# **QD1 IO Expansion Module**





# **User Manual**

**QD1 IO Expansion Module** 

Modular & Economical

Document Version V3.1.0

Issue Date 18/08/2020

QKM Technology (Dongguan) Co., Ltd.

# Preface

Thank you for purchasing QD1 produced by QKM!

This manual describes the notes for proper use of QD1.

Read this manual carefully before using QD1.

Keep this manual properly for future reference.

#### Overview

This manual provides the basic information, module composition, work principle, installation and configuration guidance of QD1 products for users to fully understand and use them.

#### Readers

This manual applies to:

Customer Engineer	Technical Support Engineer
System Configuration Engineer	Installation and Debugging Engineer

#### Signs and meanings

The signs in this document clearly indicate any dangers, warnings, attentions and descriptions that may occur while users perform the operations described in this manual; be sure to pay attention to the following signs when they appear in this document.

The signs in this manual are described in the table below:

Sign	Description
DANGER	It indicates that a dangerous situation would occur and cause deaths or serious injuries if it is not avoided.
WARNING	It indicates that a potentially dangerous situation would occur and cause personal injuries or modules damage if it is not avoided.
	It indicates that an unpredictable situation would occur and cause module damage, performance degradation, data loss, etc. if it is not avoided.
	It indicates the description of key information and tips of operation skills.

Copyright  $\textcircled{\sc c}$  2020 QKM Technology (Dongguan) Co., Ltd.

#### All Rights Reserved.

QKM Technology (Dongguan) Co., Ltd. (hereinafter referred to as QKM) owns the patents, copyrights and other intellectual property rights of this product and its software. Without the written authorization of QKM, no unit or individual shall extract or copy the contents of this document; and shall not directly or indirectly reproduce, manufacture, process and use this product and related parts.

### Trademark

granted ownership of this trademark.

### Disclaimer

QKM does not assume any direct, indirect, special or incidental loss or liability due to improper use of this manual or product.

#### Notice

The contents of this document are subject to change at any time due to product upgrade or other reasons. QKM reserves the right to change the product or specifications in this manual without prior notice. To know the latest information of this product, please visit QKM's website and download.

Tower A, Building 17, Headquarters 1, No. 4 Xinzhu Road, Songshan Lake High-tech Industrial Development Zone, Dongguan City, Guangdong Province, China

Tel: +86 0769-27231381 Fax: +86 0769-27231381-8053 Zip code: 523808 Email: service@qkmtech.com Website: www.qkmtech.com

# **Version History**

The version history contains the accumulated information on each update of the document, and the latest version of the document includes the updates in all previous versions of the document.

Version	Date	Content
V1.0.0	18/08/2020	The first version was officially released.

# Contents

Preface	I
Version History	
Description of Terms	VIII
Chapter 1 Product Overview	1
1.1 Introduction of components	2
1.1.1 Modular type	2
1.1.2 Economical type	3
1.2 Product dimensions	4
1.2.1 Modular type	4
1.2.2 Economical type	5
1.3 Technical parameters	6
1.3.1 Electrical parameters	6
1.3.2 Dimension parameters	6
1.3.3 Installation environment	7
1.4 Product features	7
Chapter 2 Work Principle	8
2.1 Communication networking	8
2.1.1 Modular type	9
2.1.2 Economical type	9
2.2 Definition of IO pins	10
2.2.1 Modular type	10
2.2.2 Economical type	11
2.3 Digital input (DI)	13
2.3.1 Modular type	13
2.3.2 Economical type	13
2.4 Digital output (DO)	14
2.4.1 Modular type	14
Document Version V3.1.0 (2020-08-18)	V

2.4.2	2 Economical type	14
2.5 Desc	ription of LEDs	15
2.5.7	1 Modular QD1	15
2.5.2	2 Economical QD1	16
Chapter 3	Product Installation	18
3.1 Prepa	aration before installation	18
3.2 Insta	llation of modular QD1	18
3.2.7	1 Installation of guide	18
3.2.2	2 Installation of communication cable	18
3.2.3	3 Installation of power wire	20
3.3 Insta	llation of economical QD1	20
3.3.2	1 Installation of motherboard	20
3.3.2	2 Installation of daughterboards	21
3.3.3	3 Installation of communication cable	22
3.3.4	1 Installation of power wire	24
3.4 Chec	k after installation	25
Chapter 4	Configuration before Use	26
4.1 Initial	lization configuration	26
4.1.1	I IP address configuration	26
4.1.2	2 IP address check	28
4.2 Confi	guration for connection to robot	28
4.2.7	1 Network connection	28
4.2.2	2 Configuration for connection to Web Server	29
4.2.3	3 Web Interface IO control	31
4.2.4	4 Call QD1 instruction in GDE	32
4.3 Confi	guration for connection to PC	32
4.3.2	1 PC configuration	32
4.3.2	2 File load configuration	33

4.4 Dynamic library configuration	47
Appendix A FAQ	49

# **Description of Terms**

The special terms in this manual are described as follows:

Term	Description
QKM	Quotient Kinematics Machine
QD1	QKM Digital Input and Output
GDE	Guidance Development Environment
Ю	Input & output
DI	Digital intput
DO	Digital output
LED	Light Emitting Diode
LAN	Local Area Network
DHCP	Dynamic Host Configuration Protocol

# Chapter 1 Product Overview

QD1 indicates a compact remote digital IO module independently developed by QKM, and can transmit more optocoupler-isolated signals to main controller; it has multiple IO channels to support bus expansion.

QD1 supports Modbus TCP commonly used in the industry. Main controller can access and control each IO channel upon connection to QD1 with network cable. QD1 module has dual RJ45 network ports with native routing and forwarding function, and can be flexibly and freely networked with main controller via cascade communication or daisy chain topology.

QD1 product series includes modular type of 32 In/32 Out, modular type and economical type of 16 In/16 Out, as shown in Figure 1-1, Figure 1-2 and Figure 1-3:



Figure 1-1 Modular QD1 (32 In/32 Out)



Figure 1-2 Modular QD1 (16 In/16 Out)



Figure 1-3 Economical QD1

## 1.1 Introduction of components

#### 1.1.1 Modular type

The components of modular QD1 are detailed in Figure 1-4. Modular QD1 (16 In/16 Out) has the same power interface and network ports as modular QD1 (32 In/32 Out).







Figure 1-5 Back view of modular QD1 (32 In/32 Out)

#### 1.1.2 Economical type

Economical QD1 consists of a motherboard and a daughterboard with components detailed in Figure 1-6 and Figure 1-7:



Figure 1-6 Motherboard of economical QD1



Figure 1-7 Daughterboard of economical QD1

- 1.2 Product dimensions
- 1.2.1 Modular type



Figure 1-8 Dimensions of modular QD1 (32 In/32 Out) (unit: mm)



Figure 1-9 Dimensions of modular QD1 (16 In/16 Out) (unit: mm) 1.2.2 Economical type



Figure 1-10 Dimensions of QD1 motherboard (unit: mm)



Figure 1-11 Dimensions of QD1 daughterboard (unit: mm)

## 1.3 Technical parameters

### 1.3.1 Electrical parameters

Table	1-1	QD1	electrical	parameters
i abio			01000110001	paramotoro

ltem	Modular type	Economical type	
Rated input voltage	24 V d.c. (20.4~28.8 V d.c.)	24 V d.c. (20.4~28.8 V d.c.)	
Interface type	RJ45 (two)	RJ45 (two)	
Interface speed	10/100 M adaptive	10/100 M adaptive	
Input channel	32 Input/16 Input	16 Input (up to 128 Input)	
Output channel	32 Output/16 Output	16 Output (up to 128 Output)	
Input mode	Default: NPN Customized: PNP	Default: NPN Customized: PNP	
Input signal "0"	0~5 V	0~5 V	
Input signal "1"	15~28 V	15~28 V	
Output mode	NPN mode	NPN mode	
Output drive capability	200 mA (continuous)	200 mA (continuous)	

#### 1.3.2 Dimension parameters

#### Table 1-2 Dimension parameters of QD1 product (unit: mm)

Item	Modular type (32 In/32 Out)	Modular type (16 In/16 Out)	Economical type
Dimensions	110(W)*260(L)*140(H)	110(W)*180(L)*140(H)	85(W)*83(L)*30(H)

#### 1.3.3 Installation environment

Install the product in an environment that meets the following conditions to exploit /maintain its performance and use it safely.

