UIROBOT®
United Intelligence Robot Technology

# User Manual

## UIM2501
## RS232-CAN2.0B Converting Controller
## for UIM242XX Motion Controller

Please pay attention to the following before using the UIROBOT products:

- UIROBOT products meet the specification contained in their particular Data Sheet.
- UIROBOT will only work with the customer who respects the Intellectual Property (IP) protection.
- Attempts to break UIROBOT's IP protection feature may be a violation of the local Copyright Acts. If such acts lead to unauthorized access to UIROBOT's IP work, UIROBOT has a right to sue for relief under that Act.

Information contained in this publication regarding controller applications and the like is provided only for your convenience and may be superseded by updates. It is your responsibility to ensure that your application meets with your specifications. UIROBOT MAKES NO REPRESENTATIONS OR WARRANTIES OF ANY KIND WHETHER EXPRESS OR IMPLIED, WRITTEN OR ORAL, STATUTORY OR OTHERWISE, RELATED TO THE INFORMATION, INCLUDING BUT NOT LIMITED TO ITS CONDITION, QUALITY, PERFORMANCE, MERCHANTABILITY OR FITNESS FOR PURPOSE. UIROBOT disclaims all liability arising from this information and its use. Use of UIROBOT products in life support and/or safety applications is entirely at the buyer's risk, and the buyer agrees to defend, indemnify and hold harmless UIROBOT from any and all damages, claims, suits, or expenses resulting from such use. No licenses are conveyed, implicitly or otherwise, under any UIROBOT intellectual property rights.

**[Trade Mark/ Layout-design/Patent]**

The UIROBOT name and logo are registered trademarks of UIROBOT Ltd. in the P.R. China and other countries.
UIROBOT's UIM24XXX series Step Motor Controllers, UIM25XX series CAN-RS232 Converter and their layout designs are patent protected.

# UIM2501
# RS232-CAN2.0B Converting Controller
# (For UIM242XX series Controller)

## Features

**Embedded DSP Microprocessor**

- Embedded high-performance DSP (Digital Signal Processor)
- Simple, intuitive, rich instructions
- Intelligent, fault-tolerating, user-friend interface

**CAN2.0B Active Communication**

- 2-wire interface
- 1 Mega bit/sec operation, long distance
- Differential data bus, high noise immunity

**RS232 Communication**

- RS232 three-wire serial communication
- Max baud rate 112500 bps

**Wide Supply Voltage**

- Wide supply voltage range 6 ~ 40VDC

## Description

The UIM2501 RS232-CAN Converting Controller is used in conjunction with UIM242XX stepper motor controller to provide a RS232 interface on the user side and a CAN bus interface on the motor side (factory side).

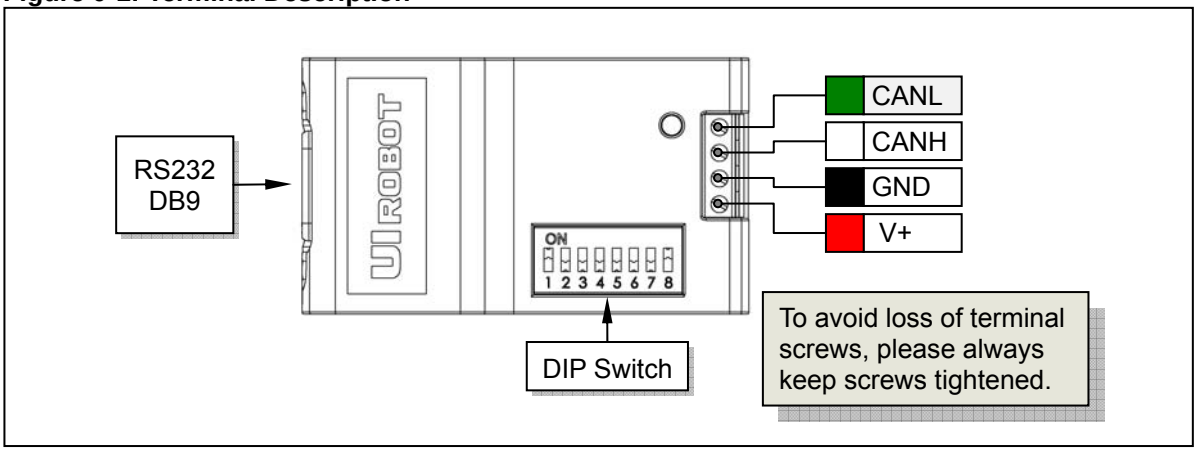
With the UIM2501, user will be benefited from the advantages of the CAN network and the simplicity of RS232 protocol, and no need to deal with the complicated CAN protocol, no worry about the communication distance and noise immunity.

