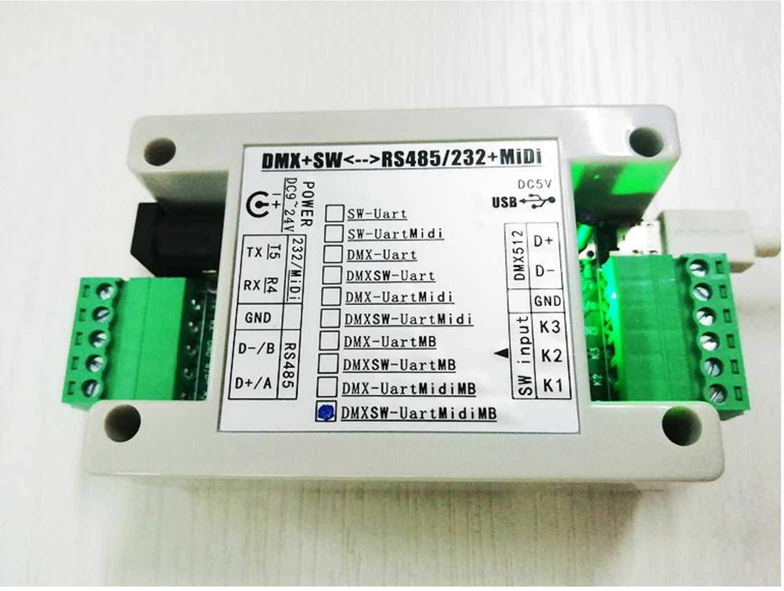
DMX512+Switch+UART+MIDIConversion Protocol

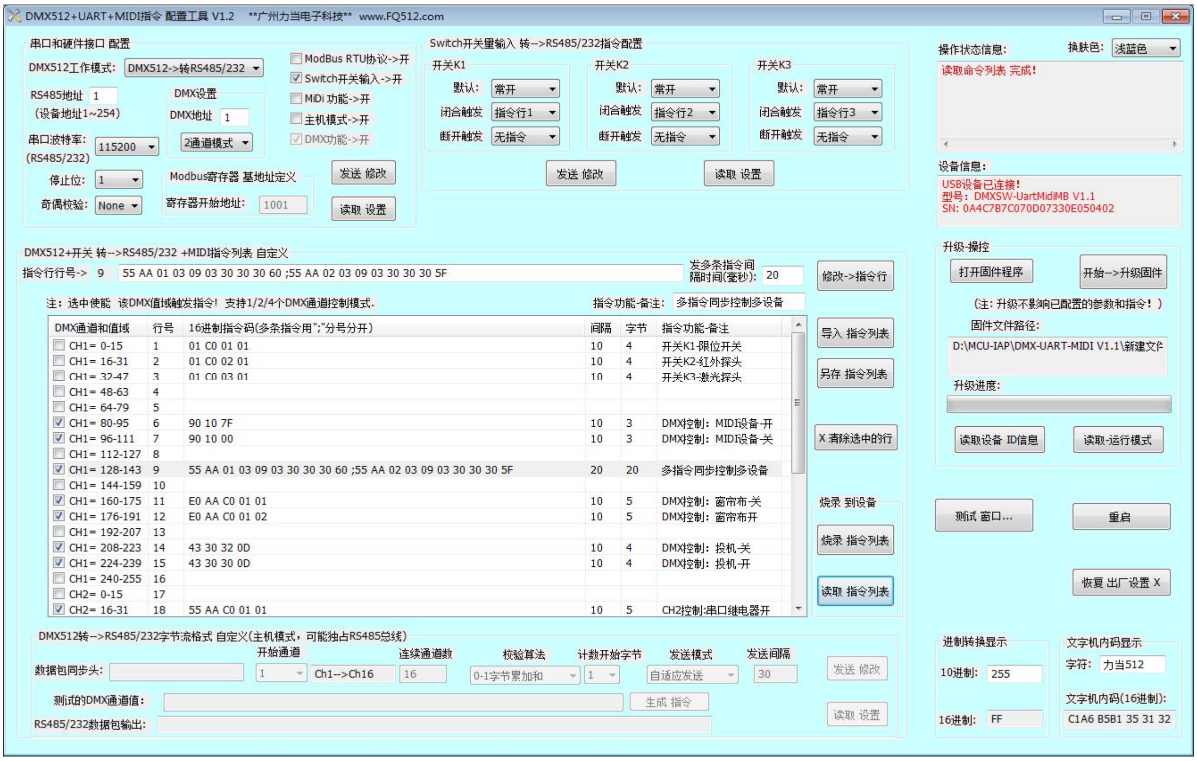
(Version:V1.0.2)

