# AUBO Modbus Slave Configuration

AUBO (Beijing) Robotics Technology CO., LTD.

# Statement

- It is strictly forbidden to reprint any of the contents of this material.
- The information in this material is subject to change without notice and should not be regarded as a commitment by AUBO (Beijing) Robotics Technology Co., Ltd.
- Please read this manual before installing and using the product.
- Please keep this manual to read and as reference any time.
- The pictures in this manual are for reference only, please refer to the actual product received.

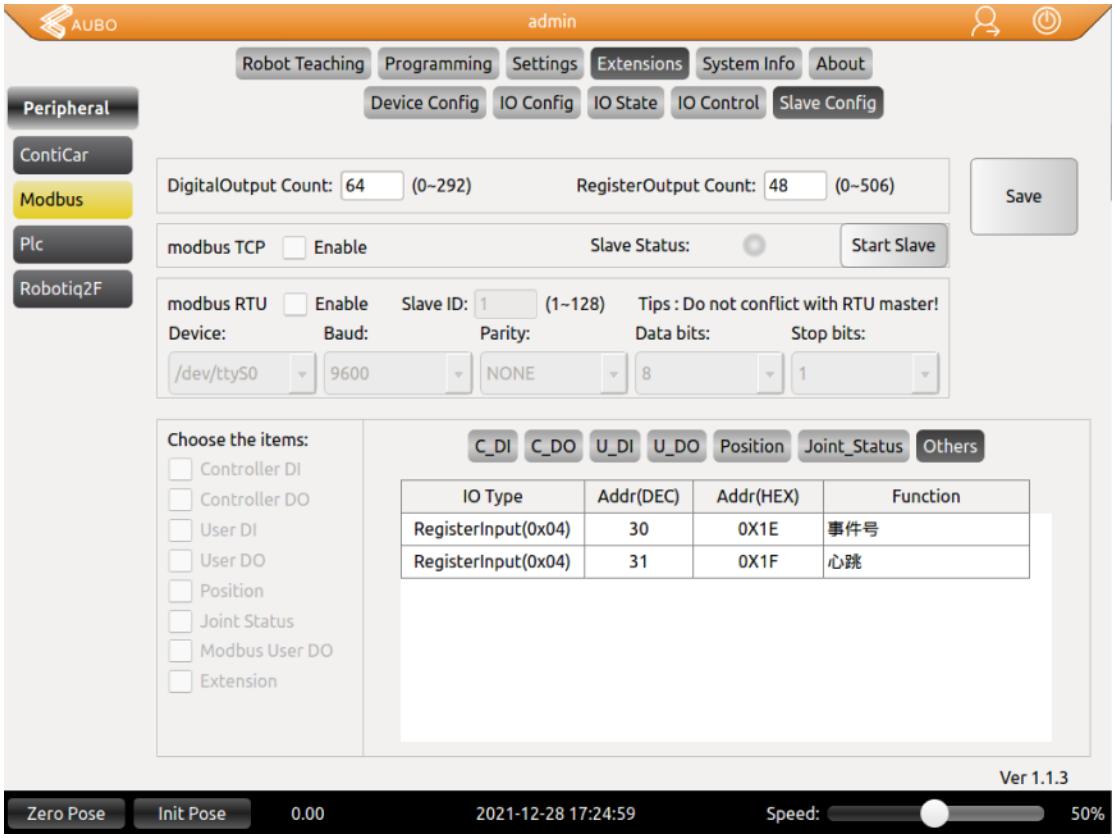
# Configuration and

# Usage of Modbus Slave

## 1. Introduction

Modbus slave functionality has been added to AUBOPE, including Modbus-RTU slave and Modbus-TCP server.

A Modbus-RTU master station or a Modbus-TCP client can communicate with the robot or read a part of its status. The slave configuration interface is in **[Extension]→[peripheral]→[MODBUS]→[slave configuration]**.