Installation environment	Remark	
Working temperature	0°C ~ +55°C	
Storage temperature	-40°C ~ -70°C	
Maximum relative humidity	< 95% (without condensation)	
Altitude	< 2,000 m	
Atmospheric pressure	800 kPa ~ 110 kPa	
Operating environment	The equipment is suitable for use in environments at pollution level 2 or below, with only non-conductive pollution under normal circumstances and occasionally temporary conductivity caused by condensation.	

Table 1-3 QD1	working	environment	parameters
---------------	---------	-------------	------------

### 1.4 Product features

- Multibus supports Modbus TCP and multiple Ethernet bus protocols after upgrade;
- Flexible expansion supports multi-slave-station cascading topology;
- Stable and reliable with IO channels completely isolated from the system, strong resistance to interference, power protection and IO LEDs;
- High integration small size with high integration.

# Chapter 2 Work Principle

QD1 products communicate as slave stations with master stations (such as computers, controllers, etc.) via Modbus TCP. You can load library functions and header files from folder into program using Microsoft Visual Studio and call related functions to read and write QD1.

QD1 supports two data types: DI and DO.

DI is a discrete input, in which each address corresponds to one data bit, and users can only read its status, but cannot modify it.

DO is a coil output, in which each address corresponds to one data bit, and users can set, reset, and read back its status.

#### 2.1 Communication networking

QD1 supports Modbus TCP protocol communication. Main controller can access and control each IO channel upon connection to QD1 with network cable; QD1 module has dual RJ45 ports and native routing & forwarding function for free forwarding at 10/100 M network, and can flexibly communicate with main controller via cascade communication or daisy chain topology. The diagrams of typical networking and free networking are shown in Figure 2-1 and Figure 2-2, respectively:

## 2.1.1 Modular type



Figure 2-1 Modular QD1 networking diagram

2.1.2 Economical type



Figure 2-2 Economical QD1 networking diagram

## 2.2 Definition of IO pins

#### 2.2.1 Modular type



Figure 2-3 Modular QD1 interfaces of 32 In/32 Out (top view)



Figure 2-4 Modular QD1 interfaces of 16 In/16 Out (top view)

Interface	Module	Signal		Function	
DI interface	32 In/32 Out	DI 00~32	24 V	Digital signal input expansion 01~32	24 V power
	16 ln/16 Out	DI 00~16	GND	Digital signal input expansion 01~16	Common port
DO interface	32 In/32 Out	DO 00~32	24 V	Digital signal output expansion 01~32	24 V power
	16 In/16 Out	DO 00~16	GND	Digital signal output expansion 01~16	Common port

#### Table 2-1 Description of modular QD1 interfaces

# 2.2.2 Economical type



Figure 2-5 IO pin interfaces

PIN	Signal	Function	PIN	Signal	Function
1	DI_14	Input 14	19	DI_06	Input 06
2	DI_15	Input 15	20	DI_04	Input 04
3	DI_12	Input 12	21	DI_07	Input 07
4	DI_13	Input 13	22	DI_05	Input 05
5	DI_02	Input 02	23	DI_08	Input 08
6	DI_03	Input 03	24	DI_09	Input 09
7	DI_00	Input 00	25	DI_10	Input 10
8	DI_01	Input 01	26	DI_11	Input 11
9	DO_00	Output 00	27	DO_07	Output 07
10	DO_01	Output 01	28	DO_06	Output 06
11	DO_02	Output 02	29	DO_05	Output 05
12	DO_03	Output 03	30	DO_04	Output 04
13	DO_08	Output 08	31	DO_15	Output 15
14	DO_09	Output 09	32	DO_14	Output 14
15	DO_10	Output 10	33	DO_13	Output 13
16	DO_11	Output 11	34	DO_12	Output 12
17	GND	Power ground	35	+24V	24V power supply
18	GND	Power ground	36	+24V	24V power supply

Table 2-2 Definition of IO pins



Do not connect an external power source to 0 V or 24 V port on the terminal block of economical QD1 to avoid burn out.

# 2.3 Digital input (DI)

#### 2.3.1 Modular type

Modular QD1 supports 32- or 16-channel optocoupler-isolated digital input, and the input circuit diagram is shown in Figure 2-6.



Figure 2-6 Input circuit diagram of modular QD1

## 2.3.2 Economical type

Economical QD1 supports 16-channel optocoupler-isolated digital input, and a single slave station can achieve up to 112-channel optocoupler-isolated digital input via cascaded daughterboards. The input circuit diagram is shown in Figure 2-7.



Figure 2-7 Input circuit diagram of economical QD1

# 2.4 Digital output (DO)

#### 2.4.1 Modular type

Modular QD1 supports 32- or 16-channel optocoupler-isolated digital output. Each channel has a rated output current of 200 mA, and the output circuit diagram is shown in Figure 2-8:





#### 2.4.2 Economical type

Economical QD1 supports 16-channel optocoupler-isolated digital output, and a single slave station can achieve up to 112-channel optocoupler-isolated digital output via

cascaded daughterboards. Each channel has a rated output current of 200 mA, and the output circuit diagram is shown in Figure 2-9.





- 2.5 Description of LEDs
- 2.5.1 Modular QD1



Figure 2-10 Modular QD1 LEDs

LED	Color	Function	Status description
Work LED (LED1)	Green	Work LED	"NO" indicates the module works normally.
Communication LED (LAN1/LAN2)	Green Yellow	Communication LED	"NO" indicates that a communication connection has been established. "Flashing" indicates communication is being performed (100 Mbit/s). "NO" indicates that a communication connection has been established. "Flashing" indicates communication is being performed (100 Mbit/s).

#### Table 2-3 Description of modular QD1 LEDs

## 2.5.2 Economical QD1



Figure 2-11 LEDs on QD1 motherboard

Table 2	-4 Descri	ption of	QD1 L	EDs
		p	~	

LED	Color	Function	Status description
LED1	Green	Work LED	"NO" indicates the module works normally.
LED2	Yellow	Power LED	"NO" indicates 3.3 V power supply works normally.
LED3	Red	Fault LED	"NO" indicates the module fails.
LED4	Red	Network LED	"NO" indicates there is a conflict or serious error in IP address. "Flashing" indicates network connection timeout.
Communication port LAN1/LAN2	Green Communication		"NO" indicates that a communication connection has been established. "Flashing" indicates communication is being performed (100 Mbit/s).
	Yellow	Communication LED	"NO" indicates that a communication connection has been established. "Flashing" indicates communication is being performed (10 Mbit/s).

# Chapter 3 Product Installation

## 3.1 Preparation before installation

Tools: anti-static gloves, anti-static wrist strap, screwdrivers.