Table of contents

See photos for equipment appearance



"DMX512+UART+MIDI command configuration tool.exe" configuration interface



DMX512+UART+MIDI指令 配置工具 V1.2 **广州力当电子科技** www.FQ512.com

串口和硬件接口 配置

DMX512工作模式: DMX512->转RS485/232

RS485地址 1
(设备地址1~254)

串口波特率: 115200
(RS485/232)

停止位: 1

奇偶校验: None

DMX设置

DMX地址 1

2通道模式

Modbus寄存器 基地址定义

寄存器开始地址: 1001

□ ModBus RTU协议->开
☑ Switch开关输入->开
□ MiDi 功能->开
□ 主机模式->开
☑ DMX功能->开

Switch开关里输入 转-->RS485/232指令配置

	开关K1	开关K2	开关K3
默认:	常开	常开	常开
闭合触发	指令行1	指令行2	指令行3
断开触发	无指令	无指令	无指令

操作状态信息: 换肤色: 浅蓝色

读取命令列表 完成!

设备信息:

USB设备已连接!
型号: DMXSW-UartMidiMB V1.1
SN: 0A4C7B7C070D07330E050402

DMX512+开关 转-->RS485/232 +MIDI指令列表 自定义

指令行行号-> 9 55 AA 01 03 09 03 30 30 30 60 ;55 AA 02 03 09 03 30 30 30 5F

发多条指令间隔时间(毫秒): 20

注: 选中使能 该DMX值域触发指令! 支持1/2/4个DMX通道控制模式.

指令功能-备注: 多指令同步控制多设备

DMX通道和值域	行号	16进制指令码(多条指令用";"分号分开)	间隔	字节	指令功能-备注
□ CH1= 0-15	1	01 C0 01 01	10	4	开关K1-限位开关
□ CH1= 16-31	2	01 C0 02 01	10	4	开关K2-红外探头
□ CH1= 32-47	3	01 C0 03 01	10	4	开关K3-激光探头
□ CH1= 48-63	4				
□ CH1= 64-79	5				
☑ CH1= 80-95	6	90 10 7F	10	3	DMX控制: MIDI设备-开
☑ CH1= 96-111	7	90 10 00	10	3	DMX控制: MIDI设备-关
□ CH1= 112-127	8				
☑ CH1= 128-143	9	55 AA 01 03 09 03 30 30 30 60 ;55 AA 02 03 09 03 30 30 30 5F	20	20	多指令同步控制多设备
□ CH1= 144-159	10				
☑ CH1= 160-175	11	E0 AA C0 01 01	10	5	DMX控制: 窗帘布-关
☑ CH1= 176-191	12	E0 AA C0 01 02	10	5	DMX控制: 窗帘布开
□ CH1= 192-207	13				
☑ CH1= 208-223	14	43 30 32 0D	10	4	DMX控制: 投机-关
☑ CH1= 224-239	15	43 30 30 0D	10	4	DMX控制: 投机-开
□ CH1= 240-255	16				
□ CH2= 0-15	17				
☑ CH2= 16-31	18	55 AA C0 01 01	10	5	CH2控制:串口继电器开

升级-操控

(注: 升级不影响已配置的参数和指令!)

固件文件路径:

D:\MCU-IAP\DMX-UART-MIDI V1.1\新建文f

升级进度:

DMX512转-->RS485/232字节流格式 自定义(主机模式, 可能独占RS485总线)

数据包同步头:	开始通道		连续通道数	校验算法	计数开始字节	发送模式	发送间隔
	1	Ch1-->Ch16	16	0-1字节累加和	1	自适应发送	30

测试的DMX通道值:

RS485/232数据包输出:

