User Manual

UIM2523
Industrial Ethernet TCP/IP - CAN Gateway

UIM2523

UIM2523 Ethernet-CAN Gateway Model
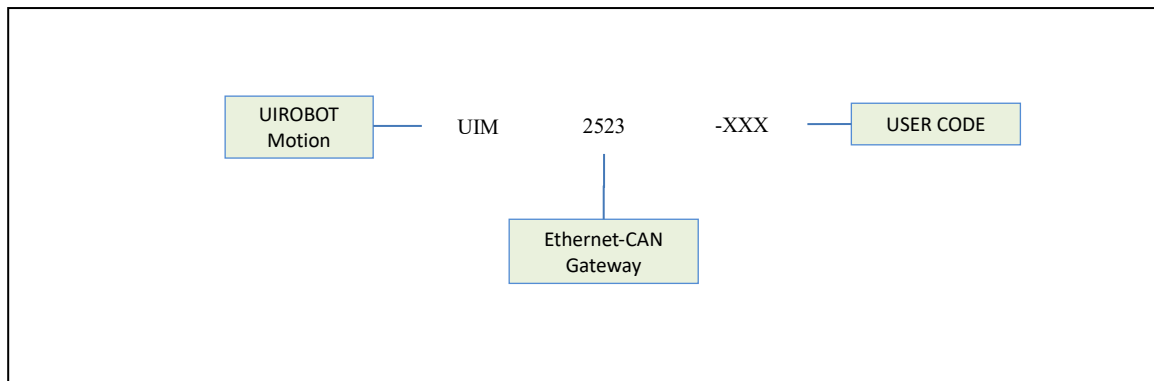
【 Revision History 】

Manual version	Revision date	Change
V1.0	November 1st, 2023	initial version

UIM2523
Industrial Ethernet TCP/IP - CAN gateway

UIM2523 is an industrial Ethernet TCP/IP to CAN gateway. Users can send instructions through UIM2523 to control UIM series servo stepper, AC servo motion controller and I/O controller products.

UIROBOT provides complete SDK, dll, lib, so, and sample codes based on C++, C#, etc., for Linux, Windows 32 bit and 64 bit.

Control System

−   Robust DSP hardware

−   Fault tolerance, fail safe user interface

−   Provide complete SDK, lib and dll

−   Provide C++, C# sample codes

Ethernet communication

−   TCP/IP protocol

−   RJ45 interface

−   Transformer isolation

CAN Networking

−   Active CAN 2.0, 1 Mbps Max.

−   1500V photoelectric isolation

−   Support 3 kinds of CAN IDs: Node ID, Group ID and Global ID

Other

−   Wide voltage input 12~48VDC

   Aluminum alloy casing, sturdy and durable for easy heat dissipation

UIM2523

Contents

Safety

To prevent personal injury and property damage, please be sure to pay attention to the following before use.

	Precautions	Consequences of neglect
⚠	Do not use it in humid, corrosive, flammable gas environments or places near flammable substances	Fire or malfunction
⚠	Do not use the wire when it is soaked in oil/water	Fire or malfunction
⚠	Do not frequently power on / off	Malfunction
⚠	Never modify, disassemble or repair by yourself	Fire, malfunction
⚠	Power supply voltage must meet the product requirements	Malfunction
⚠	Cut off the power when not in use for a long time	Fire , malfunction

Maintenance

Please perform regular maintenance and inspection on the controller for safe use. Please pay attention to the following during maintenance and inspection:

① When performing an insulation resistance test, be sure to disconnect all connections, otherwise malfunction may result.
② Do not use gasoline, thinner, alcohol, acidic and alkaline cleaning agents to avoid discoloration or damage to the casing.

Daily inspections and periodic inspections should be carried out according to the following items .

Type	Period	Check Item
Daily Inspection	Every day	• Confirm the operating temperature and humidity • No foreign matter entering • Abnormal vibration, sound and odor • Abnormal power supply voltage • Damaged wiring parts
Periodic Inspection	1 year	• No looseness in the fastening parts • Broken terminal blocks and loose fastening parts

Hardware

Figure 0-1: UIM2523 wiring diagram
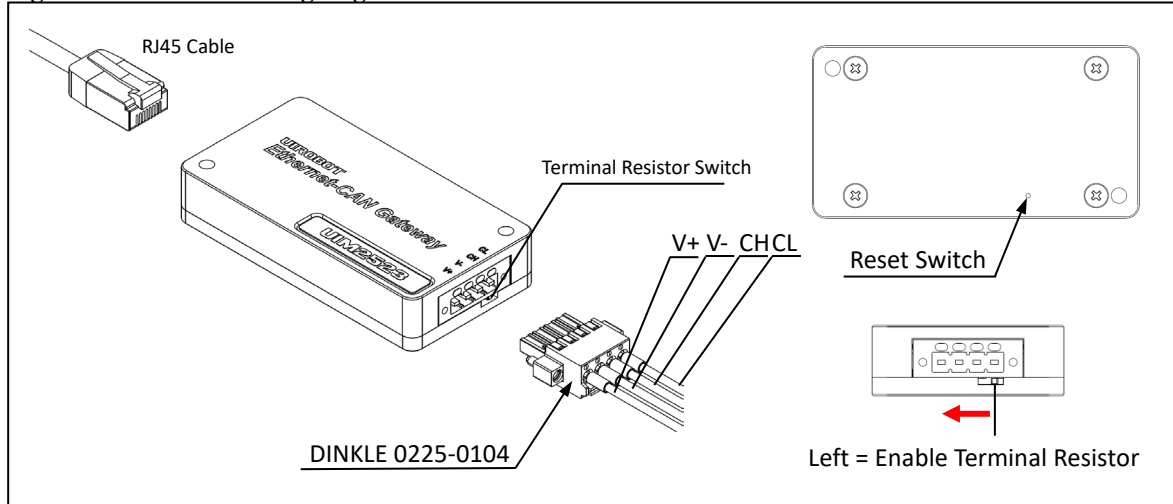
label	Illustrate
V+	Supply voltage positive, 12 - 48V DC (red)
V-	Supply voltage negative pole , 0V (black)
CH	CANH signal wire(white)
CL	CANL signal wire(green)

Quick Start

The following explains how to quickly build and run a motor system consisting of 2 servo stepper motors equipped with UIM342 controllers and 1 UIM2523 gateway.