- Do not touch the QD1 circuit board directly with your hands to prevent electrostatic damage.
- Be sure to wear an anti-static wrist strap to perform installation, and ensure that no conductors or other conductive objects fall into PCB during QD1 installation.

## 3.2 Installation of modular QD1

3.2.1 Installation of guide rail

WARNING

- Step 1 Install the fixed buckle at one end of QD1 onto the guide rail.
- Step 2 Press down on the other end of QD1 to fix it to the guide rail, as shown in Figure 3-1:



Figure 3-1 Modular QD1 installation diagram

#### 3.2.2 Installation of communication cable

Step 1 Installation of network cable

1. Select a communication cable suitable for working scenario according to

Section 2.1.

2. Insert the RJ45 cable into the communication port of QD1, as shown in Figure 3-2:



Figure 3-2 Installation of modular QD1 network cable



Install the network cable according to configuration scenario and with reference to Section 2.1 Communication networking. This diagram only presents one example scenario for reference.

#### Step 2 Installation of IO signal wires

- 1. Use a screwdriver to press down on the signal interface and hold it.
- 2. Connect the signal wires of a third-party device to the input and output interfaces of QD1.
- 3. Connect other input and output signal wires in the same way, as shown in Figure 3-3:



Figure 3-3 Installation of modular QD1 IO signal wires

- 3.2.3 Installation of power wire
  - Step 1 Use a screwdriver to press down on the power interface and hold it.
  - Step 2 Connect the DC power wire to the power interface of QD1.
  - **Step 3** Release the power interface and check whether the connection is firm, as shown in Figure 3-4:



Figure 3-4 Installation of modular QD1 power wire

## 3.3 Installation of economical QD1

- 3.3.1 Installation of motherboard
  - **Step 1** Install screws and copper posts onto the motherboard as shown in Figure 3-5;



Step 2 Arrange space required for the installation according to installation size.

Figure 3-5 Motherboard installation diagram (unit: mm)

#### 3.3.2 Installation of daughterboards

$\triangleleft$	
$\equiv$	NOTE

The single board of economical QD1 has 16 (input) \* 16 (output) channels. One QD1 motherboard supports installation of up to 7 daughterboards. Upon expansion, it supports 32, 48, 64... 112 IO channels, which can flexibly meet users' requirements.

- Step 1 Install screws and bolts in the mounting holes at the four corners of daughterboard;
- **Step 2** Tighten nuts to fix the daughterboard to the motherboard, as shown in Figure 3-6:



Figure 3-6 Daughterboard installation diagram

Step 3 To install more daughterboards, you need to install a new daughterboard onto the previous fixed daughterboard. QD1 supports extended installation of up to 6 daughterboards, as shown in Figure 3-7:



Figure 3-7 Extended installation of daughterboards



Single board (motherboard) has a maximum output current of 3.2 A (200 mA per channel, 16 channels in all); upon expansion, installed daughterboards need to be independently powered (with a total output current of 3.2 A for 16 channels per daughterboard, and the motherboard does not provide this output) via SCSI interface; up to 6 daughterboards can be installed and all must be powered independently.

- 3.3.3 Installation of communication cable
  - Step 1 Installation of network cable
    - **1.** Install a communication cable according to Section 2.1 Communication networking.
    - **2.** Insert the communication cable into the RJ45 port of QD1, as shown in Figure 3-8:



Figure 3-8 Installation of network cable



Install the network cable according to configuration scenario and with reference to Section 2.1 Communication networking. This diagram only presents one example scenario for reference.

Step 2 Installation of IO signal wires

- 1. (Optional step) View the definition of pins in Section 2.2.
- **2.** Produce an IO connector, as shown in Figure 3-9:



Figure 3-9 Production of IO connector

- If the QD1 product you choose to buy is not equipped with a standard IO plug, you can produce an IO plug by welding in this way.
- When connecting IO signal wires to the connector by welding, the connector should match the motherboard/daughterboard of economical QD1 symmetrically.
- 3. Insert the IO connector into the IO interface, as shown in Figure 3-10:

NOTE



Figure 3-10 Installation of IO signal wires

#### 3.3.4 Installation of power wire

WARNING

- When connecting power supply to QD1, be sure to follow the provisions on 24 V d.c. power connection. Do not connect the positive and negative poles in reverse, otherwise circuit will not work or even circuit board will be burnt;
- If many loads are connected or power consumption is large, external power supply can be separately provided for loads, but common GND terminals need to be connected.
- Step 1 Use a screwdriver to press down on the power interface and hold it.
- Step 2 Connect 24 V d.c. power wire to the power interface of QD1.
- **Step 3** Release the power interface and check whether the connection is firm, as shown in Figure 3-11:





## 3.4 Check after installation

Check whether QD1 is installed correctly before configuration for normal use. To ensure safe and stable operation, please perform the following checks:

- Check whether all fixing screws are tightened.
- Check whether power wire is connected properly and firmly, and whether its positive and negative poles are connected in reverse. Conduct measurement to ensure there is no short circuit before turning on the power.
- Check whether signal wires are connected properly and firmly.
- Check whether foreign objects fall on the circuit board.
- Check whether all wires and cables are tied reasonably and neatly, and whether there are interferences around them to ensure they are not excessively twisted.
- Check whether cable labels are correct. Ensure they are not dropped or damaged, and are in the same direction for easy viewing.
- Check whether power LED and signal LEDs at network port are lit.

# Chapter 4 Configuration before Use

**Prerequisites:** Make sure that GDE, VS2005 or VS2008 and IPconfig.exe have been installed on your computer before using QD1.



IPconfig.exe is a program for IP address configuration and can be downloaded from the official website of QKM.

### 4.1 Initialization configuration

- 4.1.1 IP address configuration
  - Step 1 Connect QD1 to 24 V d.c. power supply. If the power LED is always on, it indicates that power is normally supplied to the module.
  - Step 2 Connect PC to QD1 with network cable, and run the IPconfig.exe on the PC.

名称	*	
🚳 hicp.dll		
IPconfig		
📟 unins000		
🚺 unins000		
🖂 unins000		

Figure 4-1 Run the IPconfig.exe

**Step 3** Click the <Scan> button in the lower right corner of the interface and you will see the information shown in the figure below; query the default IP address (0.0.0.0) of the module.







If you cannot view the IP address of the module after clicking the <Scan> button, click the <Settings> button to make configuration changes as shown below, and click the <OK> button. Then click the <Scan> button again.

Settings
Network Interface
Broadcast from a Specific Network Interface Controller
Network Interface Controller
Realtek PCIe GBE Family Controller
Internal DHCP server Warning! Internal DHCP server should only be used to recover modules
Enable internal DHCP server
OK Cancel

Figure 4-3 Scan IP address

**Step 4** Double-click the address under IP address bar to set a required IP address and its subnet mask; you should turn off DHCP to fix the IP address of the module, as shown in Figure 4-4:

Configure: 00-	30-11-0E-4F-BE	<b></b>
Ethernet configura	tion	DHCP
Subnet mask:	255 . 255 . 255 . 0	C On
Default gateway:	0.0.0.0	
Primary DNS:	0.0.0.0	
Secondary DNS:	0.0.0.0	
Hostname:		
Password:		Change password
New password:		
		Set Cancel

Figure 4-4 Turn off DHCP

Step 5 Click the <Set> button and a new IP address of the module is set successfully. After power-off and restart, you can use the new IP address to connect the module.