进制转换显示

10进制: 255

16进制: FF

文字机内码显示

字符: 力当512

文字机内码(16进制): C1A6 B5B1 35 31 32

one,LiDBusAgreement Section

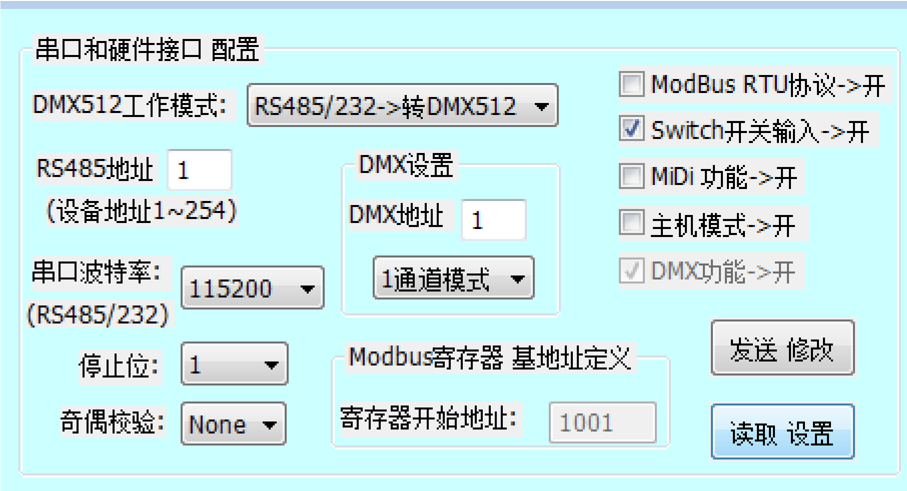
1.1 LiDBusProtocol packet format

Introduction to LiDBus Protocol

The LiDBus protocol is a protocol developed by Guangzhou Lidang Electronic Technology Co., Ltd.WWW.FQ512.COM) is a simple and efficient serial communication bus protocol defined for communication between various intelligent devices.

(1) Hardware serial port

In the default slave mode, the default address code = 1, useUARTport for communication,8bitData bits, no parity, 1 stop bit, default baud rate115200bps, communication parameters can be set through the configuration tool (as shown below). Support RS485 and RS232 mode transmission.



串口和硬件接口 配置

DMX512工作模式：RS485/232->转DMX512

RS485地址 1
（设备地址1~254）

串口波特率：115200
（RS485/232）

停止位：1

奇偶校验：None

DMX设置

DMX地址 1

1通道模式

Modbus寄存器 基地址定义

寄存器开始地址：1001

□ ModBus RTU协议->开
☑ Switch开关输入->开
□ MiDi 功能->开
□ 主机模式->开
☑ DMX功能->开

(2) Package format

Any commandIn packetsTo transmit, each data packet begins with a synchronization header SOP:A5 AA This ensures the reliability of communication and the unambiguous nature of the transmitted data. The fields are defined as follows:

A5	AA	**Addr**	**Cmd**	**Flag**	**Len_H**	**Len_L**	**Data**[Len]...	**CS_H**	**CS_L**

List display analysis:

Function Fields Field	Function Name Name	Byte length Size Byte	Bit Bit	Functional Description Description
1	SOP: Synchronous header	2	-	SOP =0xA5, 0xAA
2	Addr: Device address	1	-	The default device address is Addr=0x01?
3	Cmd:Command opcode	1	-	Operation command Cmd=?
4	Flag: Control Flag	1	Bit7 Bit6 Bit5 Bit4 Bit3 Bit2 Bit1 Bit0	Control Flag Flag=? Bit7 (highest bit): 1 = request response; 0=no response required; Bit6: 1 = This packet is an acknowledgement packet; 0 = Not Response Packet Bit5: unused = 0; Bit4: Unused = 0; Bit3: Unused = 0; **Bit2, Bit1, Bit0: Indicates data packet check Verification calculation mode:** 0 = No checksum(The 16-bit check word is fixed to 0 = 0x00, 0x00); 1 = Start counting from the 3rd byte Addr The sum, excluding the check byte CS itself, is greater than 16-bit overflow discard 2 = Modbus CRC16 checksum, starting from the third byte Addr
5	Len:Data length	2	-	The following data length Len=?
6	Data[...]: data	. . .	-	Data content, depends on the instruction
7	CS: Check word	2		No check/16-bit check/CRC16 check

Byte stream explanation:

- **Synchronous header SOP:**fixedA5 AA(2 bytes)
- **Addr**: Slave address (1 byte), address range is 1~254; 0 is broadcast address; 255 is standby address and is not used
- **Cmd**: Command code byte (1 byte)
- **Flag**: Control flag byte (1 byte)Bit7,Bit6,Bit5,Bit4,Bit3,Bit2,Bit1,Bit0: Bit7 (highest bit): 1 = response required; 0 = no response required;

  Bit6: 1 = This packet is a response packet; 0 = This packet is not a response packet

  Bit5: unused = 0;

  Bit4: Unused = 0;

  Bit3: Unused = 0;

Bit2, Bit1, Bit0: Indicates the packet check calculation mode

0 = No checksum calculation(16-bit check word is fixed to 0 = 0x00, 0x00);

1 = Calculate the cumulative sum from the third byte Addr, excluding the check byte CS itself, and discard if it is greater than 16 bits. 2 = Modbus crc16 check

- **Len**：Data length (2 bytes), high byte first,Length range: 0~1050, the content is commanded byCmdCome and decide.
- **Len Data[] data**：The data packet contains Len bytes of data, the content is determined by the commandCmdCome and decide.
- **CS**：16-bit check code (2 bytes), high byte first, composed ofFlaglowest3The bit value specifies the check count mode: Flag：Bit2,Bit1,Bit0=indicates the data packet check count mode: 0 =no calculation check (16-bit check word is fixed to 0=0x00,0x00).

