Modbus RTU 3 Color Signal Light Visual Alarm

for Workshop Machines and Industrial Equipment

Model No.: FN-VA520

Version: V1.0

1. Brief Introduction

FN-VA520 Modbus RTU 3-color signal light visual alarm, launched by Flyron Technology Co., Ltd., has the characteristics of low power consumption, long life, flexible installation, and convenience of use. This product adopts Modbus RTU standard communication protocol based on RS485 serial communication. According to different requirements, different light colors and flashing modes can be set easily. Except for workshop machines, this device can be used for all kinds of industrial equipment.

1.1. Features

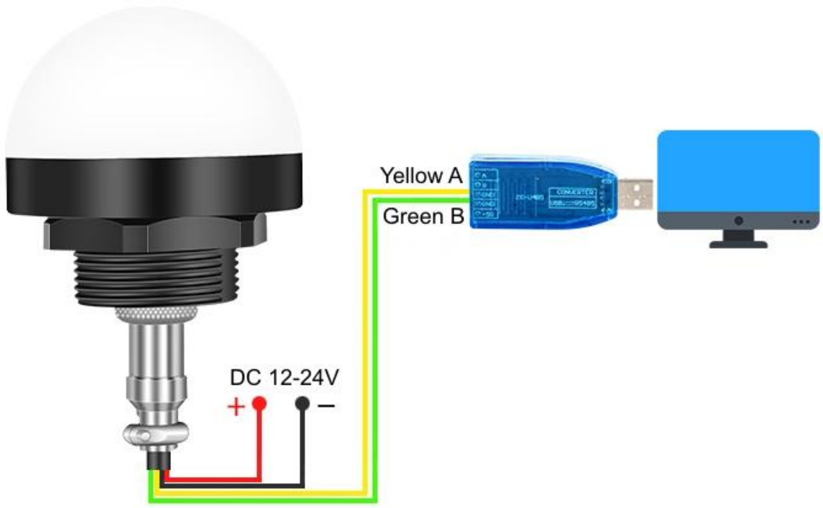
- ✧ Adopts Modbus RTU standard communication protocol.
- ✧ 3 color (red/yellow/green) signal light.
- ✧ Supports 3 types of flash modes (quick flash/slow flash/always on).
- ✧ Free to set the signal light to work in any color and any flash mode above.
- ✧ Able to set different communication baud rates (4800/9600/19200/38400/57600/115200/256000/35250).
- ✧ Semi-circular sphere design, beautiful and elegant.
- ✧ Simple to install and easy to use.

1.2. Technical parameters

Operating voltage	DC 12V-24V	Light color	red/yellow/green
Flash mode	quick flash/slow flash/always on	Protection class	IPX54