#### 4.1.2 IP address check

After power-on again, repeat the steps in Section 4.1.1 for IP address configuration. If the IP address is the address set by the user, it means that the setting is successful; if it fails, check whether the set IP address is occupied by another device.

#### 4.2 Configuration for connection to robot

**Prerequisites:** GDE installed on computer; personnel familiar with operations on GDE.

#### 4.2.1 Network connection

QD1 can be quickly configured to connect with QKM robot controller. Connect QD1 to LAN where robot controller is located with network cable, which can be achieved via industrial switches, routers, etc.



Figure 4-5 Installation of economical QD1 network cable

#### 4.2.2 Configuration for connection to Web Server

**Step 1** Open webpage with IE browser. Enter the IP address (e.g. 192.168.01) of robot or controller in the address bar. If network connection is normal, click to enter the interface as shown in Figure 4-6.

$\leftarrow$ $\rightarrow$ C $\bigcirc$ 192.	168.01	U		<b>e</b> z :
	ISE System	1: MotorBox-00	)003 no-r&c	d-no-access
Welcome to	o the Precise A	Automation "G Viewer"	uidance Contr	oller Web
You are con Controller s Controller Software V	nnecting to name: serial #: 'ersion:	MotorBox-00003 0014FF-02302020 GPL 4.1A2, Sep 3 2	no-r&d-no-access 2015, Beta Release	
Select Access Level: Application	Operator	Maintenance	Admin	Readme 3
Enter pass	word:			
To login to	the system, please <b>e</b>	enter your password	l and select the	

Figure 4-6 Open Web Server

Step 2 Open <Administrator>, <Control Panel> and <Teaching Panel> in sequence. Choose Robot as needed. <Robot1> is default choice on the page (If multiple robots are in the device and controlled in master-slave mode, the master host Robot1 is preferred).

PRECISE AUTOMATION	System: Mo	torBox-	·00003 -	no-r&	d-no-ac	CESS Disable Power
Control Panels Setup Moti Update Auto No Auto Control Panels Operator Control Panel Virtual Pendant Web Dialog Virtual Robots Communication B Local I/O B Remote I/O B System Information	Position: Cartesian X -0.036 Joint 1t 1/7 -0.036 Joint System sta Robot St System sta Robot hom Jog Cont	Application Application WtThm Y 0.036 Jt 2/8 0.036 atus atus ate: Off: rol er Wor t Axis/Joi	Fixture Z -0.036 Jt 3/9 0.036 Switch on w No	NO-F& ut Help 0100000 yaw 0.000 Jt 4/10 0.036 vait Enu Ho bool J Jc	0 Hide Joint pitch 180.000 Jt 5/11 able me	Show Tool roll -179.964 Jt 6/12 Robot 1 Robot 1 Robot 2 Robot 3
	Select	Inc Inc	log Speed Inc Inc 10 100 10	10 20 40 60	+ - % 59t 80 90 100	+

Figure 4-7 Select Robot

Step 3 Open <System Settings>, <System Parameter Database>, <Controller>, <Modbus (TCP)> and <Node 1> in sequence. Find Data ID 583 on the <Modbus Introduction> page and configure Web Server IP N according to the configured maximum number of extended IO channels for QD1. The configuration diagram and rules are as follows:

PRECISE AUTOMATION System: MotorBox-00003 no-r&d-no-access						
Select Robot Robot 1 •		GPL allows for remote I/O to	Modbus Details r up to 4 Modbus nodes to be configured. This permits 3rd party be used with the Guidance Controller. The table below shows both	•		
Wizards and Setup Tools     Hardware Tuning and Diagnost		the setup and	i status of ModBus TCP.			
- Parameter Database - Controller - System ID		Parameter name Green = restart	Robot: 1 , MtThmFixture 0100000			
Operating mode     Debug and trace     Serial Ports	ID	required Red = high power must be off	Parameter value			
Network     System DIO	580	MODBUS/TCP scanner time in sec	0.016			
ModBus TCP  Node 1 Modbus Tcp Node 2 Modbus Tcp	581	MODBUS/TCP scanner timeout in sec	0.1			
Node 3 Modbus Tcp	582 583	MODBUS/TCP scanner mode MODBUS/TCP	1 192.168.15.20,1:B,16,10101,1:B,16,101			
EtherNet/IP Task control Data logger Misc	584	Scanner map MODBUS/TCP scanner connect count	1			
GPL program variables ⊕ Robot: MtThmFixture 0100	585	MODBUS/TCP scanner status	0			
		Ca	ancel changes Set new values Save All to Flash	÷		

Figure 4-8 Configure Web Server IP N

- Change the value of Data ID 582 to 0 before every change of Data ID 583 value, otherwise the Data ID 583 value that is already in effect cannot be modified.
- When the number of IO channels extended for QD1 exceeds 64, repeat Step 2~Step 3 to configure Data ID 583 in Node2 Modbus TCP.
- If configured for single motherboard of QD1, the QD1 motherboard supports a maximum of 16 IO channels, then modify its value to "192.168.2.247, 1:B, 16, 10101, 1:B, 16, 101".
- If configured for one motherboard plus one daughterboard of QD1, QD1 supports a maximum of 32 IO channels, then modify its value to "192.168.2.247, 1:B, 32,10101,1:B,32,101".
- If configured for one motherboard plus multiple daughterboards of QD1, when

NOTICE

the number of IO channels extended for QD1 exceeds 64, repeat Step 2~Step 3 to configure Data ID 583 in Node2 Modbus TCP (e.g. 6 daughterboards are installed with a maximum of 112 IO channels, and "192.168.2.247, 1:B, 64, 10201, 1:B, 64, 201" is entered in Date ID 583 of Node2 Modbus TCP).

$ \begin{array}{cccccccccccccccccccccccccccccccccccc$						
Ф	IP address configured for QD1 motherboard					
2	The first input channel at the beginning of Input, with an initial value of 1					
3	The last input channel at the end of Input, with a maximum value of 64					
Ø	The first input channel mapped to controller, with an initial value of I 10101					
9	The first input channel at the beginning of Output, with an initial value of 1					
6	The last input channel at the end of Output, with a maximum value of 64					
Ø	The first output channel mapped to controller, with an initial value of O 101					
_						

Table 4-1 Parametric rules	of Web Server IP
----------------------------	------------------

**Step 4** Click <Set new values> to save the settings.

- **Step 5** Change the value of Data ID 582 in Figure 4-8 to 1. ("1" means effective, "0" means ineffective).
- **Step 6** Click <Set new values> again to save the settings.
- **Step 7** Finally, click <Save All to Flash> to save the values permanently.

#### 4.2.3 Web Interface IO control

After configuring connection to Web Server, open <Control Panel>, <Remote I/O> and <RIO 1> in sequence, and the IO control of QD1 is displayed on the IO control panel of Web Interface.