1）Wiring the system as shown below. Double check the wiring before turn on the power.

Figure 0-2: Quick connection diagram
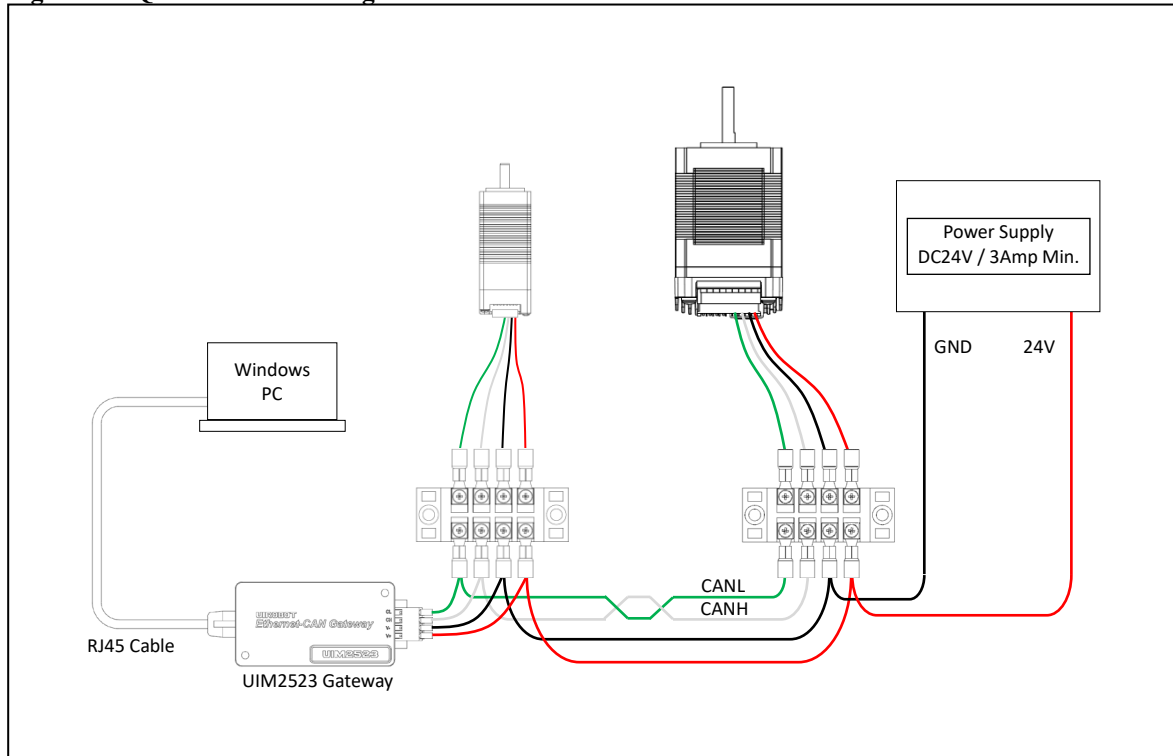
⚠ **WARNNING**

- Avoid using star connections. When the CAN cable length exceeds 20 meters, dedicated CAN cables should be used. The length of branch cable to each node should not exceed 20 cm.

- UIM2523 has a built-in terminal resistor, which can be activated by the toggle switch. It is recommended to connect a 120-ohm terminal resistor to the other end of the CAN cable.

- Strictly avoid hot-plugging while the power is on. Hot-plugging may lead to ground loss (i.e. the power supply V+ is connected while V- is disconnected). In such instances, power V+ will flow into other UIM devices via the CAN cable, causing the burnout of multiple UIM devices.

- Connect all UIM devices to a common ground. Activating a high-power device can raise the voltage on one ground significantly. Without a common ground, this elevated voltage may flow via the CAN cable to other grounds of other devices, risking the burnout of multiple devices.

2）Before connecting, make sure that the IP address of the computer LAN port is in the 192.168.1.xxx network segment, otherwise the gateway cannot be searched.

Figure 0-3: The local IP address of the user's computer is set to the same network segment as the gateway IP address
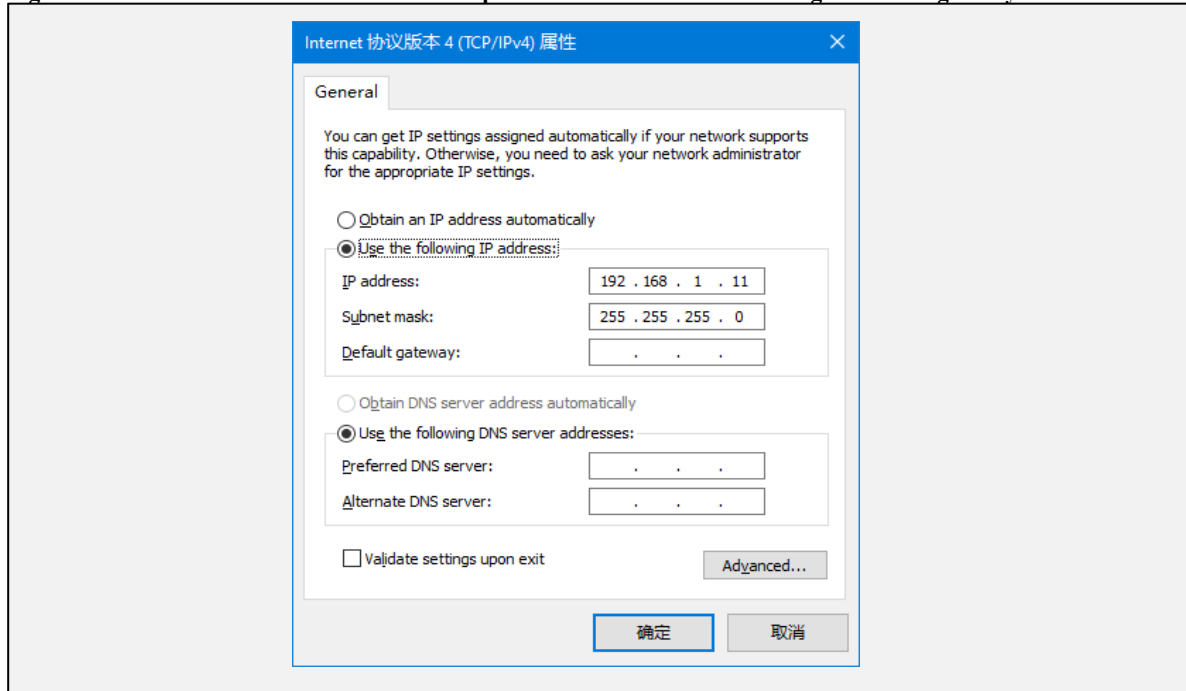
3）Download and click to run the Windows based control panel "StepEva3".