## 2. Interface description

The interface is mainly divided into 7 parts:

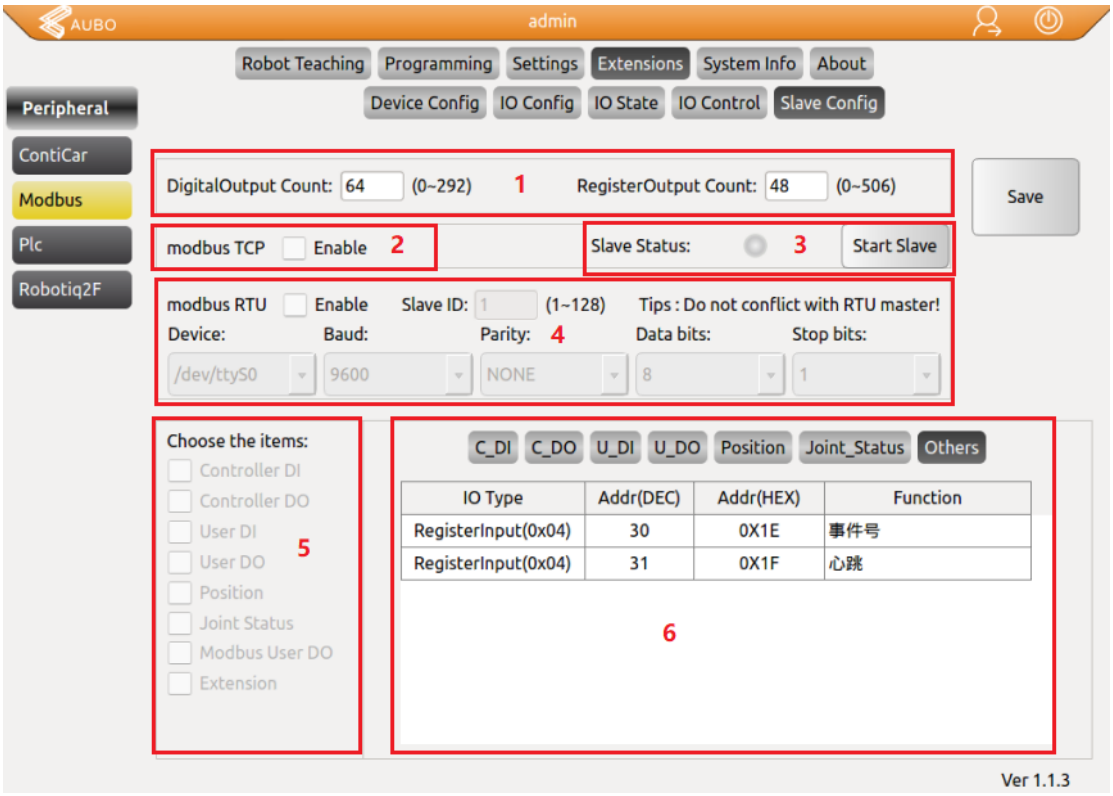
1. By default, there are 64 coils (digital output) and 48 registers (register output). This data is required for PROFINET protocol conversion compatible with smart200. The number can be modified to no more than 128 if this approach is not utilized for PROFINET communication.

2. The modbusTCP server enable: By default, the server is turned off. If you need to turn it on, please check **[Enable]**, and press **[Save]**. It will take effect after restarting the device. Server port **502**, station number **255 (0xFF)**.
3. Slave status display: The slave service ON indicator is green and the slave OFF indicator is gray.
4. Start and configure Modbus-RTU slave station. By default, the slave station is turned off. If it needs to be turned on, please check **[enable]**, and press **[Save]**. It will not take effect until the device is restarted.
   a) Slave ID is the slave station number, which defaults to 1 and can be modified to a value between 1 and 128. Save and restart after modification.
   b) Device: Select the corresponding serial port number. This data is more important, please read it carefully first. If you use the RS485-DB9 serial port that comes with the control box, please select /dev/ttyS2 for the device number. If you use the USB port of the control box to switch to 485 communication, /dev/ttyUSB will be refreshed in the device and select it. If you need to use the robot's modbusRTU master station and slave station at the same time, be sure to connect the two to different serial ports (for example, the master station uses its own RS485-DB9 port, and the slave station uses USB to 485), otherwise there will be conflicts.
   c) Baud rate：9600/19200/38400/115200.
   d) Data bit：8.
   e) Stop bit：1.
5. Status monitoring items. By default, no robot status is monitored. If necessary, please check the corresponding entry, save and restart to take effect. For **[Modbus control user DO function]** and **[Extension]**, see the modbus_slave address correspondence table for specific addresses.