AUTOMATION System: MotorBox-00003 no-r&d-no-access									
Control Panels Setup Motion	Control Panels Setup MotionBlocks Utilities Application Web Logout Help								
Update Auto No Auto Control Panels Operator Control Panel Virtual Pendant Web Dialog	Remote I/O status panel (node 1) There are 64 remote inputs and 64 remote outputs available. To manually force I/O on or off, click on the icon for the address. This will open the Force control panel.	*							
Virtual Robots     Tropping	Digital Inputs Digital Outputs								
B Local I/O Remote I/O	110101         110102         110103         110104         0101         0102         0103         0104           ↓ ⊢								
<mark>RIO 1 Status</mark> RIO 2 Status	I10105         I10106         I10107         I10108         O105         O106         O107         O108           ⊣⊢         ⊣⊢         ⊣⊢         ⊣⊢         -⊖         -⊖         -⊖         -⊖         -⊖								
RIO 3 Status RIO 4 Status	$\begin{array}{cccccccccccccccccccccccccccccccccccc$								
Network Node 2 I/O	I10113 I10114 I10115 I10116 0113 0114 0115 0116	-							

Figure 4-9 Web Interface

#### 4.2.4 Call QD1 instruction in GDE

- **Step 1** Call the IO control instruction <Signal.DIO(XXX) =True> or <Signal.DIO(XXX) = False> in GDE.
  - Example: When Signal.DIO (101) =True, set Output 101 to 1;
  - Example: When Signal.DIO (102) = False, set Output 102 to 0;

**Step 2** Set other output ports in sequence according to the examples above.

## 4.3 Configuration for connection to PC

**Prerequisites:** Visual Studio 2015 installed on computer; personnel familiar with Visual Studio development environment.

#### 4.3.1 PC configuration

QD1 can be controlled via multiple terminals. Users can control QD1 via PC/host computer in addition to QKM robot. Install Visual Studio 2015 before configuring PC. The requirements for hardware installation are shown in Table 4-2 Requirements for hardware configuration:

Hardware	Minimum requirement
Processor	1.6 GHz
Memory	384 MB

#### Table 4-2 Requirements for hardware configuration

#### 4.3.2 File load configuration

Files to be configured include:

DII — modbus.dll: dynamic link library of Modbus protocol.

DII — ws2\_32.dll: application program interface of Windows Sockets.

example: example program.

include: header files that program needs to include.

library: library functions of Modbus protocol.



The following provide instructions for use based on win10/64-bit operating system, taking the creation of a console application as an example.

Step 1 Connect PC to QD1 with network cable, as shown in Figure 4-10:



Figure 4-10 Connect PC to QD1

Step 2 Open Visual Studio 2015 on your computer via "Blend.exe" from files location.

名称 ^	修改日期	类型					
WindowsPhone	2020/8/7 16:32	人任失					
📕 Xml	2020/8/7 14:28	文件夹					
📕 zh-CHS	2020/8/7 17:25	文件夹					
📕 zh-CN	2020/8/7 17:32	文件夹					
📕 zh-Hans	2020/8/7 17:37	文件夹					
📕 zh-Hant	2020/8/7 15:20	文件夹					
🖾 Blend.Application	2016/5/25 7:19	PKGDEF 文件					
🕍 Blend	2016/6/20 14:45	应用程序					
P Blend.exe	2016/6/20 14:01	XML Configurati					
🕅 Blend	2016/2/28 21:15	PKGDEF 文件					
🖾 Blend	2016/2/28 21:15	PKGUNDEF 文件					
BlendDesc.dll	2016/6/20 14:45	应用程序扩展					
🗟 BlendDesc-Ln.dll	2016/6/20 14:45	应用程序扩展					
🗟 BlendMnu.dll	2016/6/20 14:45	应用程序扩展					
L PuildNatificationApp	2016/0/10 12-5/	心田招序					
Figure 4-11 Open Visual Studio							

Æ	
=	NOTE

It may not open Win32 Console Application based on Win10 system, if you open Visual Studio 2015 directly via the icon. Please follow the step to run VS 2015.

**Step 3** Create a new project.

**1.** Click <file> and create a new project as shown in Figure 4-12.

×	Blend.exe.cor	fig - Micros	soft Visual	Studio								
File	Edit View	Project	Debug	Team	XML	То	ols	Test	Analyze	Window	Help	
	New				•	*	Proje	ect			Ctrl+Shift+N	
	Open				+	*⊕	Web	Site			Shift+Alt+N	
	Add				×	*	Tean	n Proje	ct			
	Close					*	Repo	ository.				
×	Close Solution					*כ	File				Ctrl+N	.crosoft.Build.N
	Save Blend.exe	.config		Ctrl+S			Proje	ect Froi	m Existing (	Code		awing, version:
	Save Blend.exe	.config As				="tru TFram	ie"≻ iework,	Versio	n=v4.5" />			_
	Advanced Save	Options										
- <b>1</b>	Save All			Ctrl+Shi	ft+S	1>						
	Export Templat	te				1>						
	Page Setup					true"	/>					
-	Print			Ctrl+P								
	Account Settin	gs				led=" re in	'true" hthe f	/> form of	'key1=true	false;key2	=true false>	
	Recent Files				×	ngFra	mesInI	Cons=f	alse;>	ntionIfDicn	sedCancellation	okenSounce-false:
	Recent Project	s and Soluti	ons		Þ	rosof	t-com:	asm.v1	">	peronribisp	JSEGCANCEIIACION	okensource-raise;
x	Exit			Alt+F4		ies;P	rivate	eAssemb:	lies;Common	Extensions\/	Microsoft∖Templat	eProviders;Commor
	24 =	<dependen< th=""><th>mblyIdenti</th><th>&gt; ity name='</th><th>System</th><th>Threa</th><th>ding T</th><th>Tasks D</th><th>ataflow" nu</th><th>hlicKevToke</th><th>n="b03f5f7f11d50;</th><th>3a" culture="neut</th></dependen<>	mblyIdenti	> ity name='	System	Threa	ding T	Tasks D	ataflow" nu	hlicKevToke	n="b03f5f7f11d50;	3a" culture="neut
	26	 bind	lingRedired	ct oldVers	ion="4	.0.0.0	-4.5.6	55535.6	5535" newVe	rsion="4.5.	24.0"/>	
	27	<th>ntAssembly</th> <th>y&gt;</th> <th></th> <th></th> <th></th> <th></th> <th></th> <th></th> <th></th> <th></th>	ntAssembly	y>								
	28 -	<aependen (acce</aependen 	mblvIdenti	> ity name-'	Micros	oft vi	cual St	tudio C	omnonentMod	elHost" publ	licKeyToken-"h03	5f7f11d50a3a" cul
	30	  disse	lingRedired	ct oldVers	ion="2	.0.0.0	-12.0.	.0.0" n	ewVersion="	14.0.0.0"/>		
	31	<th>ntAssembly</th> <th>y&gt;</th> <th></th> <th></th> <th></th> <th></th> <th></th> <th></th> <th></th> <th></th>	ntAssembly	y>								
	<b>20</b>	* d = d = -	4 A = 4	•								

#### Figure 4-12 Open Visual Studio

2. Click <Visual C++> in the New Project page and select Win32 Console Application, then modify the project name and click <OK>, as shown in Figure 4-13.

