1) The power source for the master and MaxArm robotic arm can be different, but they must share a common ground when connected to ensure the stable communication voltage levels.

2) When connecting devices, please note that the TX and RX pins of UART serial port must be crossed, otherwise communication will not be possible.