

RM10x0 instruction manual



www.rockmong.com

上海岩獾科技有限公司
V1.5

Contents

RM10x0 instruction manual.....	1
一、 Product Features	3
二、 Product selection	3
三、 Main parameters	3
四、 Wiring instructions	4
五、 Drive.....	4
六、 Operation indicator light.....	4
七、 Introduction to Debugging Tool Usage	5
1. Overview of Tools	5
2. Digital I/O	6
2.1 Introduction to Pin Modes.....	6
2.2 Introduction to Internal Resistance	6
2.3 Status Introduction	7
3. Modify the power on status	7
4. Modify SN.....	8
八、 Introduction to Basic Programming Library Functions.....	8
1. Obtain the device.....	8
2. Configure IO port input/output direction	8
3. Read input status	9
4. Control output status	9
5. Read the output status.....	9
九、 Advanced programming - operating multiple IOs simultaneously.....	9
1. Initialize multiple IOs simultaneously.....	10
2. Multiple IO reads simultaneously	10
3. Multiple IO writes simultaneously	10
4. Multiple output IOs simultaneously read status	11

一、 Product Features

- USB to 16 channel digital IO, each channel can be individually configured with input and output directions through software
- USB communication speed: One command read and write only takes 0.5ms (depending on factors such as computer environment)
- Support for secondary development: No need to worry about the underlying implementation, just call library functions, multi-threaded, easy to port, simple to use, and efficient
- Cross platform: Supports Windows Linux、Android、Mac OS
- Provide mainstream language examples: C/C++, C #, Python, Java, LabView, etc
- Support modifying the input and output status when powered on
- USB overcurrent protection, interface IO transient suppression protection

二、 Product selection

model	IO port high voltage
RM1000	3.3V
RM1010	5V

三、 Main parameters

parameter	describe
Rated voltage of power supply	USB power supply (MAX 500mA) or external DC5V
input/output	16 channels of digital IO, each channel can be individually configured with software for input and output direction
Output driving capability	Maximum 10mA; Generally used for signal transmission control, if you need a high current drive controller, please choose the RM16 series.
communication interface	USB
communication rate	Maximum 2KHz
temperature range	Industrial grade, -40℃~85℃

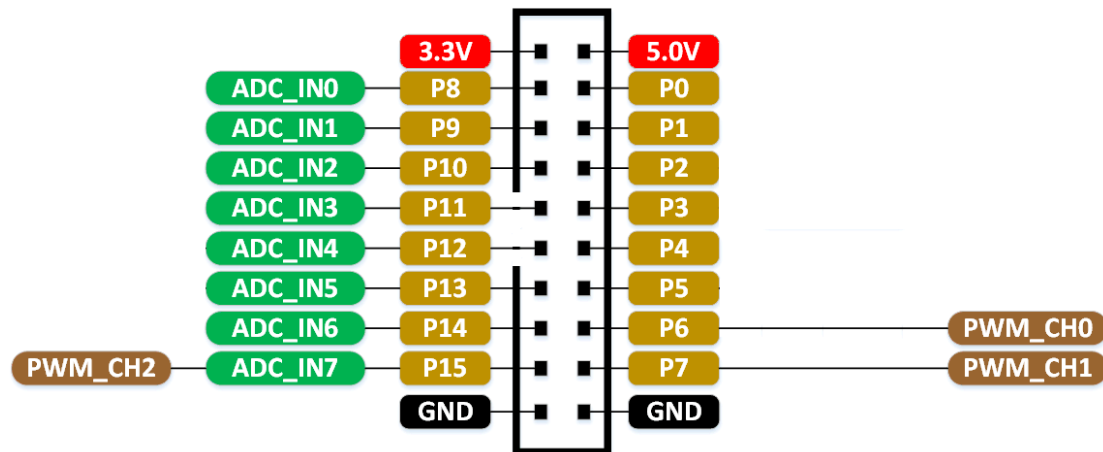
1. 5V can be input or output. But USB power supply has a maximum of 500mA, so the external output current will be less than 500mA.。
2. Power supply with 3.3V output.。

四、Wiring instructions

The 5V power supply can be input or output. But USB power supply has a maximum of 500mA, so the external output current will be less than 500mA.

The power supply has an output of 3.3V.

Power supply GND, ground wire. External sensors or devices are used for common ground. P0~P15,16 input/output channels. Each channel can be individually configured with software for input and output directions. In input mode, software can be configured to enable pull-up and pull-down resistors for certain purposes (such as key/switch input recognition, etc.).