One UIM2501 controller can network with up to 100 UIM242XX controllers. Interfacing the UIM2501 is simple, intuitive and fault tolerating. Users are not required to have stepper motor driving or CAN protocol knowledge.

UIM2501 is compact in size. The enclosure is made of die-cast aluminum to provide a rugged durable protection and improves the heat dissipation.

# Terminal Description

**Figure 0-1: Terminal Description**




## Description of Screw Terminals

| Terminal No. / Color | Description | | Input / Output | | | |
|---|---|---|---|---|---|---|
| | | | MIN | NOM | MAX | UNIT |
| 1 / Red | V+ | Supply voltage | 6 | | 40 | VDC |
| 2 / Black | GND | Supply voltage ground | | 0 | | VDC |
| 3 / White | CANH | CAN High-Level Voltage I/O | | | | VDC |
| 4 / Green | CANL | CAN Low-Level Voltage I/O | | | | VDC |

RS232 Connector is a DB-9 (Female Pin) connector.

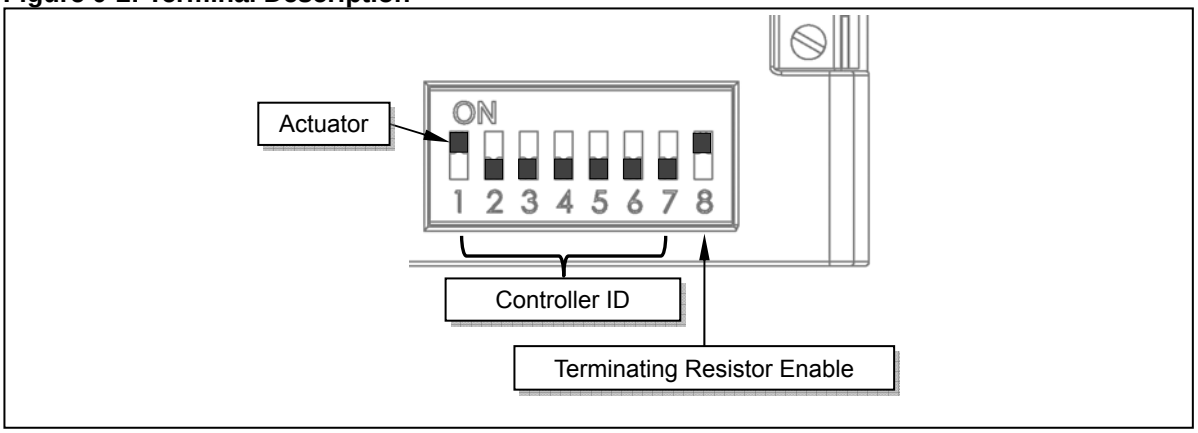
## DIP Switch

UIM2501 controller has an 8-bit DIP switch, which serves multiple functions. When powering up, DIP1~DIP7 are read as the UIM2501 controller's ID/address. After powering up, DIP1 and DIP2 are assigned to other RS232 related features (refer to the "RS232 communication" section).

DIP8 is used to enable the built-in terminating resistor.

Unless necessary, please maintain the DIP positions as shown in following Figure.