Figure 0-4: Icon of StepEva3
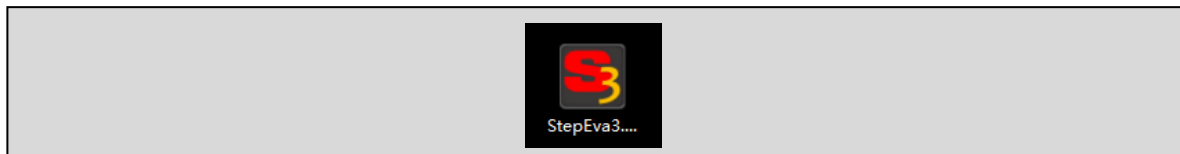
4）Click to select the UIM2523 Gateway.

Figure 0-5: Gateway selection
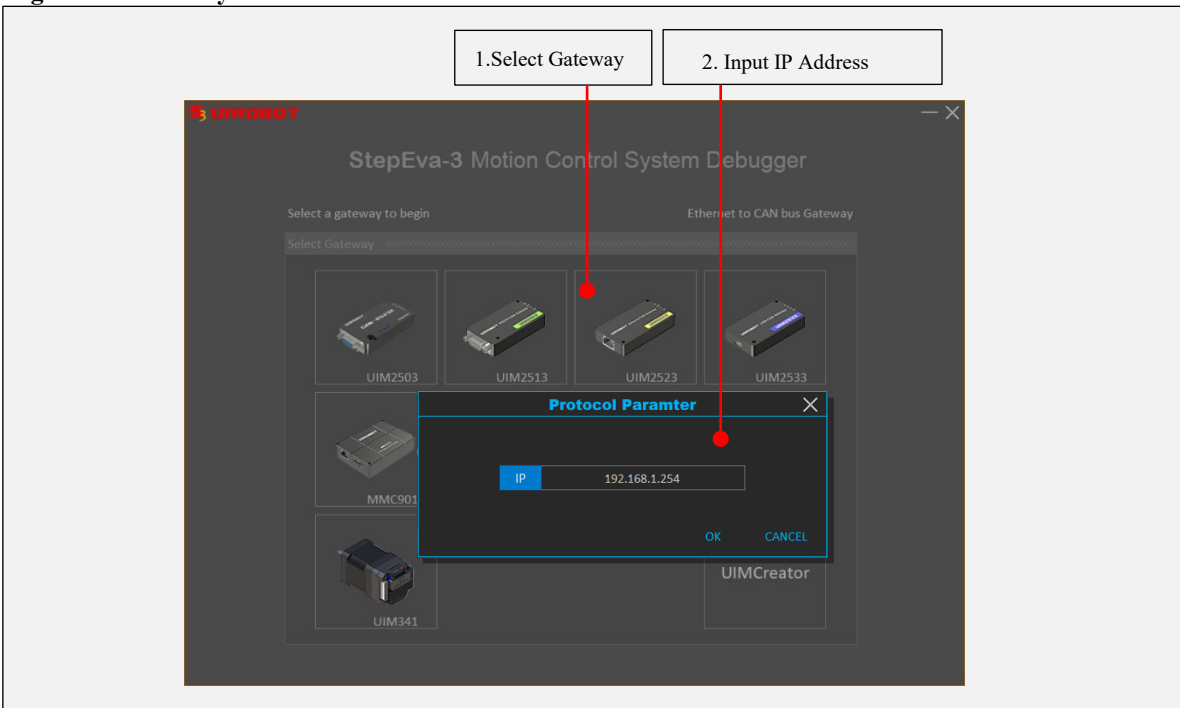
5）Click to select the device to operate from the device list on the left in Figure 0-5.