1 = Count from the 3rd byte AddrCumulative Sum, does not include the check byte CS itself (if it is greater than 16 bits, it will be discarded if it overflows)

For example: UINT16 Sum = (Addr + Cmd + Flag + Len_H + Len_L + Len bytes of Data) 2 =Modbus CRC16 checksum(CRC16 is counted from the third byte Addr, excluding the check byte CS itself). Other values = undefined for use.

(3) Communication Process

Any command in this protocol follows the following process:

- First, the host sends a command request packet.
- After receiving the command request packet, the slave completes the relevant operation and if a response is required, the device returns a response packet and returns the operation result to the host.Command codereturnAnd set the Flag byteBit6Setting =1 LogoyesResponse Packet, the length of the returned data is determined by CMD, and it is different for different CMDs;It is not a response packet of a read command. The 8th byte = operation status response code ASK (ASK=0 means operation OK; ASK=greater than 0 means operation error)

Not reading instructions**Response Packet**Field Function Description (Except for read instruction): 10 bytes

Function Fields Field	Function Name Name	Byte length Size Byte	Bit Bit	Functional Description Description
1	SOP: Synchronization header	2	-	0xA5, 0xAA
2	Addr: Device address	1	-	Device default address = 0x01?
3	Cmd: command operation code	1	-	Cmd
4	Flag: Control Flag	1		Control flag Flag =? (verification mode) Bit6=1
5	Len:Data length	2	-	0x0001
6	Data[1]: data	1	-	0x00=ASK_OK operation successful; Greater than 0 = Operation failed!
7	CS：Check word	2	-	=No check/16-bit check/CRC16 check?

Note:If the device is required to respond, the master device must wait for the slave device to return the response code before sending the next frame of data to the device, which will slow down the communication speed. If high-speed communication control is required, in order to increase the speed, it is best not to request a response and set the highest bit of the Flag byte, Bit7 = 0.

(4) Byte Order

Some fields in the data packet consist of multiple bytes.Big-endian mode, that is, the high byte comes first and the low byte comes last.

1.2 System-related instruction codes

(1) Restart device command:Cmd = 0x0A

Field function description:

Function Fields Field	Function Name Name	Byte length Size Byte	Bit Bit	Functional Description Description
1	SOP: Synchronization header	2	-	0xA5, 0xAA
2	Addr: Device address	1	-	Device default address = 0x01
3	Cmd: command operation code	1	-	0x0A
4	Flag: Control Flag	1		Control Flag =? (verification mode, answer no)
5	Len:Data length	2	-	0
6	Data[...]: data	. . .	-	
7	CS: Check word	2	-	=No check/16-bit check/CRC16 check?

For example, device address = 1, request response, 16-bit cumulative sum check

instruction:A5 AA 01 0A 81 00 00 00 8C

answer:A5 AA 01 0A 41 00 010000 4D

(2) Read device-related information instructions:Cmd = 0x0B

Field function description:

Function Fields Field	Function Name Name	Byte length Size Byte	Bit Bit	Functional Description Description
1	SOP: Synchronization header	2	-	0xA5, 0xAA
2	Addr: Device address	1	-	Device default address = 0x01
3	Cmd: command operation code	1	-	0x0B
4	Flag: Control Flag	1		Control Flag =? (verification mode, answer no)
5	Len:Data length	2	-	0
6	Data[...]: data	. . .	-	
7	CS: Check word	2		=No check/16-bit check/CRC16 check?