**Figure 0-2: Terminal Description**



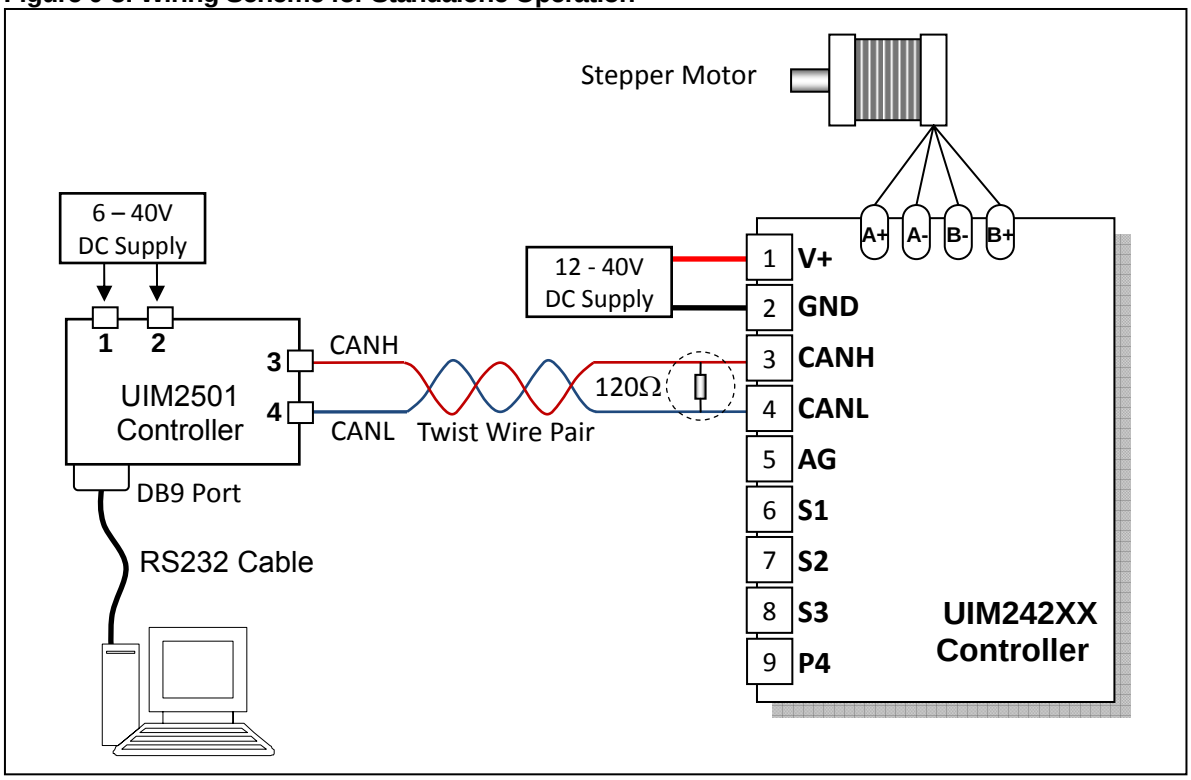
## Typical Application

### Standalone Operation

When only one UIM242 controller is needed, user can use the following wiring scheme.

Please note that, the standalone wiring scheme is mandatory when assigning a controller ID to a UIM242 controller (motor is not required). For details, please refer to section 3.1.

**Figure 0-3: Wiring Scheme for Standalone Operation**




Notice:

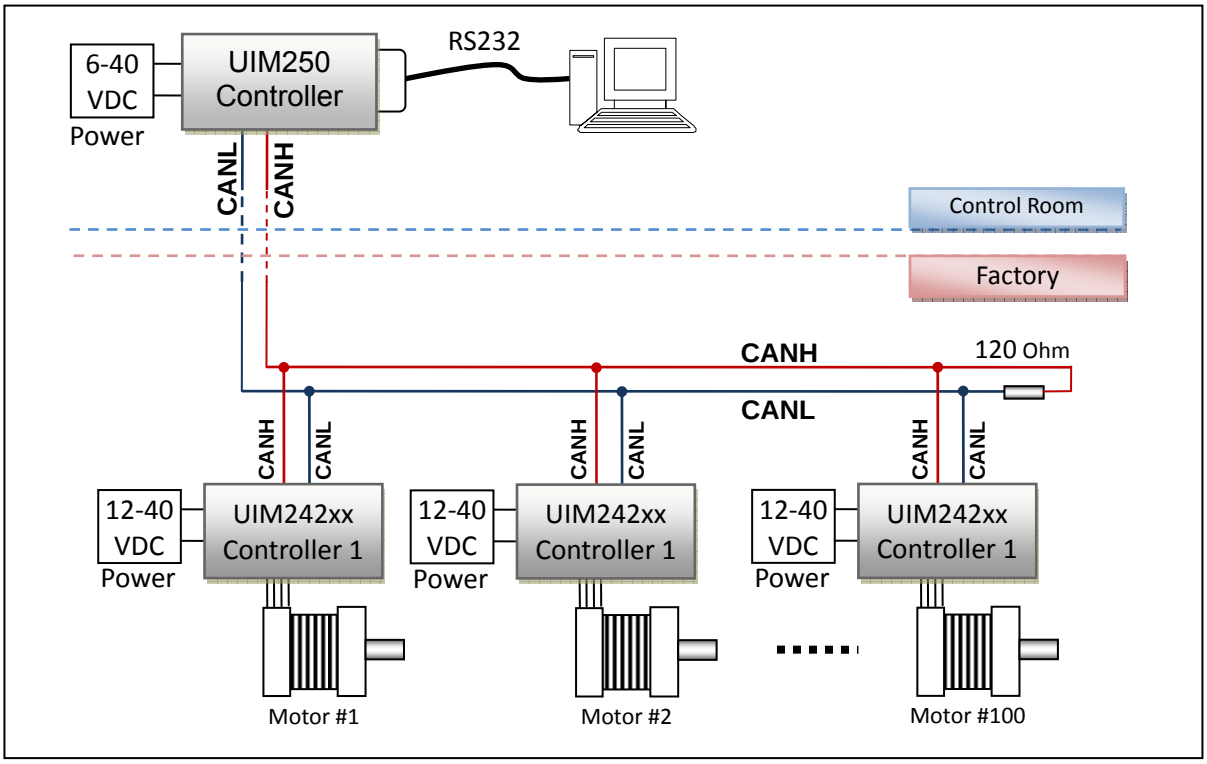
For long distance transfer, both ends of the bus should be terminated with120Ω terminating resistors. UIM2501 converter already has a build-in terminating resistor. To enable it, user needs toggle the DIP8 to the ON position. On the UIM242 controller side, user needs to attach a resistor to the end of the bus as shown in above figure 0-3.

To achieve the best communication, CANH and CANL should be a twisted wire pair.

## Network Operation

Multiple UIM242XX controllers can be wired together to form a reliable motor network. Following figure provides a typical network wiring solution.

**Figure 0-4: Wiring Scheme for Network Operation**




Notice:

In multi-node CAN applications, it is important to maintain a direct point-to-point wiring scheme. A single pair of wires should connect each element of the CAN bus, and the two ends of the bus should be terminated with 120Ω resistors. A star configuration should never be used.

UIM2501 converter has a build-in terminal resistor. To enable the UIM2501 converter's terminating resistor, please toggle the DIP8 to ON position. User only needs to attach a resistor at the UIM242 end of the bus.

In addition, any deviation from the point-to-point wiring scheme creates a stub. The high-speed edge of the CAN data on a stub can create reflections back down the bus. These reflections can cause data errors by eroding the noise margin of the system. Although stubs are unavoidable in a multi-node system, care should be taken to keep these stubs as small as possible.

## Characteristics

### Absolute Maximum Ratings (†)

Supply Voltage..... 6V to 40V
Voltage on RX with respect to GND ..... -25V to +25V
Voltage on TX with respect to GND..... -13.2V to +13.2V
Ambient temperature under bias……..... -20°C to +85°C
Storage temperature..... -50°C to +150°C

†NOTICE: 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 operation listings of this specification is not implied. Exposure to maximum rating conditions for extended periods may affect device reliability.