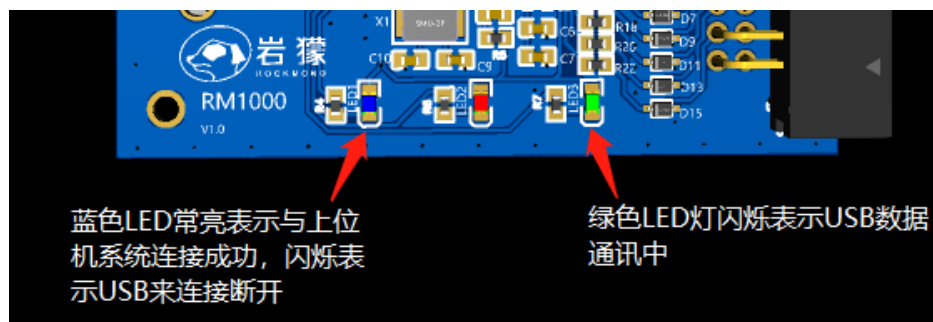


五、Drive

Windows, Linux, Android, Mac OS require no installation of drivers. (Win7 or lower versions require installation)

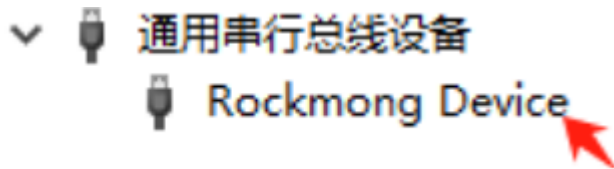
六、Operation indicator light

Operation indicator light as shown in the following figure:



If the blue LED is off, it indicates that the USB connection was not successful. Please check the following items:

1. Is the USB cable connected properly.
2. Under Windows, it is necessary to check if the USB driver is installed properly. Other systems are designed without drivers, so there is no need to check this.



3. Whether the working voltage is normal.

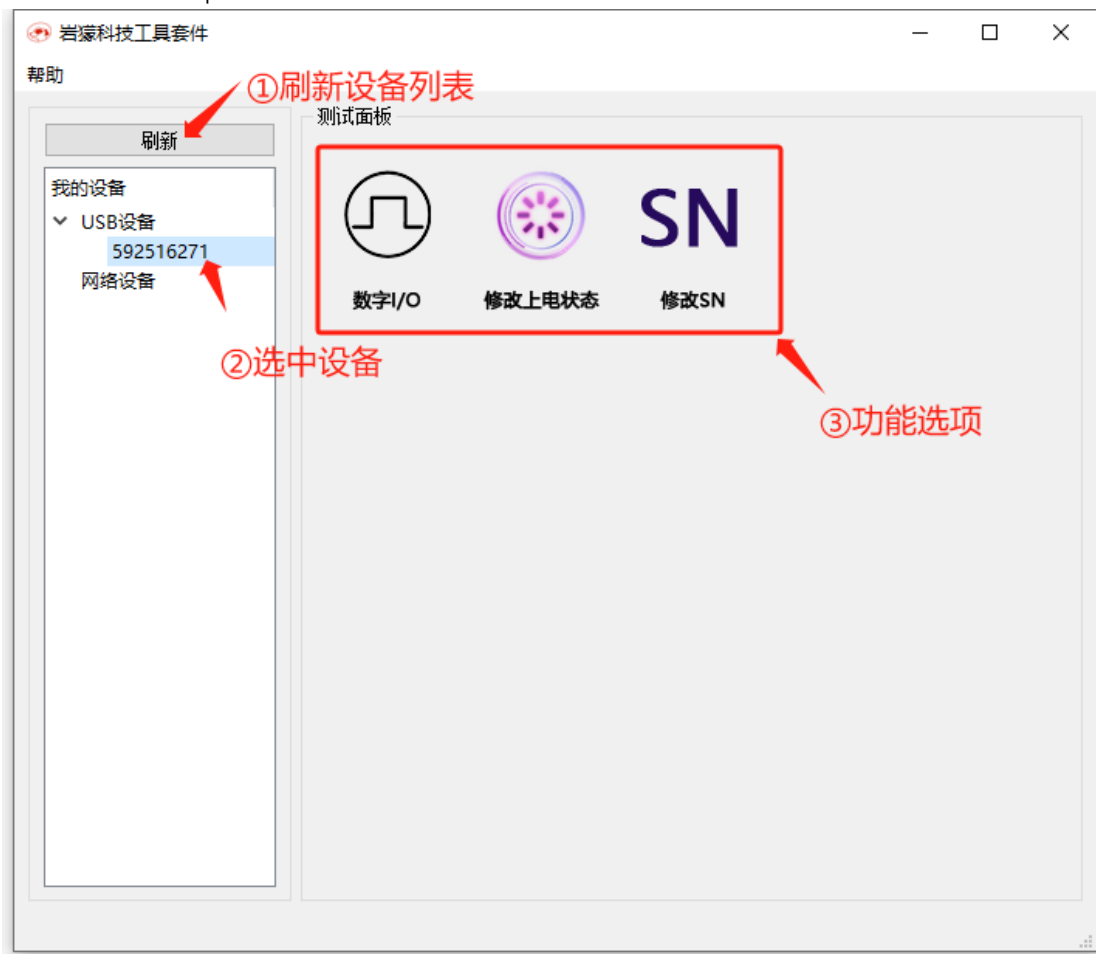
七、 Introduction to Debugging Tool Usage

Debugging tool icon:



1. Overview of Tools

Double click to open:



2. Digital I/O

Used for real-time reading and writing of IO. After clicking on "Digital I/O" to open it:



2.1 Introduction to Pin Modes

1. Input mode: used to read the level status of IO. For example, it is used to identify switch status, read sensor outputs, and so on.
2. Output mode: Used to control the output IO level status. For example, controlling transistors to drive relays, light bulbs, and so on.

2.2 Introduction to Internal Resistance

Used in input mode. In addition to some wiring purposes, it can also play an anti-interference role.

1. Pull up: An internal pull-up resistor is enabled on this pin. When this pin is not connected to a load, it will remain at a high level.

For example, in the following figure, there is a switch with one end connected to P0 and the other end connected to GND. When the switch is turned off, read P0 as high level. When the switch is turned on, read P0 as low level.



2. Pull down: An internal pull-down resistor is enabled on this pin. When this pin is not

connected to a load, it will remain low.

For example, in the following figure, there is a switch with one end connected to P0 and the other end connected to VCC. When the switch is turned off, read P0 as low level. When the switch is turned on, read P0 as high level.



2.3 Status Introduction

1. In input mode: it is an IO level status indicator light. Black represents low level, green represents high level.
2. In output mode: it is an IO level control switch button. Black represents low level, green represents high level. Clicking it will flip the voltage level.

3. Modify the power on status

Used to modify the input/output status of IO when powered on. Refer to the description in the "Digital I/O" section for mode, internal resistance, and status options. After modification, click the write button. After the successful write dialog box pops up, re plug and unplug the USB cable to take effect.

上电状态修改 型号: RM1000 序号: 592516271

引脚名称	P0	P1	P2	P3	P4	P5	P6	P7
模式	输入	输入	输入	输入	输入	输入	输入	输入
内部电阻	上拉	上拉	上拉	上拉	上拉	上拉	上拉	上拉
状态	●	●	●	●	●	●	●	●

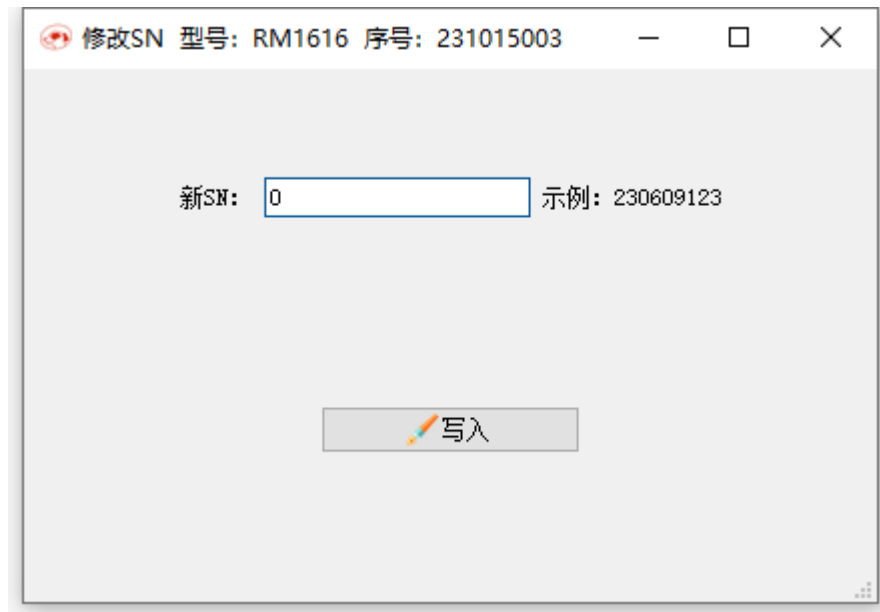
引脚名称	P8	P9	P10	P11	P12	P13	P14	P15
模式	输入	输入	输入	输入	输入	输入	输入	输入
内部电阻	上拉	上拉	上拉	上拉	上拉	上拉	上拉	上拉
状态	●	●	●	●	●	●	●	●

写入

恢复默认配置

4. Modify SN

First, fill in the new SN with a length range of 1-9 digits. After writing, click the write button. After the prompt is successful, return to the main program page and click the refresh button to refresh the device list.



八、 Introduction to Basic Programming Library Functions

We need to use two libraries, librockmong and libusb-1.0

This only introduces library functions in C language, similar to other languages. Please refer to the relevant routines for details.