Figure 0-6: Gateway information
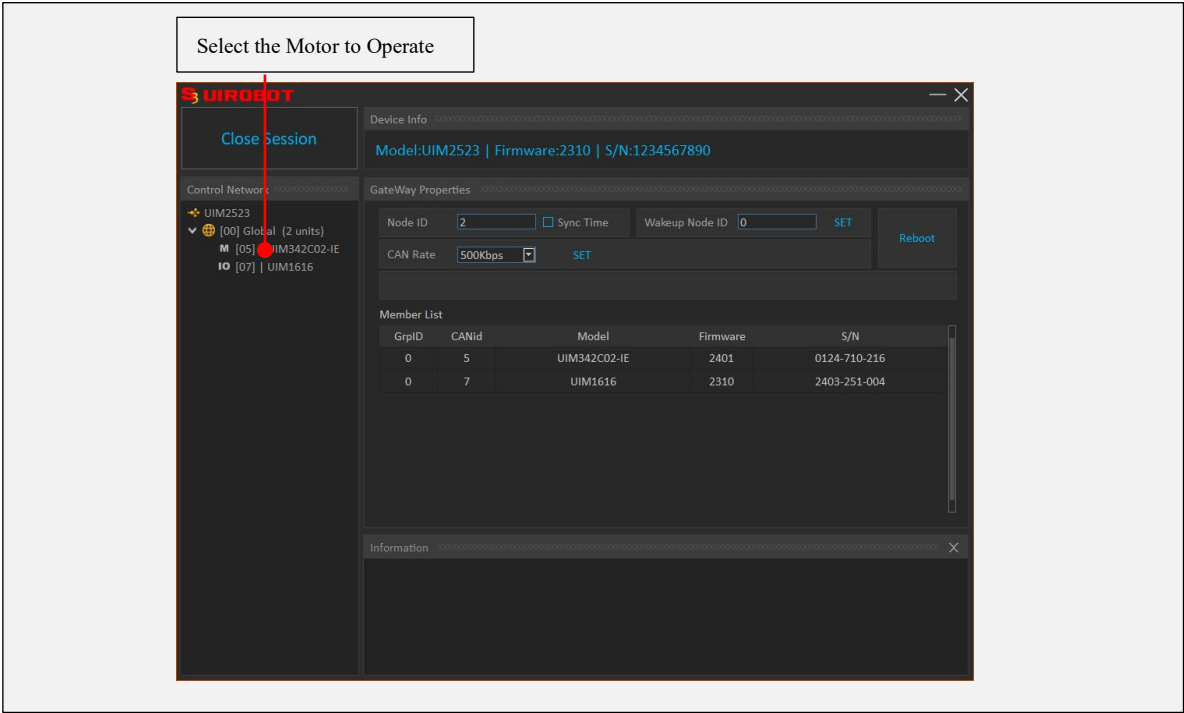
6）Refer to the corresponding User Manual of the selected device to operate.

Specifications

Absolute Maximum Ratings

Ambient temperature under bias..-40°C to 85°C
Storage temperature .... -65°C to +150°C
Voltage on V+ with respect to V- .... -0.3V to 50V

Stresses above those listed under "Absolute Maximum Ratings" may cause permanent damage to the device. This is a stress rating only and functional operation of the device at those or any other conditions above those indicated in the operational listings of this specification is not implied. Exposure to maximum rating conditions for extended periods may affect device reliability.

Operating Conditions (at ambient temperature 25°C)

Parameter	Min	Max	Units
Supply voltage (DC)	12	48	Volt
Input Current	0	100	mA

Operating Environment

Cooling	Free Air
Environment	Avoid dust, oil mist and corrosive gases
Operating temperature	-40 °C ~85 °C
Humidity	<80%RH, no condensation, no frost
Vibration	3GMax
Storage	-65 °C ~ 150 °C

Communication Interface

CAN	Active CAN 2.0
CAN Physical	2 wires, CANH, CANL, twisted pair
CAN Driver	• Max. 1 Mbps • Meets ISO-11898 standard physical layer requirements. • 1500V Isolated • Differential bus
User Interface	TCP/IP
TCP/IP Physical	RJ45

Other

Dimensions	72mm x 38mm x 14mm
weight	0.15 Kg

1.0 Protocol

This section introduces UIM2523's Instruction Set, Reply (ACK), Error Reporting, and Real-time Notification.

1.1 Instruction and Reply (ACK)

UIM2513 and the user controller exchange information using UIMessages, as shown in Figure 1-1. The length of a UIMessage is fixed to 16 bytes.

Instruction is a functional message from user controller to UIM devices.

ACK is the reply message from UIM devices, after receiving an instruction.
For "Set" instructions, the ACK message is typically a repeat of the instruction.
For "Get" instructions, the ACK message provides the data being queried.

Figure 1-1: UIMessage Frame Structure
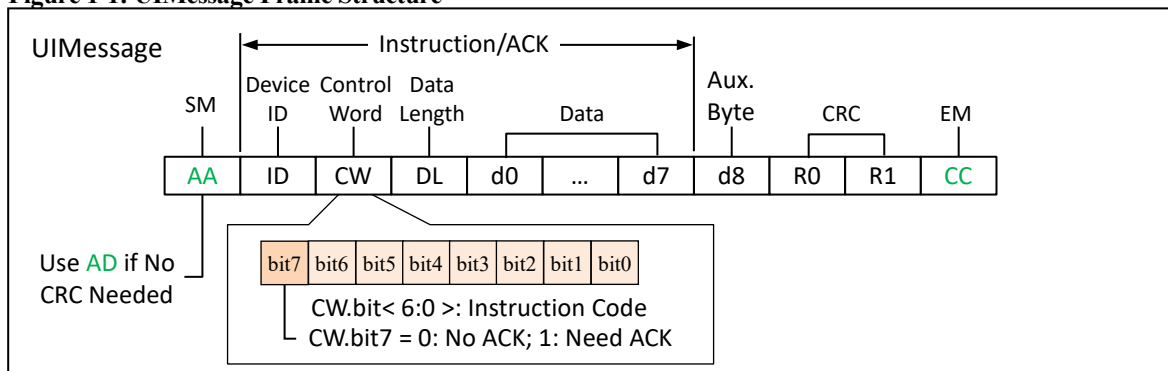
SM   Start of Message. If need CRC, using 0xAA, else using 0xAD instead.

ID   UIM Devices' ID (1 byte), could be one of the 4 types listed below:

1）UIM342's Node ID, configurable via instruction,
2）Group ID of UIM devices, configurable via instruction,
3）Global ID = 0, fixed, or
4）UIM2523's Node ID = 2, fixed.