For example, device address = 1, request response, 16-bit cumulative sum check

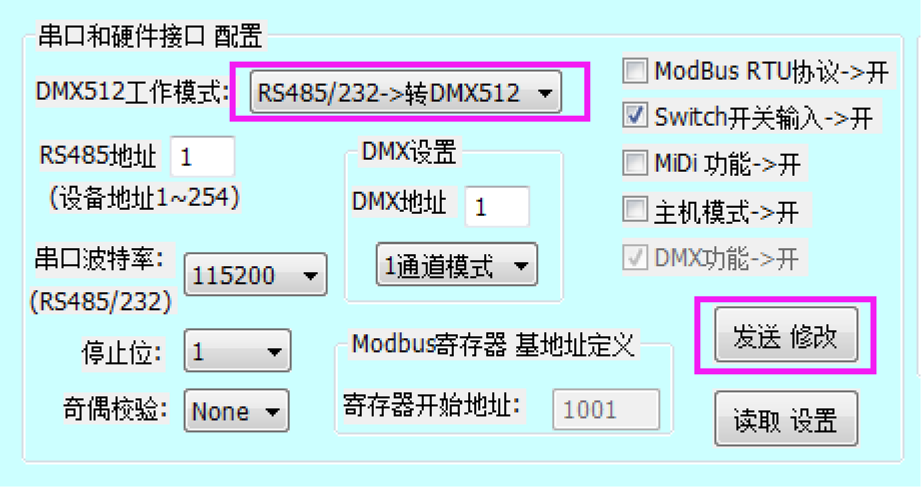
instruction:A5 AA 01 0B 81 00 00 00 8D The response is:twenty fourByte

serial number + device model + version number such as:A5 AA 01 0B

41 00 2B 18 30 41 34 43 37 42 37 43 30 37 30 44 30 37 33 33 30 45
30 35 30 34 30 32 10 44 4D 58 53 57 2D 55 61 72 74 4D 69 64 69 4D 42 0B 0B 3C

1.3 RS485/232-->convert to DMX512 command:

Set the working mode interface:



串口和硬件接口 配置

DMX512工作模式: RS485/232->转DMX512

RS485地址 1 (设备地址1~254)

串口波特率: (RS485/232) 115200

停止位: 1

奇偶校验: None

DMX设置

DMX地址 1

1通道模式

Modbus寄存器 基地址定义

寄存器开始地址: 1001

□ ModBus RTU协议->开

☑ Switch开关输入->开

□ MiDi 功能->开

□ 主机模式->开

☑ DMX功能->开

(1) All 512 DMX channels output the same value command:Cmd = 0x10

Field function description:

Function Fields Field	Function Name Name	Byte length Size Byte	Bit Bit	Functional Description Description
1	SOP: Synchronization header	2	-	0xA5, 0xAA
2	Addr: Device address	1	-	Device default address = 0x01
3	Cmd: command operation code	1	-	0x10
4	Flag: Control Flag	1		Control Flag =? (verification mode, answer no)
5	Len:Data length	2	-	0x0002
6	Data[2]: data	. . .	-	Specify DMX domain u = 0x00
				DMX ValuesChVal=?
7	CS: Check word	2		=No check/16-bit check/CRC16 check?

If the setting is 512 channels, all values are 255 = 0xFF, device address = 1, response is required, 16-bit accumulation and verification

instruction:A5 AA 011081 00 02 00 FF 01 93

answer:A5 AA 01 104100 0100 00 53

For example, if you set the 512 channel values to 0 = 0x00 = black, the device address = 1, and request a response, 16-bit accumulation and verification

instruction:A5 AA 011081 00 02 answer:00 00A5 00 94

AA 011041 00 010000 53

(2) Specify the starting DMX channel and the number of consecutive channels, and set the value of each channel:

Cmd=0x11

In order to increase the control rate and the amount of data is large, it is recommended to set no response! Only open verification! Field function description:

Function Fields Field	Function Name Name	Byte length Size Byte	Bit Bit	Function description and hexadecimal code Description
1	SOP: Synchronization header	2	-	0xA5, 0xAA
2	Addr: Device address	1	-	Device default address = 0x01
3	Cmd: command operation code	1	-	0x11
4	Flag: Control Flag	1		Control Flag =? (verification mode, answer no)
5	Len:Data length	2	-	Len=5+n
6	Data[5+n]: data	5+n	-	Specify DMX domain u = 0x00 (1 Byte)
				Start DMX channel = 1 to 512 (2 Byte)
				Number of continuous channelsn=1 to 512 (2 Byte)
				nDMX channel value = 0 to 255 (n Byte)
7	CS: Check word	2	-	=No check/16-bit check/CRC16 check?