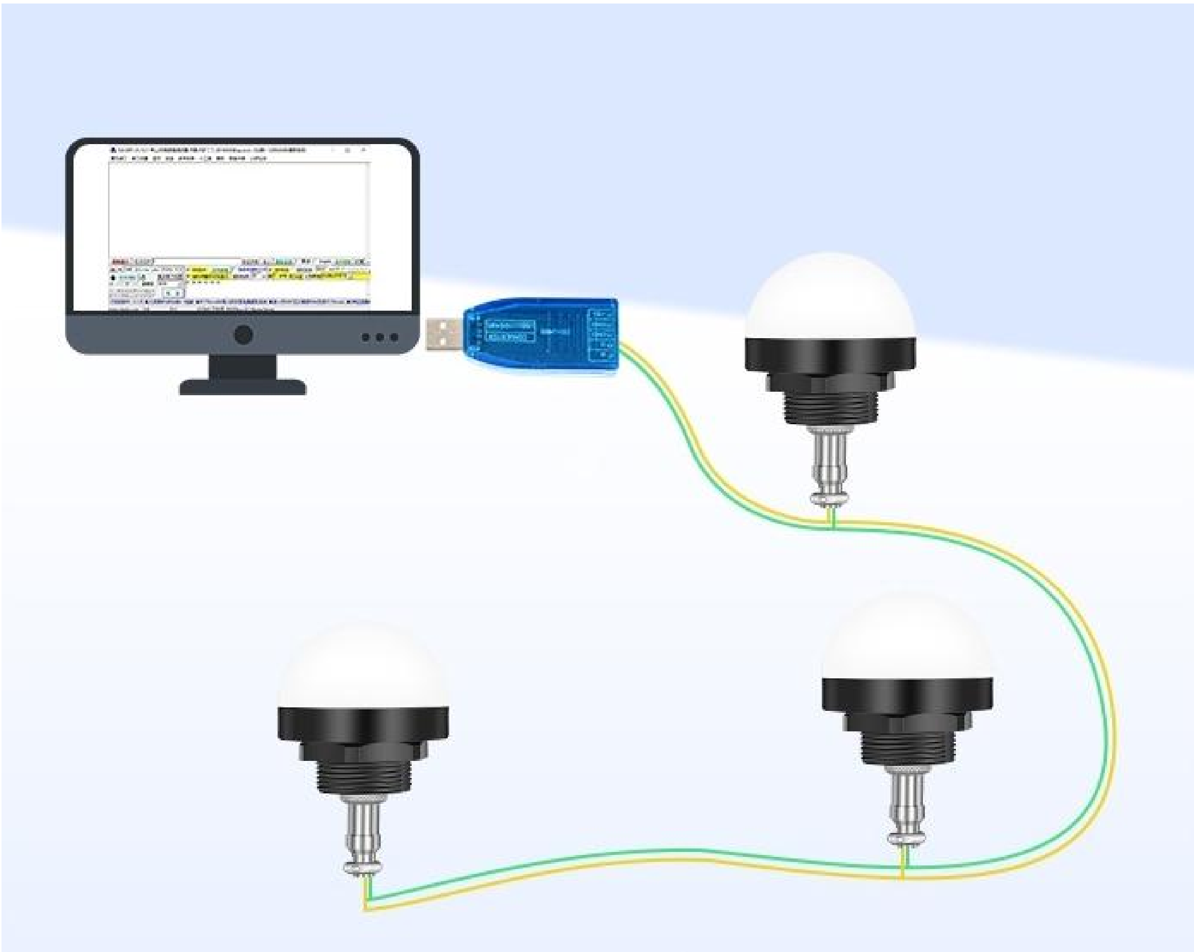
1.3. Dimensions

1.4. Wiring Example



This wiring example shows that the device is connected to a computer via a USB to RS485 converter. Running a serial debugging software on computer, users can debug and test the alarm easily before applying it to a workshop machine or an industrial equipment. When it is connected to a PLC or an industrial controller, in the same way, the yellow wire from the alarm is connected to the RS485 communication port A (+) of the PLC/industrial controller, while the green wire is connected to the RS485 communication port B (-).

If you need to control multiple slave devices, you can refer to the wiring example of RS485 as below.



2. Serial Communication

This product adopts Modbus RTU serial communication protocol based on RS485, and the default baud rate is 9600. Because of its binary representation and compact data structure, Modbus RTU has high communication efficiency and it is widely used in industrial occasions.

Factory default parameters for the device:

Default device address: 01

Default baud rate: 9600

RS485 wiring: Yellow A +/ Green B -

RS485 communication settings:

Baud rate: 9600 (default)

Data bits: 8

Stop bits: 1

Parity: None

Flow control: None

2.1. Communication format

Data bits: 8; Stop bits: 1; Error detection: CRC (Cyclic Redundancy Check)							
Addr	**Fun**	**Data start reg hi**	**Data start reg lo**	**Data #of regs hi**	**Data#of regs lo**	**CRC16_L**	**CRC16_H**
01	06	00	03	00	01	B8	0A
slave address	function code	register address high byte	register address low byte	register data high byte	register data low byte	check low byte	check high byte
Address code (1 byte), function code (1 byte), start address (2 bytes), data (2 bytes), check code (2 bytes)							