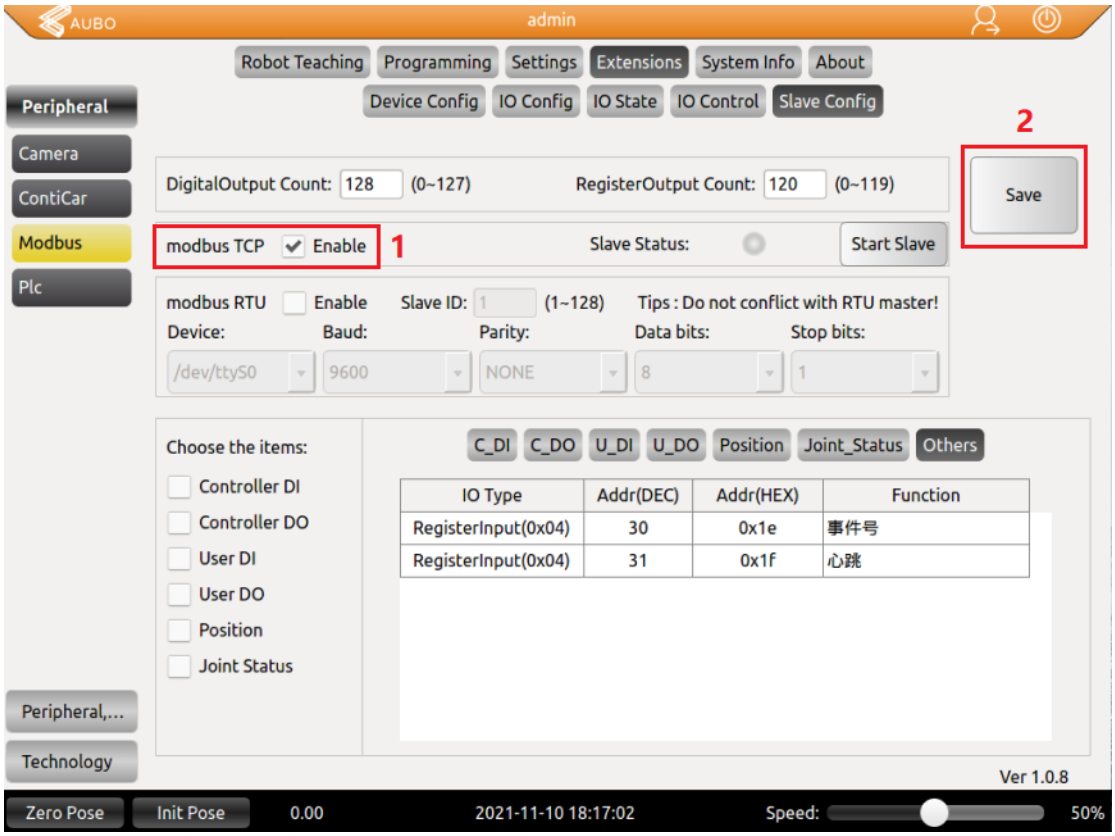
Note: The number of monitoring states will affect the overall communication speed of modbus, so please check only the necessary items.

6. The status monitoring address table is convenient for query.
7. Save: After modifying the configuration, you must click **[Save]** to write to the system. It will take effect after restarting.