CW   Control Word (1 byte)

1）CW.bit <6:0> is the function code (0x00…0x7F).
2）For instruction: if CW.bit7=1, UIM will ACK, if CW.bit7=0, UIM will not ACK.
3）For ACK: CW.bit7 = 0.

DL   Valid number of bytes of data (1 byte)

d0…d7 Data bytes, low byte first.

d8   Auxiliary byte, don't care

R1:R0 RTU CRC16, range covers ID…d8, refer to Appendix- 2 for source code.

EM   End of Message; is fixed to 0xCC.

UIM2523

1.2 Mnemonic

For user convenience, UIM products provide a set of mnemonics to represent function codes. For example:

mnemonic	Script code	Function
MO	0x15	Turn on/off motor drive

The relation between mnemonic expressions and instructions is shown below:

Figure 1-2: Mnemonic Expressions & Instruction Messages
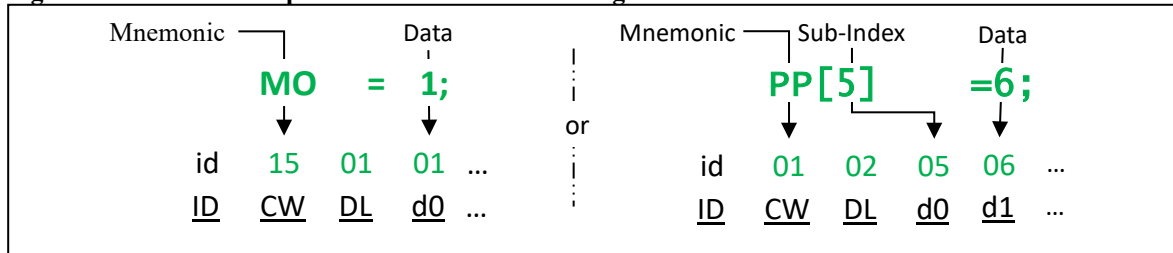
1.3 Error Report

When the UIM device receives a wrong instruction, it will reply an error message instead of executing the instruction. Meanwhile, users can use `ER[i]` to get at most 8 historical errors. The format of an error message is listed below:

Figure 1-3: Error Report Format
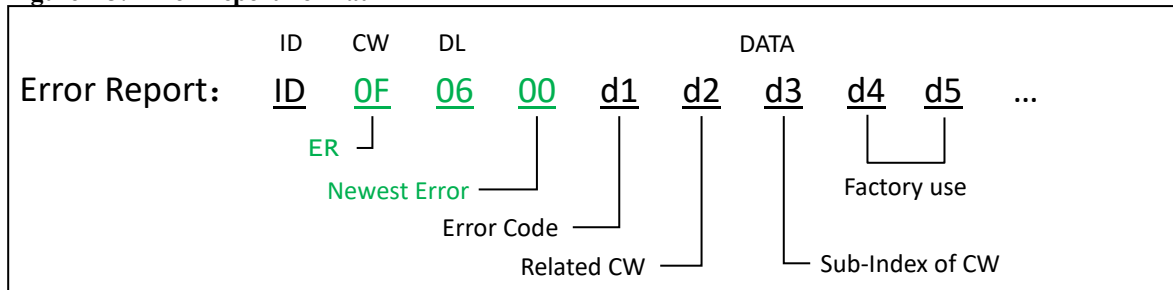
d1 - Error Code;

Error code of UIM2523	Description
0x32	Instruction Syntax error
0x33	Instruction Data error
0x34	Instruction Sub-Index error

d2 – CW related to the error;

  For example, d2 = `SN` indicates an error on previous `SN` instruction.

d3 – Sub-Index of the CW related to the error;

  For example, d2= `PP`; d3 = 5 indicates an error on previous `PP[5]` instruction.

d4, d5 – Factory use, don't care.

2.0   Instruction Set

This chapter provides details of instructions for UIM 2523.

Note:

1. Unless otherwise specified, all message bytes are in hex format;

2. The instruction examples all use the SM of 0xAD (i.e., no CRC);

3. This chapter only lists UIM2513 related instructions. Please refer to the relevant user manual for other UIM device's instructions.

4. Abbreviation definitions are listed below:

    a) INST – Instruction;

    b) ACK – Acknowledgment / Reply;

    c) SM – Start of Message;

    d) EM – End of Message;

UIM2523

2.1 `PP[i]` Protocol Parameter

<div style="text-align:right">Protocol Parameter</div>

CW	Disable ACK		0x01	Request ACK	0x81
DATA	Sub-Index(i)		unsigned 8 bit	Data	unsigned 8 bit

<table>
<tr>
<td rowspan="2">Description</td>
<td colspan="5">

`PP[i]`   **Get Protocol Parameters**
INST data length   1   Data   d0 (=i)
ACK data length   2   Data   d0 (=i), d1

`PP[i]=N`   **Set Protocol Parameters**
INST data length   2   Data   d0 (=i), d1
ACK data length   2   Data   d0 (=i), d1

</td>
</tr>
</table>

i	Description	Value (N)
5	CAN bit rate (bps)	0: 1000K 1: 800K 2: 500K 3: 250K 4: 125K
7	Node ID	Get Node ID: UIM2523 ID fixed = 2.

Example GET

INST	ID		CW	DL	d0	d1	d2	d3	d4	d5	d6	d7
	02		81	01	05	00	00	00	00	00	00	00

Sub-Index d0 = 5 (CAN bitrate).

ACK	ID	CW	DL	d0	d1	d2	d3	d4	d5	d6	d7
	02	01	02	05	01	00	00	00	00	00	00

Sub-Index d0 = 5 (CAN bitrate); Date d1 = 1 (800 K).

Example SET

INST	ID	CW	DL	d0	d1	d2	d3	d4	d5	d6	d7
	02	81	02	05	02	00	00	00	00	00	00

Sub-Index d0 = 5 (CAN bit rate); Date d1 = 2 (SET 500 K).