New Project								?	$\times$
▷ Recent		.NET Fr	amework 4.5.2 - Sort by: Def	ault	• #* <b>E</b>	Search Instal	led Templates (C	trl+E)	<b>ب</b> م
<ul> <li>Installed</li> </ul>			Win32 Console Application		Visual C++	Type: Visu	al C++		
<ul> <li>✓ Templates</li> <li>▶ Visual C#</li> <li>▶ Visual Basic</li> </ul>			Win32 Project	]	Visual C++	A project for application	or creating a Win	32 conso	le
Visual F# Visual C++	1		Empty Project		Visual C++				
Windows MFC Cross Platform		CO.	Install Visual C++ 2015 Tools for	Visual C++					
		Install Windows XP support for C++			Visual C++				
Extensibility SQL Server Python ▷ JavaScript			Makefile Project		Visual C++				
▷ Online	Ŧ		Click here to go onl	ine and find template	<u>s.</u>				
Name:	ConsoleApplicat	ion1							
Location:	c:\users\00\docu	ments\vis	ual studio 2015\Projects			Browse			
Solution:	Create new solut	ion							
Solution name:	ConsoleApplicat	ion1			Create directory for solution				
					L	Add to Sou	rce Control		
							OK	Cano	:el

Win32 Application Wizard - QDIO-TEST1   ?						
Welcome	to the Win32 Application Wizard					
Overview Application Settings	These are the current project settings: • Console application Click <b>Finish</b> fromany window to accept the current settings. After you create the project, see the project's readme.txt file for information about the presence of the generated.	roject				
	< Previous Next > Finish	Cano	æl			

Figure 4-13 Create Win32 Console Application

# **3.** After clicking <Next>, Select Console Application and uncheck empty project, as shown in Figure 4-14.

Win32 Application Wizard - QDIO-TEST1						
C:\_	Settings					
Overview Application Settings	Application type:       Add common header files for:         Windows application      ATL         ● Console application      MFC         ● DL      Static library         Additional options:      Enpty project         ● Export symbols      Precompiled header         ✓ Segurity De velopment Lifecyde (SDL) checks	Cano	el			
	Figure 4-14 Application Settings					



If select empty project, you might not find the directory of C++ **NOTE** in property pages.

**Step 4** Add paths to include files and library files.

1. Click <Project> and < Properties> in sequence, as shown in Figure 4-15.

ile	Edit	View	Proj	ect Build	Debug	Team	Tools	Test	Analyze	Window	Help
6	O → O <sup>*</sup> ← Cetarget solution									Local Wind	ows Debugge
Server Explorer Toolbox	Console	Applicat oleAppli 1 = 2   3 4 5 5 6 7 = 8 8 9 9 0 2		Add Class Class Wizard Add Resourc Add New Ite Add Existing New Filter Show All File Unload Proje Rescan Solut Add Referen Add Connect	 e Item s ct ion ce ed Service		Ctrl+Shi Ctrl+Shi Shift+Al	ft+X ft+A t+A	≥) ntry po	pint for t	he console
	100 %		₩ ₩	Build Custom Manage Nuc Properties	izations iet Package	25					



#### 2. Select all platforms.

ConsoleApplication4 Property	ages	? ×
Configuration: Active(Debug	Platform: All Platforms Active(Win32) Win32	Configuration Manager
General Debugging VC++ Directories C/C++ Linker Manifest Tool XML Document Genera Browse Information Build Events Custom Build Step Code Analysis	Executable Directories         All Platforms         Reference Directories         Library Directories         Library WinRT Directories         Source Directories         Source Directories         Exclude Directories         Exclude Directories         Path to use when searching for executal	cludePath); <different options=""> <different options=""> \$(WindowsSDK_MetadataPath); \$(VC_SourcePath); \$(VC_IncludePath);\$(WindowsSDK_IncludePath);\$(MSBuild_Executab) \$(VC_IncludePath);\$(WindowsSDK_IncludePath);\$(MSBuild_Executab) ble files while building a VC++ project. Corresponds to environment va</different></different>
		OK Cancel Apply

Figure 4-16 Select platforms

3. Then go back to property pages, click <VC++ Directories>, <include

Directories> and <edit...> in sequence. Add paths to include files. After adding paths to include files, it will be shown in directory page.

ConsoleApplicatio	on4 Property Pages						? ×
Configuration:	Active(Debug)	<ul> <li>✓ Platform:</li> </ul>	All Platforms		$\sim$	Configuration	Manager
Configuration	Properties	✓ General					
General		Executable Direct	tories	<different options=""></different>			
Debuggin	g	Include Directori	es	\$(VC_IncludePath);\$(W	/indowsSDK_Incl	udePath);	<u> </u>
VC++ Dire	ectories	Reference Direct	ories	<edit></edit>			-
▷ C/C++		Library Directorie	2S	<different options=""></different>			
D Linker		Library WinRT Di	rectories	\$(WindowsSDK_Meta	dataPath);		
Manifest I	00I	Source Directorie	2S	\$(VC_SourcePath);			
AIVIL DOCU      Proviso Inf	formation	Exclude Directori	es	\$(VC_IncludePath);\$(W	/indowsSDK_Incl	udePath);\$(MSBu	ild_Executab
D Browse Init D Build Even	te						
Custom Build Even	uld Step						
Code Anal	vsis						
		Include Directories					
		Path to use when sea	rching for include file	s while building a VC++ p	project. Correspo	onds to environm	ient variab
				[	OK	Cancel	Apply

Figure 4-17 Add a path to include files

**4.** Go back to the property pages, click <VC++Directories>, <Library Directories> and <Edit...> in sequence. Add paths to library files and confirm as shown in Figure 4-18

ConsoleApplication4 Property P	ages	? ×
Configuration: Active(Debug	) V Platform: All Platforms	<ul> <li>Configuration Manager</li> </ul>
<ul> <li>Configuration Properties General Debugging</li> <li>VC++ Directories</li> <li>C/C++</li> <li>Linker</li> <li>Manifest Tool</li> <li>XML Document Generat</li> <li>Browse Information</li> <li>Build Events</li> <li>Custom Build Step</li> <li>Code Analysis</li> </ul>	<ul> <li>General         <ul> <li>Executable Directories             <ul> <li>Include Directories</li> <li>Reference Directories</li> <li>Library Directories</li> <li>Source Directories</li> <li>Exclude Directories</li> <li>Exclude Directories</li> <li>Exclude Directories</li> <li>Exclude Directories</li> </ul> </li> </ul> </li> <li>Correstant Source Directories</li> <li>Exclude Directories</li> <li>Exclude Directories</li> <li>Exclude Directories</li> <li>Path to use when searching for library</li> </ul>	<pre>v files while building a VC++ project. Corresponds to environment variabl</pre>
		OK Cancel Apply

Figure 4-18 Add a path to library files

- **Step 5** Add paths to additional include files and set precompiled headers.
  - **1.** Go back to the property pages again, click the directory of <C/C++>, <General>, <Additional Include Directories> and <Edit...> in sequence, select paths to additional include files and confirm, as shown in Figure 4-19.

<ul> <li>▲ Configuration Properties General Debugging VC++ Directories</li> <li>▲ Additional Include Directories Additional #using Directories ↓ Edit&gt; Debug Information Format Common Language RunTime Support Consume Windows Runtime Extension Suppress Startup Banner Ves (/nologo) Warning Level Level3 (/W3) Treat Warnings As Errors Output Files Browse Information Advanced All Options Command Line ↓ Linker Multi-processor Compilation Browse Information Advanced All Options Command Line ↓ Linker ▶ Build Events ↓ Custom Build Step</li> </ul>	Configuration: Active(Debug)	V Platform: All Platforms	~	Configuration Manager
Code Analysis           Additional Include Directories	<ul> <li>Configuration: Active(Debudg)</li> <li>Configuration Properties General Debugging VC++ Directories</li> <li>C/C++</li> <li>General</li> <li>Optimization Preprocessor Code Generation Language Precompiled Headers Output Files Browse Information Advanced All Options Command Line</li> <li>Linker</li> <li>Manifest Tool</li> <li>XML Document Generator</li> <li>Browse Information</li> <li>Build Events</li> <li>Custom Build Step</li> <li>Code Analysis</li> </ul>	Additional Include Directories Additional #using Directories Additional #using Directories Debug Information Format Common Language RunTime Support. Consume Windows Runtime Extension Suppress Startup Banner Warning Level Treat Warning As Errors Warning Version SDL checks Multi-processor Compilation Additional Include Directories Control Include Directories	<edit> Program Database for Edit And Contr Yes (/nologo) Level3 (/W3) No (/WX-) Yes (/sdl)</edit>	nue (/21)

Figure 4-19 Add a path to additional include files

**2.** Click the directory of <C/C++>, <Precompiled Headers> in property pages, select <Not Using Precompiled Headers>, as shown in Figure 4-20.