Addr: slave address (device address)

Fun: function code

Data start reg hi: data start address - register high byte

Data start reg lo: data start address - register low byte

Data #of reg hi: number of data read - register high byte

Data #of reg lo: number of data read - register low byte

CRC16_H: Cyclic redundancy check high byte

CRC16_L: Cyclic redundancy check low byte

Command format: [address code] + [function code] + [high 8 bits of register address] + [low 8 bits of register address] + [high 8 bits of data] + [low 8 bits of data] + [low 8 bits of check code] +[high 8 bits of check code]

2.1.1. Address code and function code

Address code:

The address field is at the beginning of the frame consisting of one byte, and the hexadecimal number is 0x00-0xFF. The decimal number is 0-255, of which 255 (0xFF) is our super address. These bits identify the user-specified address of the end device that will receive data from the host connected to it. The address of each terminal device must be unique, and only the addressed terminal will respond to queries containing this address. When the terminal sends back a response, the slave address data in the response tells the host which terminal is communicating with.

Function code:

The function field code tells the addressed terminal what function to perform. The following table lists the function codes that we commonly use, as well as their meanings and functions.

Function Code	Name	Function
03	Read holding register	To get the current binary value in one or more holding registers
06	Preset single register	To load a specific binary value into a holding register

03H function code: Read the data of the specified register. Our product uses this function to read the system status, the total number of files and other data, that is, the query function of our product.

06H function code: Write data into the register, that is, write the received data sent by the host into the register set by itself. The application of this function in our product is that after our chip receives the command sent by the host, the chip will store this command into the established register, and then perform corresponding actions on this instruction, that is, the control function of our product.

2.1.2. Error check field

This field allows the host and terminals to check for errors during transmission. Sometimes due to electrical noise and other interference, a set of data may change through the wire when it is transmitted from one device to another. Error checking can ensure that the host or terminal does not respond to the data that has changed during transmission. This improves the security and efficiency of the system, and the 16-bit cyclic redundancy method (CRC16) is used for error checking.

In the CRC operation, first a 16-bit register is preset to 0FFFFH (each bit is preset to 1), and then the 8 bits in each byte in the data frame are continuously operated with the current value of the register. Only the 8 data bits of each byte are involved in generating the CRC. The start bits, stop bits and the possible use of parity bits do not affect the CRC. When generating the CRC, the 8 bits of each byte are XORed with the contents of the register, and then the result is shifted to the lower bit. the upper bit is supplemented with "0", and the lowest bit (LSB) is shifted out and detected. If it is 1, the register performs an XOR operation with a preset fixed value (0A001H). If the lowest bit is 0, no processing is performed.

The above processing is repeated until the 8 shift operations are performed. When the last bit (the 8th bit) is shifted, the next 8-bit byte is XORed with the current value of the register. Above another 8 shift XOR operations are also performed in the same way. When all the bytes in the data frame are processed, the final value generated is the CRC value.

The above only explains the calculation method and function of the CRC16 cyclic redundancy check. If you don't understand it, you can ignore this part. Generally, we can hand this check code to the software for calculation, and you don't need to calculate it yourself. It doesn't matter if you don't understand it, please refer to the relevant information for details.

3. Detailed Explanation of Serial Commands

Below we describe some commonly used commands in detail.

The command format is

address code + function code + start address high byte+ start address low byte + data high byte + data low byte + CRC check

FF 06 00 CMD DH DL CRC_L CRC_H

FF is the super address, and the default device address is 01. CMD is the operation code. CRC_L and CRC_H are the cyclic redundancy check.

After sending a control command to the device, it'll return a set of same data immediately.

In the following commands, DH and DL are input values . Users enter the corresponding values according to actual requirements. Note that the command values are all in hex.