ACK	ID	CW	DL	d0	d1	d2	d3	d4	d5	d6	d7
	02	01	02	05	02	00	00	00	00	00	00

Sub-Index d0 = 5 (CAN bit rate); Date d1 = 2 (500 K).

Note

- The value of `PP[i]` will be saved to FLASH memory with a 10,000 write lifespan.
- Within a specific CAN network, Node IDs and Group IDs of all devices should never be overlapped.

2.2 ML Model

CW	0x8B
DATA	n/a
Description	**ML Get the model, function module and firmware version** INST data length 0 Data n/a ACK data length 8 Data d0, d1, d2, d3, d4, d5, d6, d7
Example GET	(see below)

INST	ID	CW	DL	d0	d1	d2	d3	d4	d5	d6	d7
	02	8B	00	00	00	00	00	00	00	00	00

ACK	ID	CW	DL	d0	d1	d2	d3	d4	d5	d6	d7
	02	0B	08	19	17	00	00	00	00	00	00

UIM2523 model information refer to the following table:

Model	d0	d1	d2	d3	d5: d4	d7: d6
UIM2523	0x19	0x17	0x00	xx	Firmware version	xxxx

x – Factory use.

UIM2523

2.3 SN Serial Number

CW	0x8C
DATA	n/a

Description	**SN** **Get the serial number of the device** INST data length 0 Data n/a ACK data length 8 Data d0, d1, d2, d3, d4, d5, d6, d7

Example GET

INST	ID	CW	DL	d0	d1	d2	d3	d4	d5	d6	d7
	02	8C	00	00	00	00	00	00	00	00	00

ACK	ID	CW	DL	d0	d1	d2	d3	d4	d5	d6	d7
	02	0C	08	01	02	03	04	05	06	07	08

Data:
[d3:d2:d1:d0] = 0x04030201 (Serial number 0067305985).
[d5:d4] = 0x0605 (Manufacturer ID 1541).
[d7:d6] = 0x0807 (Vendor ID 2055).

2.4 `ER[i]` Error Report

CW	Disable ACK	0x0F	Request ACK	0x8F
DATA	Sub-Index(i)	unsigned 8 bit	Data	unsigned 8 bit

describe	

`ER[i]` **Get Error Report**

INST data length 1 Data d0 (=i)

ACK data length 6 Data d0 (=i), d1, d2, d3, d4, d5

`ER[i]=0` **Clear Error Report**

INST data length 2 Data d0 (=i), d1, d2

ACK data length 6 Data d0 (=i), d1, d2, d3, d4, d5

i	Description
0	Get / Clear The latest error content
6	Get / Clear Power on error
10	Get / Clear latest errors
11…18	Get / Clear the last 2 … 9 errors

Refer to chapter 1.3 "Error Report" for the description of d0, d1, d2, d3, d4, d5.

Example GET

INST	ID	CW	DL	d0	d1	d2	d3	d4	d5	d6	d7
	02	8F	01	00	00	00	00	00	00	00	00

Sub-Index d0 = 0 (Get the latest error).

ACK	ID	CW	DL	d0	d1	d2	d3	d4	d5	d6	d7
	02	0F	06	00	14	81	00	00	00	00	00

Sub-Index d0 = 0 (The latest error).
Data
d1 = 0x14 (Error Code, *CAN message sending timeout (no response)*);
d2 = 0x81 (CW related to the error, 0x81 is PP);
d3 = 0x00 (Sub-Index of the CW related to the error);
d4, d5 = 0x00 (reserved).
Sum up, "Gateway sent the `PP[0]`, but no response is received."

Example SET

INST	ID	CW	DL	d0	d1	d2	d3	d4	d5	d6	d7
	03	8F	02	00	00	00	00	00	00	00	00

Sub-Index d0 = 0 (latest error); Data d1 = 0 (clear).

ACK	ID	CW	DL	d0	d1	d2	d3	d4	d5	d6	d7
	03	0F	06	00	00	00	00	00	00	00	00

Sub-Index d0 = 0 (latest error); Data d5 …d1 = 0 (no error).

UIM2523

2.5 `SY[i]` System Operation

CW	0x7E			
DATA	Sub-Index(i)	Unsigned 8 bit	Data	n/a
Description	`SY[i]` **System Operation** INST data length 1 Data d0 (=i) No ACK | i	Description | |---|---| | 1 | Reboot the device | | 2 | Restore factory defaults |			

i	Description
1	Reboot the device
2	Restore factory defaults

Example SET

INST	ID	CW	DL	d0	d1	d2	d3	d4	d5	d6	d7
	02	7E	01	00	00	00	00	00	00	00	00

Sub-Index d0 = 1 (Reboot the device).

ACK: n/a

3.0 Reset Method

This chapter introduces the method and steps for resetting UIM2523.

1. When resetting UIM2523, please first locate the small hole on the back of UIM2523 (non printed surface) (see Appendix-1 Dimensions), and the reset switch is located inside the small hole.

2. After powering on UIM2523, press and hold the reset button with the end of a pin or paper clip, and release it after the blue light on the DINKLE terminal flashes.