### Electrical Characteristics（Ambient Temperature 25℃）

| | |
|---|---|
| Supply Power Voltage | 6V ~ 40VDC |
| Current Consumption | Max 100mA |

### Communication (Ambient Temperature 25℃）

| | |
|---|---|
| To User Device | RS232 |
| Wiring Method | DB9 Female Connector |
| RS232 Baud Rate | MAX 115200 bps |
| To UIM242 Controller | Active CAN 2.0B |
| CAN wiring Method | 2-Wire,CANH,CANL |
| CAN bus | • Supports 1 Mb/s operation<br>• ISO-11898 standard physical layer requirements<br>• Suitable for 12V and 24V systems<br>• Up to 100 nodes can be connected |

### Environment Requirements

| | | |
|---|---|---|
| Cooling | | Free Air |
| Working | Environment | Avoid dust, oil mist and corrosive gases |
| | Temperature | -20 ℃ ~ + 85 ℃ |
| | Humidity | <80%RH，no condensation, no frosting |
| | Vibration | 3G Max |
| Storage Temperature | | -50 ℃ ~ + 150 ℃ |

### Size and Weight

| | |
|---|---|
| Size | 66.4mm x 38mm x 18mm (L*W*H) |
| Wight | 0.1 kg |

# Table of Contents

# OVERVIEW

## 1.0 OVERVIEW

UIM2501 RS232-CAN Converting Controller is used in conjunction with UIM242 stepper motor controller to provide a RS232 interface on the user side and a CAN bus interface on the motor controller side.

UIM2501 Controller's functions can be summarized as:

1. Receives RS232 based instructions from user device, converts the instructions into more concise and efficient CAN based instructions, and sends instructions to UIM242 controllers.
2. Converts CAN messages from UIM242 controllers into RS232 messages, and send back to the user devices.
3. Coordinates and controls UIM242 controllers in the network.

With the UIM2501, user will be benefited from the advantages of the CAN network and the simplicity of RS232 protocol, and no need to deal with the complicated CAN protocol, no worry about the communication distance and noise immunity.

One UIM2501 controller can network with up to 100 UIM242XX controllers. Interfacing the UIM2501 is simple, intuitive and fault tolerating. Users are not required to have stepper motor driving or CAN protocol knowledge. UIM2501 supports 1Mbps CAN communication speed. After optimization, all UIM242 instructions take less than 0.1 ms (normally 0.05 ms) to transfer on the bus. On the user side, UIM2501 supports 115200 bps RS232 baud rate.

UI Robot provides free Microsoft Windows XP bases VB / VC demo software and their source code to facilitate the quick start of user device side programming.

## 1.1 Instruction and Feedback Structure

As a protocol converter, UIM2501supports all instructions and conform to all instruction and feedback structures of UIM242XX. Therefore, this information is omitted in this manual. User can refer to the UIM242XX User Manual for details.

## 1.2 Motor Control Functions and Instructions

Motor control specific functions, instructions and operations are directly performed by the UIM242 controllers. As a RS232-CAN converter, UIM2501 accepts all the instructions described in the UIM242XX User Manual. User can refer to the UIM242XX User Manual for detailed information.

# RS232 COMMUNICATION

## 2.0 RS232 COMMUNICATION

UIM2501 controller communicates and exchanges information with user devices through the RS232 serial protocol. The RS232 configuration of user device, the hand-shaking methods, and the instruction used to change the baud rate will be introduce in this Chapter, along with the method to reset the baud rate to factory default.

## 2.1 Settings for User Device RS232 Port

To communicate with UIM2501, user device needs to have following RS232 port settings:

1. 8 bits data
2. 1 stop bit
3. None Parity

## 2.2 Hand-Shaking

If user device knows the baud rate, it can start sending instructions without hand-shaking.

Hand-shaking is more used as a method to check the existence and firmware version of the controller. Hand-shaking is considered successful, if the user device receives a greeting message starting with **0xAA, 0xAB, 0xAC**.

Under following two situations the UIM2501 will issue a greeting message:

1. When UIM2501 is powering up.
2. When UIM2501 receives following ASCII message: **ABC;** (case sensitive and ended with a semicolon)

A greeting message from UIM2501 has the following structure:

**0xAA 0xAB 0xAC 0x19 0x1 0 0 0 0 0 [ver1] [ver0] 0xFF**

Where,

| | |
|---|---|
| **0xAA, 0xAB, 0xAC** | denotes the greeting message. |
| **0x19, 0x1** | denotes the UIM2501 controller. |
| **[ver1:ver0]** | denotes the firmware version. |

## 2.3 Baud Rate Change Instruction（BDR）

Any out-of-box UIM2501 controller has a factory default baud rate 9600. User can use the 9600 baud rate to connect to a new UIM2501 controller.