If set8indivualDMXChannel value, start channel =1,8Channels are continuous8channels,Device address = 1, request response, 16-bit cumulative sum check

Note:Example DMX channel value CH1~CH8 =255,0,0 255,0,0, 0,255

Send command:A5 AA 01 118 1 00 0D 00 00 01 00 08FF 00 00 FF 00 00 00 FF03 A6

answer:A5 AA 01 11 41 00 0100 00 54

If set8indivualDMXChannel value, start channel =1,8Channels are continuous8channels,Device address = 1,No answer, 16-bit cumulative checksum

Send command:A5 AA 01 110100 0D 00 00 01 00 08FF 00 00 FF 00 00 00 FF03 26

answer:none

If set8indivualDMXChannel value, start channel =1,8Channels are continuous8channels,Device address = 1,No answer, No verification

Send command:A5 AA 01 110000 0D 00 00 01 00 08FF 00 00 FF 00 00 00 FF00 00

answer:none

If set8indivualDMXChannel value, start channel =1,8Channels are continuous8channels,Device address = 1,Request for response, CRC16 Modbus mode check

Send command:A5 AA 01 11 8 1 00 0D 00 00 01 00 08 FF 00 00 FF 00 00 00 FF E82D

answer:A5 AA 01 11 42 00 01 00 E8 21

If set8indivualDMXChannel value,Start Channel =9,8Channels are continuous8channels,Device address = 1, request response, 16-bit cumulative checksum

Note:Example DMX channel value CH9~CH16 =255,0,0 255,0,0, 0,255

Send command:A5 AA 01 118 1 00 0D 00 00 09 00 08FF 00 00 FF 00 00 00 FF03 AE 00 00

answer:A5 AA 01 11 41 00 01 54

like1Secondary settings512indivualDMXChannel value,Start Channel =1,512Channels are continuous512channels,Device address = 1, No response, 16-bit cumulative checksum

Send command:A5 AA 01 110 102 05 00 00 01 02 00512Byte channel value2ByteCS

(3) Set DMX channel value instructions according to the lighting model:Cmd = 0x16

(Can efficiently control DMX lights with multiple consecutive addresses)

Field function description:

Function Fields Field	Function Name Name	Byte length Size Byte	Bit Bit	Function description and hexadecimal code Description
1	SOP: Synchronization header	2	-	0xA5, 0xAA
2	Addr: Device address	1	-	Device default address = 0x01
3	Cmd: command operation code	1	-	0x16
4	Flag: Control Flag	1		Control Flag =? (verification mode, answer no)
5	Len:Data length	2	-	Len=5+n
6	Data[5+n]: data	5+n	-	Specify DMX domain u = 0x00 (1 Byte)
				Starting address code = 1 to 512 (2 Byte)
				Continuous lights c (1 Byte)
				The total number of channels n contained in the fixture (1 Byte)
				n DMX channel values (n Byte)
7	CS: Check word	2	-	=No check/16-bit check/CRC16 check?

For example, from1The channel starts to light up, that is, the starting address =1, continuous release170indivual3ChannelRGBLights, all lit red;set up

Backup address = 1,Request for response, 16-bit cumulative checksum

Note: Example RGB 3 channel values =255,0,0

170indivualRGBBright Red:

Send command:A5 AA 01 16 81 00 08 00 00 01 AA 03 FF 00 00 02 4D

Response code:A5 AA 01 16 41 00 01 00 00 59

170indivualRGBBright Green:

Send command:A5 AA 01 16 81 00 08 00 00 01 AA 03 00 FF 00 02 4D

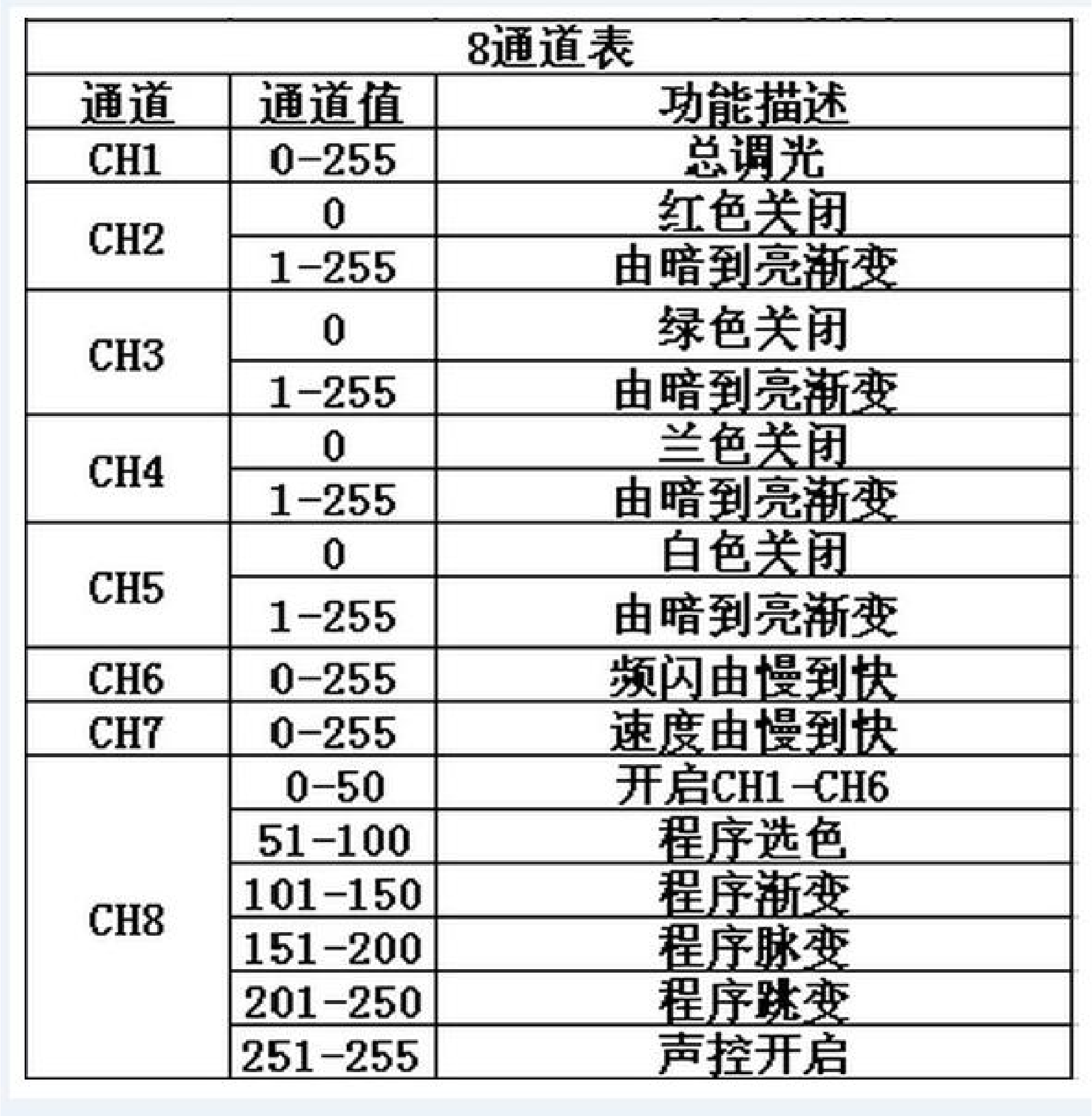
Response code:A5 AA 01 16 41 00 01 00 00 59

170indivualRGBBright Blue:

Send command:A5 AA 01 16 81 00 08 00 00 01 AA 03 00 00 FF 02 4D

Response code:A5 AA 01 16 41 00 01 00 00 59

Example: Control8Par lights in the corridor (similar to moving head lights) , the par light channel function table is as follows



8通道表		
通道	通道值	功能描述
CH1	0-255	总调光
CH2	0	红色关闭
	1-255	由暗到亮渐变
CH3	0	绿色关闭
	1-255	由暗到亮渐变
CH4	0	兰色关闭
	1-255	由暗到亮渐变
CH5	0	白色关闭
	1-255	由暗到亮渐变
CH6	0-255	频闪由慢到快
CH7	0-255	速度由慢到快
CH8	0-50	开启CH1-CH6
	51-100	程序选色
	101-150	程序渐变
	151-200	程序脉变
	201-250	程序跳变
	251-255	声控开启

If the scene6Par lamp, from1The channel starts to play continuously6Taiwan, that is,1The station address code is1; Current control6The lamps are all red, The brightest, then8The channel values are set like this:

CH1=255(total brightness),CH2=255(RRed is the brightest),CH3=0(Green Gate),CH4=0(Blue Pass),CH5=0(White Pass),

CH6=0(Strobe Off),CH7=0(Speed Off),CH8=0((Do not enable macro function)

Right now8Channel value10The base is:255, 255,0,0,0,0,0,0;16The base is:FF, FF, 00, 00,00, 00, 00, 00

Then from1The channel starts to light up, that is, the starting address =1, continuous release6indivual8ChannelPar Light, all bright red;

Device address = 1,Request for response, 16-bit cumulative checksum

Send command:A5 AA 01 16 81 00 0D 00 00 01 06 08 FF FF 00 00 00 00 00 0002 B2

Response code:A5 AA 011641 00 01 00 00 59

(4) According to the lighting model, set the DMX value of regular continuous or discontinuous intervals

instruction:Cmd=0x17

(Can efficiently control multiple DMX lights with regular continuous or certain interval channels)

Field function description:

Function Fields Field	Function Name Name	Byte length Size Byte	Bit Bit	Function description and hexadecimal code Description
1	SOP: Synchronization header	2	-	0xA5, 0xAA
2	Addr: Device address	1	-	Device default address = 0x01
3	Cmd: command operation code	1	-	0x17
4	Flag: Control Flag	1		Control Flag =? (verification mode, answer no)
5	Len:Data length	2	-	Len=6+n
6	Data[6+n]: data	6+n	-	Specify DMX domain u = 0x00 (1 Byte)
				Starting address code = 1 to 512 (2 Byte)
				Continuous lights c (1 Byte)
				Interval channel number t (uncontrolled) (1 Byte)
				Set the number of consecutive channels n (1 Byte)
				n DMX channel values (n Byte)
7	CS: Check word	2	-	=No check/16-bit check/CRC16 check?