1. Obtain the device

1. // Scan USB devices to obtain a list of device serial numbers.
2. // If the return value is greater than 0, it represents the number of devices obtained. If it is equal to 0, it means that the device has not been inserted. If it is less than 0, it means an error has occurred.
3. `int UsbDevice_Scan(int* SerialNumbers);`

2. Configure IO port input/output direction

1. // Initialize pin working mode
2. //SerialNumber: Equipment serial number


```
3. //Pin: Pin number. 0 P0. 1, P1...
4. //Mode: Input/output mode. 0, input. 1. Output. 2. Open leakage
5. //Pull: Pull up and pull down resistors. 0, none. 1. Enable internal pull-up.
   2. Enable internal pull-down
6. int IO_InitPin(int SerialNumber, int Pin, int Mode, int Pull);
```

3. Read input status

```
1. // Read pin status
2. //SerialNumber: Equipment serial number
3. //Pin: Pin number. 0 P0. 1, P1...
4. //PinState: Return pin status. 0, low level. 1. High level
5. // Function return: 0, normal< 0, Exception
6. int IO_ReadPin(int SerialNumber, int Pin, int *PinState);
```

4. Control output status

```
1. // Control pin output status
2. //SerialNumber: Equipment serial number
3. //Pin: Pin number. 0 P0. 1, P1...
4. //PinState: Return pin status. 0, low level. 1. High level
5. // Function return: 0, normal< 0, Exception
6. int IO_WritePin(int SerialNumber, int Pin, int PinState);
```

5. Read the output status

```
1. // Read the status of output pins
2. //SerialNumber: Equipment serial number
3. //Pin: Pin number. 0 P0. 1, P1...
4. //PinState: Return pin status. 0, low level. 1. High level
5. // Function return: 0, normal< 0, Exception
6. int IO_ReadOutputPin(int SerialNumber, int Pin, int *PinState);
```

九、 Advanced programming - operating multiple IOs simultaneously

Here is an introduction to advanced applications in C language. Similar to other languages, please refer to the relevant routines for details.

1. Initialize multiple IOs simultaneously

```
1. struct IO_InitStruct_Tx
2. {
3.     uint8_t Pin;
4.     uint8_t Mode;
5.     uint8_t Pull;
6. };
7. typedef struct IO_InitStruct_Tx IO_InitStruct_Tx_t;
8.
9. struct IO_InitStruct_Rx
10. {
11.     uint8_t Ret;
12. };
13. typedef struct IO_InitStruct_Rx IO_InitStruct_Rx_t;
14.
15. int IO_InitMultiPin(int SerialNumber, IO_InitStruct_Tx_t* TxStruct, IO_InitStruct_Rx_t* RxStruct, int Number);
```

2. Multiple IO reads simultaneously

```
1. struct IO_ReadStruct_Tx
2. {
3.     uint8_t Pin;
4. };
5. typedef struct IO_ReadStruct_Tx IO_ReadStruct_Tx_t;
6.
7. struct IO_ReadStruct_Rx
8. {
9.     uint8_t Ret;
10.    uint8_t PinState;
11. };
12. typedef struct IO_ReadStruct_Rx IO_ReadStruct_Rx_t;
13.
14. int IO_ReadMultiPin(int SerialNumber, IO_ReadStruct_Tx_t* TxStruct, IO_ReadStruct_Rx_t* RxStruct, int Number);
```

3. Multiple IO writes simultaneously

```
1. struct IO_WriteStruct_Tx
2. {
3.     uint8_t Pin;
```

```
4.     uint8_t PinState;
5. };
6. typedef struct IO_WriteStruct_Tx IO_WriteStruct_Tx_t;
7.
8. struct IO_WriteStruct_Rx
9. {
10.     uint8_t Ret;
11. };
12. typedef struct IO_WriteStruct_Rx IO_WriteStruct_Rx_t;
13.
14. int IO_WriteMultiPin(int SerialNumber, IO_WriteStruct_Tx_t* TxStruct, IO_WriteStruct_Rx_t* RxStruct, int Number);
```

4. Multiple output IOs simultaneously read status

```
1. struct IO_ReadOutput_TxStruct
2. {
3.     uint8_t Pin;
4. };
5. typedef struct IO_ReadOutput_TxStruct IO_ReadOutput_TxStruct_t;
6.
7. struct IO_ReadOutput_RxStruct
8. {
9.     uint8_t Ret;
10.    uint8_t PinState;
11. };
12. typedef struct IO_ReadOutput_RxStruct IO_ReadOutput_RxStruct_t;
13.
14. int IO_ReadMultiPin(int SerialNumber, IO_ReadStruct_Tx_t* TxStruct, IO_ReadStruct_Rx_t* RxStruct, int Number);
```