3. Wait for the indicator light on the DINKLE terminal to turn green to complete the reset.

Appendix-1 Dimensions

Unit: mm
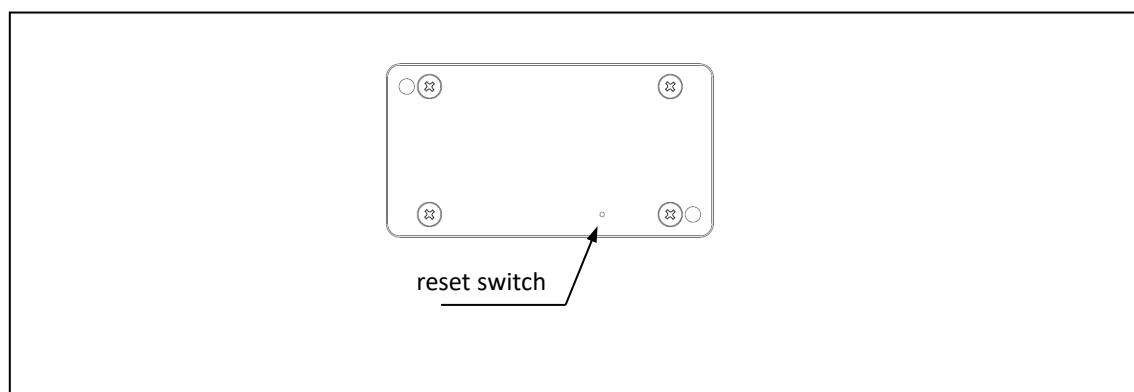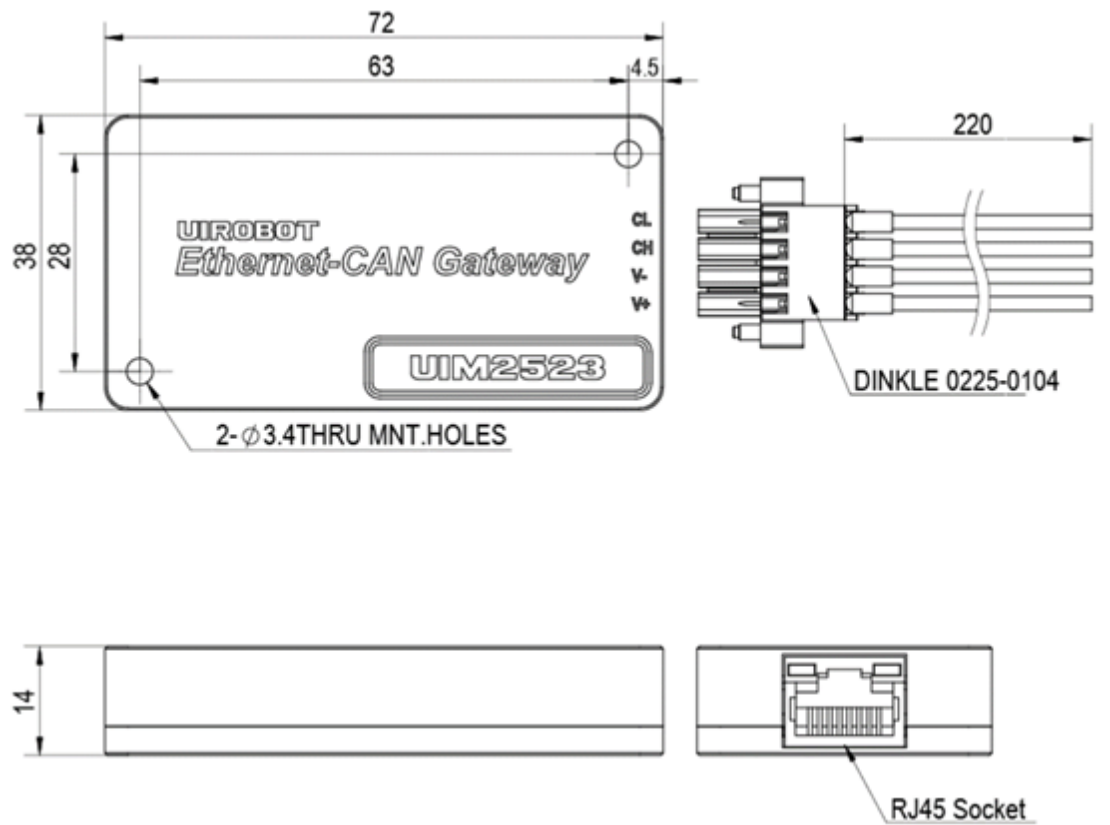
Appendix-2 RTU CRC16 Source Code

UIMessage uses CRC-16 (modbus) algorithm. For details, please refer to Online CRC-8 CRC-16 CRC-32 Calculator (crccalc.com) . The source code used in the calculation is listed below .