ConsoleApplication4 Property Pages						?	$\times$
Configuration: Active(Debug)	✓ Platform:	All Platforms		~	Configuratio	on Manag	ger
<ul> <li>Configuration Properties General Debugging VC++ Directories</li> <li>C/C++ General Optimization Preprocessor Code Generation Language</li> <li>Precompiled Headers</li> <li>Output Files Browse Information Advanced All Options Command Line</li> <li>Linker</li> <li>Manifest Tool</li> <li>XML Document Generator</li> <li>Browse Information</li> <li>Build Events</li> <li>Custom Build Step</li> </ul>	Precompiled Hea Precompiled Hea Precompiled Hea	der File Ider Output File	Not Using Precompi stdafk.h \$(IntDir)\$(TargetNam	iled Headers			×
▷ Code Analysis	Precompiled Header Create/Use Precomp	r iled Header : Enable	es creation or use of a prec	compiled heade	r during the buil	d. <mark>(</mark> /Yc	;, <b>/Y</b> u)
				ОК	Cancel	Ap	ply



# **Step 6** Click <Linker>, <General>, <Additional Library Directories>, <Edit> in sequence, select paths to additional library files, as shown in Figure 4-21.

ConsoleApplicati	ion4 Property Pages						?	×
Configuration:	Active(Debug)	<ul> <li>Platform:</li> </ul>	All Platforms		~	Configuratio	on Manag	ger
<ul> <li>✓ Configuratic General Debuggir VC++ Dir</li> <li>▷ C/C++</li> <li>▷ Linker</li> <li>▷ Manifest</li> <li>▷ XML Doc</li> <li>▷ Browse Ir</li> <li>▷ Build Eve</li> <li>▷ Custom B</li> <li>▷ Code Ana</li> </ul>	on Properties ng rectories Tool ument Generator nformation nts Build Step alysis	Output File Show Progress Version Enable Increment. Suppress Startup I Ignore Import Lib Register Output Per-user Redirecti Additional Library Link Library Deper Link Status Prevent DII Bindin Treat Linker Warni Force File Output Create Hot Patcha Specify Section Ad	al Linking Banner rary on PDirectories Indencies Indency Inputs g ng As Errors Ible Image Ittributes irectories erride the environm	\$(OutDir)\$(TargetN Not Set Yes (/INCREMENT. Yes (/NOLOGO) No No No < <edit> No No</edit>	ame)\$(TargetExt) AL)			
					ОК	Cancel	App	ply

Figure 4-21 Add a path to additional library files

# **Step 7** Click <Linker>, <Input>, <Additional Dependencies>, and <Edit...> in sequence. Fill in modbus.lib, click <OK> and <Apply> as shown in Figure 4-22.

ConsoleApplicatio	n4 Property Pages							? ×
Configuration: A	Active(Debug)	~	Platform:	All Platforms			~ Configu	ration Manager
Configuration     General     Debugging     VC++ Direc     C/C++     Linker     General     Input     Manifest     Debugg     System     Optimiz     Embedd     Window     Advance     All Optic     Comman     Manifest Tc     XML Docur     Browse Infc     Build Event     Sutd Event	Properties ctories trile ing ation led IDL is Metadata ed ons ind Line bol ment Generator ormation is ld Step	Additi Ignore Ignore Modul Add M Ember Force Delay Assem	onal Depend All Default I Specific Def e Definition lodule to Ass d Managed F Symbol Refe Loaded DIIs bly Link Reso	incies Libraries fault Libraries File embly tesource File rences	!.lib;uuid.lib;odbc32 <edit></edit>	.lib;odbccp32.l	ib;%(Additiona	IDependencies)
Code Analy	Inf Property Pages	Additiona Specifies	additional ite	cies ms to add to the	e link command line. [i.e. k	ernel32.lib]	Cancel	Apply ? × ration Manager
Configuration:     General     Debugging     VC++ Direc     C/C++     Linker     General     Input     Manifes     Debugg     System     Optimiz     Embedd     Window     Advance     All Opti     Comma     Manifest Te     XML Docur     Browse Infi     Build Event     Code Anali     Code Anali	Properties Properties ctories t File ting ded IDL s Metadata ed oons nd Line ool ment Generator ormation ts ild Step visis	Additi Ignore Ignore Modu Add N Embe Force Delay Assen	e All Default e Specific De le Definition lodule to As d Managed F Symbol Refe Loaded Dils ably Link Rese	All Platforms dencies f Additional De f Additional De valuated value Evaluated value Modbus.lib %(AdditionalI < Inherited value kernel32.lib gdi32.lib valuated value valuated value	kernel32.lib;user32. pendencies e: Dependencies) s: parent or project defaults	lib;gdi32.lib;wi	Cancel	dig32.lib;advapi32
▷ Code Analy	ysis	Addition Specifies	al Dependen additional ite	<b>cies</b> ems to add to th	e link command line. [i.e. k	ærnel32.lib]		2
						ОК	Cancel	Apply



**Step 8** Search sysWOW64 folder on your computer, then copy Modbus.dll to the folder.

$\checkmark$	
=	NOTE

- If you cannot find sysWOW64 folder, please search system32 on your computer, click <Windows> to return to upper level folder and you will find sysWOW64 folder.
- The Modbus.dll file can be found in dll folder from factory.

Step 9 Copy two dll files to the debug folder of the project.

NOTE

The operation of the program depends on some special dlls, which involves the placement of dlls. Generally, there are two debug directories under VS projects. To run or debug the program through VS, you need to place dependent dlls in debug2 and it is ineffective to place them in debug1.

For example, in debug mode, there are two debug directories under a VS project. One is at the same level as sln files (assumed to be debug1), and the other is at the level down from the project folder (assumed to be debug2). Generally, files finally generated will be placed in debug1, and those generated during the operation will be placed in debug2. When directly running an executable file (\*\*.exe) in debug1, you need to place dependent dlls under this directory; then you can write your own program, as shown in Figure 4-23.



Figure 4-23 Programming interface



This library function is compiled based on libmodbus (open source library) function transplantation; For more, visit http://libmodbus.org/.

#### Configuration of key functions:

Step 1 Create a new libmodbus environment. Use the modbus device linked by tcp and apply the modbus\_new\_tcp () function. If the new creation is successful, a struct modbus\_t pointer will be returned. Afterwards the operation on the modbus device is to operate the pointer; if it fails, a null pointer will be returned.

Example:

modbus\_t \*ctx; ctx = modbus\_new\_tcp("192.168.0.1", 502)

Multiple modbus slave devices can be connected to one physical interface. Each modbus slave device has its own independent ID, which is called slave ID and is an integer. Therefore, you need to use the modbus\_set\_slave() function to set slave ID for modbus\_t structure, indicating specific modbus device to be connected. Release a libmodbus environment called modbus\_free() function.

- Step 2 After successful creation, connection can be established using the modbus\_connect() function. Close a connection with modbus\_close() function. Refresh a connection with modbus\_flush() function.
- Step 3 After successful connection, you can call related functions to read and write QD1.