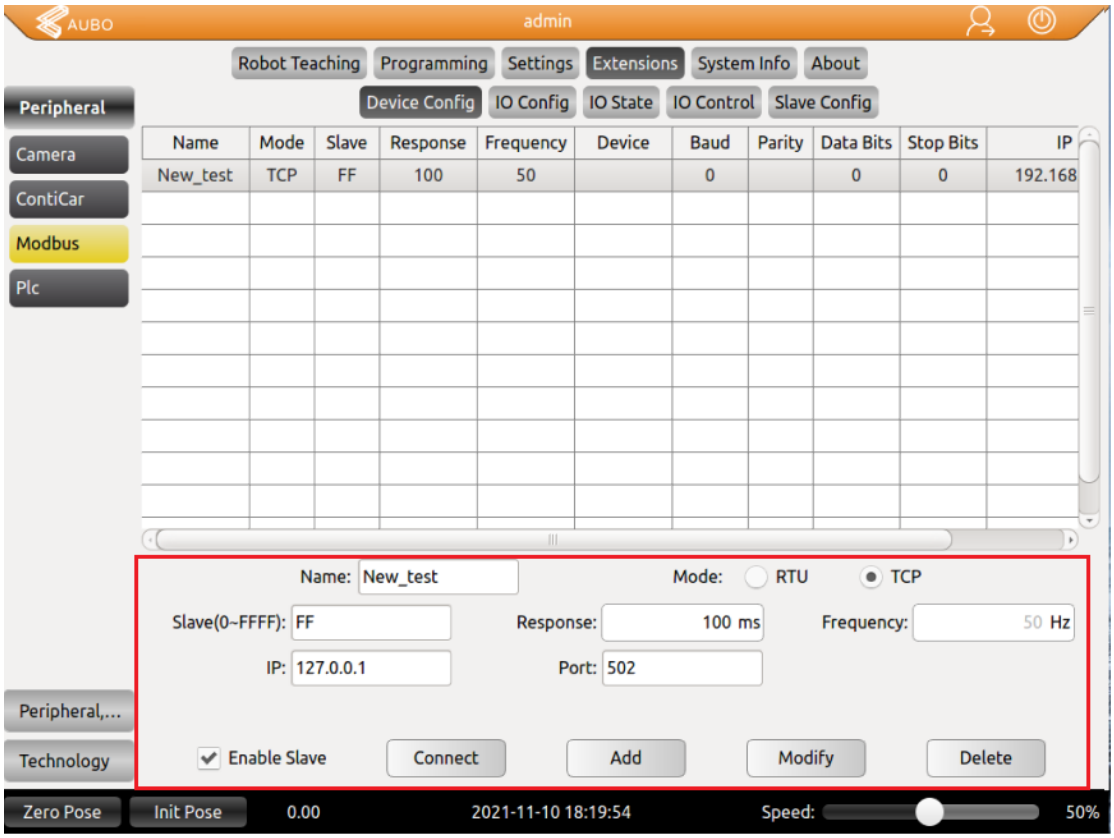

## 3. Usage setting of Modbus-TCP server

1. In **[Slave configuration]** window**,** check modbus TCP, save and restart the system.




2. In **[Device configuration]** window, check the Enable Slave, add a modbusTCP device to the teach pendant, the IP is 127.0.0.1, the port is 502, and the station number is FF. Add modbusIO according to the address table.




3. The external modbusTCP client also refers to the address table to set modbusIO.
4. Test communication.

**Precautions for the use of RTU Slave:**

1. The software realizes the communication between the teach pendant modbusTCP client and the external modbusRTU master station through program conversion.
2. The teach pendant modbusTCP client only has functions of coil and holding register (in order to exchange data with the RTU master station).
3. Important: The first half of the coil/holding register is for sending from the modbusRTU master to the modbusTCP client of the teach

pendant, and the second half is for sending from the modbusTCP client of the teach pendant to the modbusRTU master.

**For example:** The number of coils is 64, and the number of holding registers is 48. Then DO_0~DO_31 can be sent from the modbusRTU master station to the teach pendant modbusTCP client (the teach pendant client can only read but not write), and DO_32~DO_63 can be sent from the teach pendant modbusTCP client to the modbusRTU master station (the RTU master station can only read but not write).

Similarly, RegisterOutput_0~RegisterOutput_23 can be sent from the modbusRTU master station to the teach pendant modbusTCP client (the teach pendant client can only read but not write), and RegisterOutput_24~RegisterOutput_47 can be sent from the teach pendant modbusTCP client to the modbusRTU master station (the RTU master station can only read but not write).