// CRC low byte table

unsigned char tblCRCLo [256] = {
0x00, 0xC0, 0xC1, 0x01, 0xC3, 0x03, 0x02, 0xC2, 0xC6, 0x06, 0x07, 0xC7, 0x05, 0xC5, 0xC4, 0x04, 0xCC, 0x0C, 0x0D,
0xCD, 0x0 F, 0xCF, 0xCE, 0x0E, 0x0A, 0xCA, 0xCB, 0x0B, 0xC9, 0x09, 0x08, 0xC8, 0xD8, 0x18, 0x19, 0xD9, 0x1B,
0xDB, 0xDA, 0x1A, 0x1E, 0xDE, 0xDF, 0x1F, 0xDD, 0x1D, 0x 1C, 0xDC, 0x14, 0xD4, 0xD5, 0x15, 0xD7, 0x17, 0x16,
0xD6, 0xD2, 0x12, 0x13, 0xD3, 0x11, 0xD1, 0xD0, 0x10, 0xF0, 0x30, 0x31, 0xF1, 0x33, 0xF3, 0 xF2, 0x32, 0x36, 0xF6,
0xF7, 0x37, 0xF5, 0x35, 0x34, 0xF4, 0x3C, 0xFC, 0xFD, 0x3D, 0xFF, 0x3F, 0x3E, 0xFE, 0xFA, 0x3A, 0x3B, 0xFB, 0x39,
0xF9, 0xF8, 0x38, 0 x28, 0xE8, 0xE9, 0x29, 0xEB, 0x2B, 0x2A, 0xEA, 0xEE, 0x2E, 0x2F, 0xEF, 0x2D, 0xED, 0xEC,
0x2C, 0xE4, 0x24, 0x25, 0xE5, 0x27, 0xE7, 0xE6, 0x26, 0x22, 0 xE2, 0xE3, 0x23, 0xE1, 0x21, 0x20, 0xE0, 0xA0, 0x60,
0x61, 0xA1, 0x63, 0xA3, 0xA2, 0x62, 0x66, 0xA6, 0xA7, 0x67, 0xA5, 0x65, 0x64, 0xA4, 0x6C, 0 xAC, 0xAD, 0x6D,
0xAF, 0x6F, 0x6E, 0xAE, 0xAA, 0x6A, 0x6B, 0xAB, 0x69, 0xA9, 0xA8, 0x68, 0x78, 0xB8, 0xB9, 0x79, 0xBB, 0x7B,
0x7A, 0xBA, 0xBE, 0x7E, 0x7F, 0 xBF, 0x7D, 0xBD, 0xBC, 0x7C, 0xB4, 0x74, 0x75, 0xB5, 0x77, 0xB7, 0xB6, 0x76,
0x72, 0xB2, 0xB3, 0x73, 0xB1, 0x71, 0x70, 0xB0, 0x50, 0x90, 0x91, 0 x51, 0x93, 0x53, 0x52, 0x92, 0x96, 0x56, 0x57,
0x97, 0x55, 0x95, 0x94, 0x54, 0x9C, 0x5C, 0x5D, 0x9D, 0x5F, 0x9F, 0x9E, 0x5E, 0x5A, 0x9A, 0x9B, 0x5B, 0 x99, 0x59,
0x58, 0x98, 0x88, 0x48, 0x49, 0x89, 0x4B, 0x8B, 0x8A, 0x4A, 0x4E, 0x8E, 0x8F, 0x4F, 0x8D, 0x4D, 0x4C, 0x8C, 0x44,
0x84, 0x85, 0x45, 0x87, 0 x47, 0x46, 0x86, 0x82, 0x42, 0x43, 0x83, 0x41, 0x81, 0x80, 0x40 };

// CRC high byte table

unsigned char tblCRCHi [256] = {
0x00, 0xC1, 0x81, 0x40, 0x01, 0xC0, 0x80, 0x41, 0x01, 0xC0, 0x80, 0x41, 0x00, 0xC1, 0x81, 0x40, 0x01, 0xC0, 0x80,
0x41, 0 x00, 0xC1, 0x81, 0x40, 0x00, 0xC1, 0x81, 0x40, 0x01, 0xC0, 0x80, 0x41, 0x01, 0xC0, 0x80, 0x41, 0x00, 0xC1,
0x81, 0x40, 0x00, 0xC1, 0x81, 0x40, 0x01, 0 C0, 0x80, 0x41, 0x00, 0xC1, 0x81, 0x40, 0x01, 0xC0, 0x80, 0x41, 0x01,
0xC0, 0x80, 0x41, 0x00, 0xC1, 0x81, 0x40, 0x01, 0xC0, 0x80, 0x41, 0x00, 0xC1, 0 x81, 0x40, 0x00, 0xC1, 0x81, 0x40,
0x01, 0xC0, 0x80, 0x41, 0x00, 0xC1, 0x81, 0x40, 0x01, 0xC0, 0x80, 0x41, 0x01, 0xC0, 0x80, 0x41, 0x00, 0xC1, 0x81, 0
x40, 0x00, 0xC1, 0x81, 0x40, 0x01, 0xC0, 0x80, 0x41, 0x01, 0xC0, 0x80, 0x41, 0x00, 0xC1, 0x81, 0x40, 0x01, 0xC0, 0x80,
0x41, 0x00, 0xC1, 0x81, 0x40, 0 x00, 0xC1, 0x81, 0x40, 0x01, 0xC0, 0x80, 0x41, 0x01, 0xC0, 0x80, 0x41, 0x00, 0xC1,
0x81, 0x40, 0x00, 0xC1, 0x81, 0x40, 0x01, 0xC0, 0x80, 0x41, 0x00, 0 xC1, 0x81, 0x40, 0x01, 0xC0, 0x80, 0x41, 0x01,
0xC0, 0x80, 0x41, 0x00, 0xC1, 0x81, 0x40, 0x00, 0xC1, 0x81, 0x40, 0x01, 0xC0, 0x80, 0x41, 0x01, 0xC0, 0 x80, 0x41,
0x00, 0xC1, 0x81, 0x40, 0x01, 0xC0, 0x80, 0x41, 0x00, 0xC1, 0x81, 0x40, 0x01, 0xC0, 0x80, 0x41, 0x00, 0xC1, 0x81, 0x40,
0x41, 0x00, 0xC1, 0x81, 0 x40, 0x01, 0xC0, 0x80, 0x41, 0x01, 0xC0, 0x80, 0x41, 0x00, 0xC1, 0x81, 0x40, 0x01, 0xC0,
0x80, 0x41, 0x00, 0xC1, 0x81, 0x40, 0x00, 0xC1, 0x81, 0x40, 0 x01, 0xC0, 0x80, 0x41, 0x01, 0xC0, 0x80, 0x41, 0x00,
0xC1, 0x81, 0x40, 0x00, 0xC1, 0x81, 0x40, 0x01, 0xC0, 0x80, 0x41, 0x00, 0xC1, 0x81, 0x40, 0x01, 0 xC0, 0x80, 0x41,
0x01, 0xC0, 0x80, 0x41, 0x00, 0xC1, 0x81, 0x40 };

// Calculate CRC, * buf points to 0xAA ID CW … d8 data sequence, q ty = 13 bytes

```
unsigned short RtuCrc16(unsigned char* buf , unsigned int qty)
{
    unsigned char crcH = 0xFF;
    unsigned char crcL = 0xFF;
    int idx = 0;

    while (qty--)
    {
        idx = crcL ^ * buf ++;
        crcL = crcH ^ tblCRCHi [ idx ];
        crcH = tblCRCLo [ idx ];
    }

    return ( crcH << 8) | crcL ;
}
```