The related functions are as follows:

int modbus\_read\_bits(modbus\_t \*ctx, int addr, int nb, uint8\_t \*dest);

Read single IO/coil (i.e. DO) with function code of 0x01. Store the read value into the array pointed to by the dest pointer. This function returns 0 on success and -1 on failure.

Example:

```
uint8_t bits[MAX_IO] = {0};
int ret, i;
ret=modbus_read_bits(ctx,MODBUS_COIL_ADDR,MODBUS_COI
L_LEN, bits);
if (ret < 0)
{fprintf(stderr, "%s\n", modbus_strerror(errno)); }
else
{printf("BITS COILS: \n"), for (i = 0; i < ret; i++)
{printf("[%d] = %d\n", i, bits[i]); } }
int modbus_read_input_bits(modbus_t *ctx, int addr, int nb, uint8_t
*dest);
```

Read single IO/discrete input (i.e. DI) with function code of 0x02. Store the read value into the array pointed to by the dest pointer. This function returns 0 on success and -1 on failure.

Example:

ret=modbus_read_input_bits(ctx,MODBUS_DISCRETE_ADDR,MO
DBUS_DISCRETE_LEN, bits);
if (ret < 0)
{ fprintf(stderr, "%s\n", modbus_strerror(errno)); }
else
{printf("BITS DISCRETE: \n");
for (i = 0; i < ret; i++)
{printf("[%d] = %d\n", i, bits[i]); }}
int modbus_read_input_registers(modbus_t *ctx, int addr, int nb,
uint16_t *dest);

Read 16-bit IO with function code of 0x04. Starting from addr, read the values of consecutive nb registers, and put the results into the array pointed to by \*dest. This function returns the number of read registers on success and -1 on failure.

uint16_t regs[MAX_IO] = $\{0\}$ ;
ret=modbus_read_input_registers(ctx,MODBUS_INPUT_ADDR,M
ODBUS_INPUT_LEN, regs);
if (ret < 0)
{ fprintf(stderr, "%s\n", modbus_strerror(errno)); }
int modbus_write_bit(modbus_t *ctx, int addr, int status);

Write single IO/coil with function code of 0x05. This function can be used to perform write operations to QD1 slave stations. This function returns 0 on success and -1 on failure.

Example:

```
ret = modbus_write_bit(ctx, 5, TRUE);
if (ret < 0)
{ fprintf(stderr, "%s\n", modbus_strerror(errno)); }
int modbus_write_bits(modbus_t *ctx, int addr, int nb, const uint8_t
*src);</pre>
```

Write multiple coils with function code of 0x0F. Write the array pointed to by src into the nb status bits at the beginning of addr. The function returns 0 on success and -1 on failure.

```
Example:

bits[1] = TRUE;

bits[2] = FALSE;

ret = modbus_write_bits(ctx, 1, 2, &bits[1]); if (ret < 0)

{ fprintf(stderr, "%s\n", modbus_strerror(errno));}

int modbus_write_register(modbus_t *ctx, int addr, int value);
```

Write 16 IOs with function code of 0x06. Registers are addressed from 0, so register numbered 1 is addressed as 0. This function can be used for initialization. This function returns 1 on success and -1 on failure.

Example:

```
ret = modbus_write_register(ctx, 0, 0x1234);
if (ret 0)
{fprintf(stderr, "%s\n", modbus_strerror(errno)); }
```

## 4.4 Dynamic library configuration

QD1 module will consume CPU resources for its long-time idle connection. In order to reduce CPU load, QD1 module adopts an automatic power-saving mechanism. When the master station does not interact with QD1 module within 60S, QD1 will automatically disconnect from Modbus-TCP and enter the low-power mode.

After QD1 enters the low-power mode, if interaction needs to be continued, re-establish a Modbus-TCP connection for the master station; if connection needs to be kept, add "Heartbeat Package" function to ensure the continuity of connection. If developed program includes read and write operations of this function, there is no need to add "heartbeat package" function.



Heartbeat Package: A self-defined command word that periodically informs the other party of its own status between client and server, which is sent at a certain time interval (the time interval required by QD1 is within 60S), similar to heartbeat, so it's called heartbeat package.

You can create a thread for management.

Example:	
	#include <stdio.h></stdio.h>
	#include <string.h></string.h>
	#include <stdlib.h></stdlib.h>
	#include <time.h></time.h>
	#include <winsock2.h></winsock2.h>
	#include <windows.h></windows.h>
	#include <process.h></process.h>
	#include "modbus.h"
	/* This can be any input, just to keep connection alive */
	#define HEART_BEAT_BIT0
	#define HEART_BEAT_RATE30000/* 30s */
	unsignedstdcall HeartBeat(void *pParam) {
	unsigned char dest:
	modbus_t *connection = pParam;
	while (1) {modbus_read_input_bits(connect, HEART_BEAT_BIT, 1, dest);
	Sleep(HEART_BEAT_RATE);
	return;}
	int main(int argc, char **argv) {
	modbus_t *ctx;
	HANDLE hThread;

```
/* Establish the connection over TCP */
ctx = modbus_new_tcp(MODBUS_SERVER_IP,
MODBUS_SERVER_PORT);
/* Create and start the backend thread */
hThread = (HANDLE)_beginthreadex(NULL, 0, &HeartBeat, ctx, 0,
NULL);
/* You can implement your own functions here... */
/* Free the connection and disable the thread */
modbus_close(ctx);
modbus_free(ctx);
CloseHandle(hThread);
return 0;
}
```

When adding the "heartbeat package" function, please pay attention to the following points:



- When you need to enable multithreading for management, ensure the safety of multithreading. When accessing the same channel, use mutexes, locks, barriers, etc. for protection.
  - Use the thread functions related to the platform to ensure correct opening and closing of threads.

# Appendix A FAQ

This manual proposes solutions to some common problems during QD1 operation for better use of QD1 products by users:

#### Q: Why is the yellow LED2 of economical QD1 not lit?

A: Check whether the 24V power supply is properly connected and normally supplies power.

# Q: Why is the read data incorrect or corresponding channel unable to be controlled when reading or controlling channels after normal connection?

A1: Confirm whether the electrical connection of corresponding channel is normal and whether the connection is loose or dropped.

A2: Confirm whether the channel configured for software matches the actual channel of hardware.

#### Q: Why are QD1 products not scanned when using IP Config?

A1: Check whether the 24V power supply is properly connected and normally supplies power.

A2: Check whether the connection of network cable is normal and reliable.

# Q: Why is non-conduction displayed on the IO control panel (Control Panels" $\rightarrow$ "Remote I/O" $\rightarrow$ "RIO 1 Status) after Web Interface configuration?

- A1: Check the configured IP address to ensure that the IP addresses of robot and QD1 are on the same network segment.
- A2: Check whether the value of Date ID 582 in Setup  $\rightarrow$  Parameter Database  $\rightarrow$ Controller  $\rightarrow$  Modbus TCP  $\rightarrow$  Node1 Modbus TCP is 1, and whether the value of Date ID 583 is configured correctly.



## Provide Superior Robot Products and Services to Global Manufacturers

#### QKM Technology(Dongguan)Co.,Ltd

Tower A, Building 17, Headquarters 1, No.4 Xinzhu Road, Songshan Lake High-tech Industrial Development Zone, Dongguan , Guangdong, China

Tel: +86 0769-27231381 Fax: +86 0769-27231381-8053 Zip code: 523808 Email: service@qkmtech.com Website: www.qkmtech.com



Scan QR code to follow QKM