3.1. Detailed Explanation of some control commands

3.1.1. Set baud rate (0x0B)

1). The default baud rate is 9600. If you need to change the baud rate, you can use the command 0x0B to modify it.
2). After setting the baud rate, please wait for 1 second, and then send the reset command 0x0C, or power off and restart to take effect.
3). After setting the baud rate, the device will remember it, and when the device is restarted, the baud rate will become the set one.

Command format : FF 06 00 **0B** 00 DL CRC_L CRC_H (DL is the parameter of the baud rate to be set. See the table below for details)

DL parameters	Corresponding baud rate	DL parameters	Corresponding baud rate
0x01	9600	0x05	115200
0x02	19200	0x06	256000
0x03	38400	0x07	35250
0x04	57600	0x09	4800

Example: Send the command “01 06 00 0B 00 02 79 C9” or “FF 06 00 0B 00 02 6C 17” to set the baud rate to 19200 (DL=0x02)

Reference commands:

Command (xx xx represent CRC)	Function
FF 06 00 **0B** 00 01 xx xx	Set the baud rate to 9600
FF 06 00 **0B** 00 05 xx xx	Set the baud rate to 115200

3.1.2. Set the device address (0xC0)

Command format: FF 06 00 C0 00 DL CRC_L CRC_H

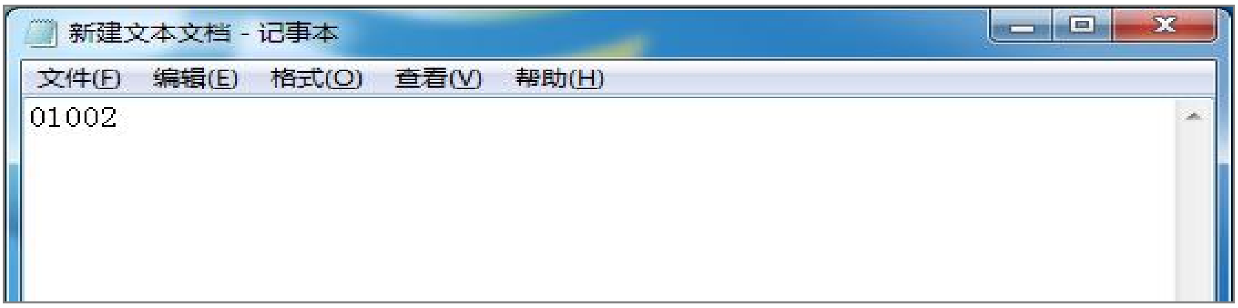
DL represents the device address that needs to be set. It can be set to 1-254 (0x01-0xFE in hex).

Example: Send the command “01 06 00 C0 00 02 08 37” or “FF 06 00 C0 00 02 1D E9” to set the device address to 02.

Reference commands:

Command (xx xx represent CRC)	Function
FF 06 00 **C0** 00 01 xx xx	Set the device address to 01
FF 06 00 **C0** 00 63 xx xx	Set the device address to 99
FF 06 00 **C0** 00 F7 xx xx	Set the device address to 247

1). After sending the command, the device address will take effect immediately and can be remembered when the power is turned off.
2). The address setting range is 1-254, so the value range is 0x01-0xFE for the DL.
3). The function of setting the address is that multiple devices can be connected to the RS485 bus, which is equivalent to giving each device a unique name, so that each device can be controlled individually. For details, you can search for the principle of RS485.
4). This product also supports setting the device address through a configuration file (text file), and the configuration file takes priority. For details, please refer to the details as below.



As you can see there are 5 digits in the configuration file (it's a text file). The first digit "0" represents the trigger mode in key control mode, the second and third digits "10" represent the volume setting (00-30 can be set here), and the fourth and fifth digits "02" are the device address (01-99 can be set here).

Notes:

a). When you create a configuration file like this, you don't need to care the first digit, because it's purposed for key control mode, but it must exist.

b). This configuration file must be placed on the root directory of the memory.

3.1.3. Control working status of signal light (0xC2)

Command format: FF 06 00 C2 00 XY CRC_L CRC_H

X represents the output mode and Y represents the color of the light.