To change to a different baud rate, user can use the instruction BDR as described below.

On receiving the BDR instruction, the new baud rate will be stored in the EEPROM and will take effect after the controller is restarted.

**Instruction BDR**

| | | | |
|---|---|---|---|
| Function | Change the RS232 baud rate | | |
| Syntax | **BDR = x;** | Variable | Integer x = 9600 … 115200 |
| ACK | **0xAA [Reserved] 0xBD 0xFF**<br>0xBD is the Message ID of instruction BDR.<br>The **[Reserved]** is for factory use. | | |
| Comment | New Baud Rate will be stored in the controller's non-volatile memory (EEPROM). New baud rate will take effect after the controller is restarted. | | |

## 2.4 Reset Baud Rate to Factory Default 9600

In case that user forgets the UIM2501 baud rate and cannot establish the connection, following process can reset the baud rate to the factory default of 9600:

1. Reboot the controller.
2. In 10 seconds, toggle the DIP1 (DIP switch 1) for two rounds. During toggling, the LED on the controller will flash. If exceed 10 seconds, please restart from the first step.
3. If step 2 is successful, the LED will turn off for one second and re-lit. That indicates the baud rate has been changed to 9600 and ready to use.
4. The BDR instruction can be used to change the 9600 Baud Rate.

# STANDALONE AND NETWORKING OPERATION

## 3.0 STANDALONE AND NETWORKING OPERATION

UIM2501 supports CAN networking. One UIM2501 can communicate with one or several UIM242 controller(s).

Before operation, every UIM242 controller needs to be assigned a unique identification number (i.e., ID or address). Two or more UIM242 controllers with an identical ID in the network may cause the network malfunctioning. The process of ID assignment is described in section 3.2.

## 3.1 Global Control Introduction

Besides the object specific operation, UIM2501 can also send commands to all subsidiary UIM242XX controllers. This process is called broadcasting or global control instruction.

### 3.1.1 Global Instruction Format

Global control instructions have the same format as shown below: a leading “g” followed by the normal one-to-one operation instruction.

**gXYZ;**

Where,

**g** denotes the global control instruction.

**XYZ** is the normal one-to-one operation instruction.

### 3.1.2 Global Instruction ACK Message

Except the instruction **gREG**, all global instructions have the same ACK message as shown below:

**0xAA [QTY] 0xAD 0xFF**

Where, **[QTY]** is the quantity of all UIM242xx controllers that are operable.

Detailed global instructions are described in section 3.4 ~ 3.13.

## 3.2 Controller ID Assignment Instruction（SETADR）

All new UIM242xx controllers have been assigned a factory default ID of 5.

User can change the ID using SETADR instruction. Before assign an ID to a UIM242XX controller, please make sure the UIM2501 controller and the UIM242XX controller are connected together using the standalone operation scheme (Figure 0-3).

After powering up, user can use the following instruction to assign an ID to the UIM242XX controller.

**Instruction SETADR**

| | | | |
|---|---|---|---|
| Function | Assign an ID to a UIM242XX controller | | |
| Syntax | **SETADR = x;** | Variable | Integer x = 5, 6, 7 … 126 |
| ACK | **0xAA [Controller ID] 0xDD 0xFF**<br>0xDD is the Message ID of SETADR. | | |
| Comment | SETADR is the abbreviation of “Set Address”.<br>Every time there can be only one UIM242XX controller linked to the UIM2501 controller to get the ID assigned. Once an ID is assigned, the ID will be stored in the UIM242XX controller’s non-volatile memory. | | |

## 3.3 Object Specify Instruction（ADR）

To operate a specific UIM242xx controller, user needs to first notify the UIM2501 controller which UIM242 controller is suppose to receive instructions. Therefore, before sending motor control instructions, user needs to use the following instruction to specify the UIM242xx controller’s ID.