Values that can be used for X	Corresponding function	Values that can be used for Y	Corresponding function
1	Always on	1	Works in red light
2	Slow flash	2	Works in yellow light
3	Quick flash	3	Works in green light
		0	No light

1) After the device receives the related command, the signal light will always light up according to the command. If you need to turn off the signal light, you need to send an command to turn it off or send other commands to change the working status of the signal light.

2) When the value of Y is 0, the signal light is going to be turned off. For example, sending the command "FF 06 00 C2 00 60 3C 24" to turn off of the signal light without affecting the current audio playback.

Reference commands:

Command (xx xx represent CRC)	Function
FF 06 00 **C2** 00 11 xx xx	Signal light works in red and always on
FF 06 00 **C2** 00 12 xx xx	Signal light works in yellow and always on
FF 06 00 **C2** 00 13 xx xx	Signal light works in green and always on
FF 06 00 **C2** 00 21 xx xx	Signal light works in red and slow flash
FF 06 00 **C2** 00 22 xx xx	Signal light works in yellow and slow flash
FF 06 00 **C2** 00 23 xx xx	Signal light works in green and slow flash
FF 06 00 **C2** 00 31 xx xx	Signal light works in red and quick flash
FF 06 00 **C2** 00 32 xx xx	Signal light works in yellow and quick flash
FF 06 00 **C2** 00 33 xx xx	Signal light works in green and quick flash

FF 06 00 **C2** 00 60 xx xx	Signal light is off

3.2. Detailed explanation of some query commands

3.2.1. Query the current light status (0x70)

Send the command "FF 03 00 70 00 00 51 CF" to query the current light status.
Return data: "01 03 02 DH DL CRC_L CRC_H"

The returned data includes information such as the device address, and current status of signal light and it's color.

Let's take the returned data "01 03 02 00 12 38 49" as an example. The current device address is 01. DH=0 x00. DL=0X12, which means that the signal light lights up yellow and it's always on.

In the returned data, the address code indicates the current device address, DL represents the light status. For example, DL=0x11 means that the current light status is red and always on. The corresponding status of DL value is shown in the table as below.

DL value	Corresponding light status	DL value	Corresponding light status
11	Red and always on	23	Green and slow flash
12	Yellow and always on	31	Red and quick flash
13	Green and always on	32	Yellow and quick flash
21	Red and slow flash	33	Green and quick flash
22	Yellow and slow flash	10 / 20 / 30 / 60 /06... etc .	Signal light goes out
If the returned value is not the values listed above, the light status should be off. If the command sent does not set the light status, the returned data may not correspond to the actual light status. For setting the light status, please refer to 3.1.2.			

3.3. Command References

Command	Function
01 06 00 **C2** 00 11 xx xx	Control the device of address 01: the light works in red and always on
01 06 00 **C2** 00 12 xx xx	Control the device of address 01: the light works in yellow and always on
02 06 00 **C2** 00 13 xx xx	Control the device of address 02: the light works in green and always on
02 06 00 **C2** 00 21 xx xx	Control the device of address 02: the light works in red and slow flash
03 06 00 **C2** 00 22 xx xx	Control the device of address 03: the light works in yellow and slow flash
03 06 00 **C2** 00 23 xx xx	Control the device of address 03: the light works in green and slow flash
04 06 00 **C2** 00 31 xx xx	Control the device of address 04: the light works in red and quick flash
05 06 00 **C2** 00 32 xx xx	Control the device of address 05: the light works in yellow and quick flash
1A 06 00 **C2** 00 33 xx xx	Control the device of address 26: the light works in green and quick flash
63 06 00 **C2** 00 60 xx xx	Control the device of address 99: the light is off
FF 06 00 **C0** 00 63 xx xx	Set the device address to 99
FF 06 00 **0B** 00 03 xx xx	Set the baud rate to 38400
FF 03 00 70 00 00 xx xx	Query the current light status