**Instruction ADR**

| | | | |
|---|---|---|---|
| Function | Specify the operation object by its ID | | |
| Syntax | **ADR = x;** | Variable | Integer ID = 5, 6, 7 … 126 |
| ACK | **0xAA [Controller ID] 0xD0 0xFF**<br>0xD0 is the Message ID of ADR. | | |
| Comment | ADR is the abbreviation of address.<br>Once an operation object is specified, all object specific instructions will be sent to this specified object (the UIM242xx Controller), UNTILL another operation object (new ID) is specified. | | |

## 3.4 Global Register Instruction (gREG)

**Instruction gREG**

| | | | |
|---|---|---|---|
| Function | Record the quantity of all subsidiary UIM242 controllers and their IDs | | |
| Syntax | **gREG;** | Variable | N/A |
| Feedback | **0xCC [QTY] 0xD0 A1 A2 A3 A4 A5 A6 A7 A8 0xFF**<br>QTY is the quantity of UIM242xx controllers found.<br>0xD0 is the Message ID of gREG. (Notice, 0xD0 also is the Message ID of ADR. However, ACK message of ADR is started with 0xAA, while the feedback message of gREG is started with 0xCC. )<br>A1~A8 are the first 8 IDs found. Ax=0 means not found. | | |
| Comment | Once this instruction is executed by the UIM2501 Controller, the total number of all subsidiary UIM242xx controllers and their ID are recorded in the UIM2501 controller. The returned QTY servers as an indicator of the health of the network. | | |

## 3.5 Global Motor Enable Instruction (gENABLE)

**Instruction gENABLE**

| | | | |
|---|---|---|---|
| Function | Enable the H-Bridge of all UIM242xx Controller | | |
| Syntax | **gENABLE;** | Variable | N/A |
| ACK | **0xAA [QTY] 0xAD 0xFF** | | |
| Comment | User can first set the motion parameters to every UIM242xx Controller one by one, and then issue this instruction to enable all motors' motion. | | |

## 3.6 Global Motor Disable Instruction (gOFFLINE)

**Instruction gOFFLINE**

| | | | |
|---|---|---|---|
| Function | Disable the H-Bridge of all UIM242xx Controller | | |
| Syntax | **gOFFLINE;** | Variable | N/A |
| ACK | **0xAA [QTY] 0xAD 0xFF** | | |
| Comment | User can shutdown all motors using this instruction, especially in emergency. | | |

## 3.7 Global Motor Current Setup Instruction (gCUR)

**Instruction gCUR**

| | | | |
|---|---|---|---|
| Function | Setup motor current for all subsidiary UIM242 Controllers | | |
| Syntax | **gCUR = x;** | Variable | Integer x = 0、1 … 80 |
| ACK | **0xAA [QTY] 0xAD 0xFF** | | |

## 3.8 Global Auto Current Reduction Instruction (gACR)

**Instruction gACR**

| | | | |
|---|---|---|---|
| Function | Enable/disable the ACR function of all subsidiary UIM242 Controllers | | |
| Syntax | **gACR = x;** | Variable | Integer x = 0, 1 |
| ACK | **0xAA [QTY] 0xAD 0xFF** | | |

## 3.9 Global Micro-Stepping Resolution Setup Instruction (gMCS)

**Instruction gMCS**

| | | | |
|---|---|---|---|
| Function | Setup micro-stepping resolution for all subsidiary UIM242 Controllers | | |
| Syntax | **gMCS = x;** | Variable | Integer x = 1, 2, 4, 16 |
| ACK | **0xAA [QTY] 0xAD 0xFF** | | |

## 3.10 Global Direction Instruction (gDIR)

**Instruction gDIR**

| | | | |
|---|---|---|---|
| Function | Setup motion direction for all subsidiary UIM242 Controllers | | |
| Syntax | **gDIR = x;** | Variable | Integer x = 0, 1 |
| ACK | **0xAA [QTY] 0xAD 0xFF** | | |

## 3.11 Global Speed Setup Instruction (gSPD)

**Instruction gSPD**

| | | | |
|---|---|---|---|
| Function | Setup motor speed for all subsidiary UIM242 Controllers | | |
| Syntax | **gSPD = x;** | Variable | Integer x = 0, 1 … 65535 |
| ACK | **0xAA [QTY] 0xAD 0xFF** | | |

## 3.12 Global Position Control Instruction (gSTP)

**Instruction gSTP**

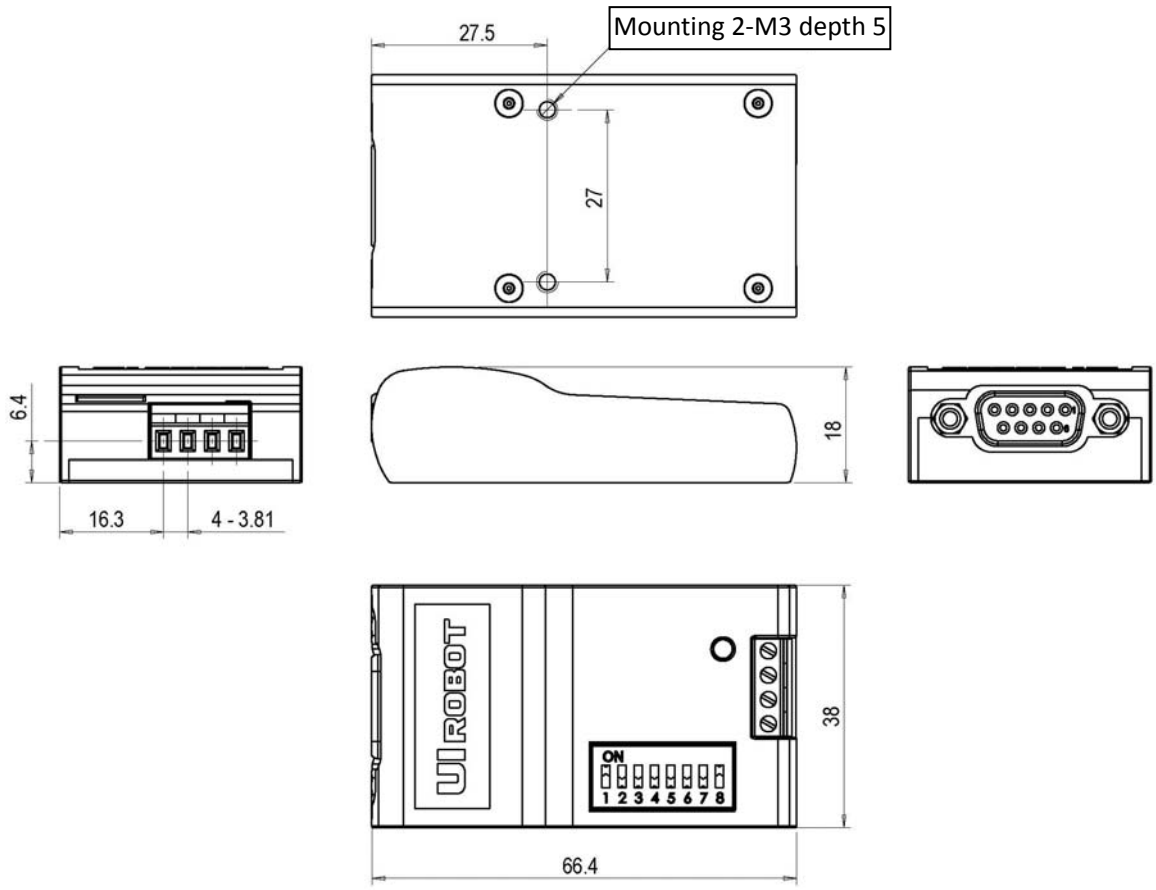
| | | | |
|---|---|---|---|
| Function | Setup relative position control for all subsidiary UIM242 Controllers | | |
| Syntax | **gSTP = x;** | Variable | Integer x = 0、1 … 2,000,000,000 |
| ACK | **0xAA [QTY] 0xAD 0xFF** | | |

## 3.13 Global Origin Point Setup Instruction (gORG)

**Instruction gORG**

| | | | |
|---|---|---|---|
| Function | Clear absolute position counters of all subsidiary UIM242 Controllers | | |
| Syntax | **gORG;** | Variable | N/A |
| ACK | **0xAA [QTY] 0xAD 0xFF** | | |

## Appendix A Dimensions




单位：mm