Take controlling 3-channel RGB lights as an example:

Prerequisite: 170 RGB lights start from channel 1, that is, address code = 1; every 3 lights occupy 3 DMX aisle.

Now only the first 50 even-numbered lights are controlled to light up red:

That is, start from the second light, that is, the 4th channel, because each light occupies 3 channels, and the number of channels for even-numbered lights is 5, so only one red channel is needed;Device address = 1,Request for response, 16-bit cumulative sum check

Send command:A5 AA 01 17 8100 07 0000 043205 01FF01 DB
Answer code: A5 AA 01 17 41 00 01 00 00 5A

For example, if you only control the first 50 even-numbered lights to turn on green and turn off the red lights, it is equivalent to controlling only two consecutive channels. Device address = 1,Request for response, 16-bit cumulative checksum

Send command:A5 AA 01 17 8100 08 0000 0432 04 0200 FF01 DC 00 5A
Answer code: A5 AA 01 17 41 00 01 00

(3) Specify RGB light point control command (control of up to 170 RGB points):

Cmd=0x1A

(Very suitable for running water control and lamp post incremental dynamic control)

Field function description:

Function Fields Field	Function Name Name	Byte length Size Byte	Bit Bit	Function description and hexadecimal code Description
1	SOP: Synchronization header	2	-	0xA5, 0xAA
2	Addr: Device address	1	-	Device default address = 0x01
3	Cmd: command operation code	1	-	0x1A
4	Flag: Control Flag	1		Control Flag =? (verification mode, answer no)
5	Len:Data length	2	-	Len=5
6	Data[5]: data	5	-	Specify DMX domain u = 0x00 (1 Byte)
				Specify the RGB point to be controlled (1~170) (1 Byte)
				Red R(RGB order is subject to the actual light) (1 Byte)
				Green G (1 Byte)
				Blue B (1 Byte)
7	CS: Check word	2	-	=No check/16-bit check/CRC16 check?

If the specified control10indivualRGBLight up red,Device address = 1,Request for response, 16-bit cumulative checksum

Send command:A5 AA 01 1A 81 00 05 00 0A FF 00 00 01 AA 00
answer:A5 AA 01 1A 41 00 01 00 5D

If the specified control100indivualRGBLight up green,Device address = 1,Request for response, 16-bit cumulative checksum

Send command:A5 AA 01 1A 81 00 05 00 64 00 FF 00 02 04

answer:A5 AA 01 1A 41 00 01 00 00 5D

(6) RGB incremental control with directional background and foreground color instructions (up to 170 RGB

Point control) command:Cmd=0x1B

(Very suitable for running water control and lamp post incremental dynamic control)

Field function description:

Function Fields Field	Function Name Name	Byte length Size Byte	Bit Bit	Function description and hexadecimal code Description
1	SOP: Synchronization header	2	-	0xA5, 0xAA
2	Addr: Device address	1	-	Device default address = 0x01
3	Cmd: command operation code	1	-	0x1B
4	Flag: Control Flag	1		Control Flag =? (verification mode, answer no)
5	Len:Data length	2	-	Len=9
6	Data[9]: data	9	-	Specify DMX domain u = 0x00 (1 Byte)
				Specify the color of the first n lights (1 Byte)
				Specify direction: 0 = forward, 1 = reverse (1 Byte)
				Specify the first n lights to be red R (1 Byte)
				Specify the first n lights to be green G (1 Byte)
				Specify the first n lights to be blue B (1 Byte)
				Other lights red R (1 Byte)
				Other lights green G (1 Byte)
				Other lights blue B (1 Byte)
7	CS: Check word	2	-	=No check/16-bit check/CRC16 check?

Before designation50The light is red, and the background color of other lights is dark purple. **redRGB=FF,0,0; Dark purpleRGB=0F,00,0F**

,Device address = 1,Request for response, 16-bit accumulation and checksum

Send command:A5 AA 011B81 answer:00 09 00 32 00 FF 00 00 0f 00 0f 01 F5

A5 AA 01 1B 41 00 01 00 00 5E

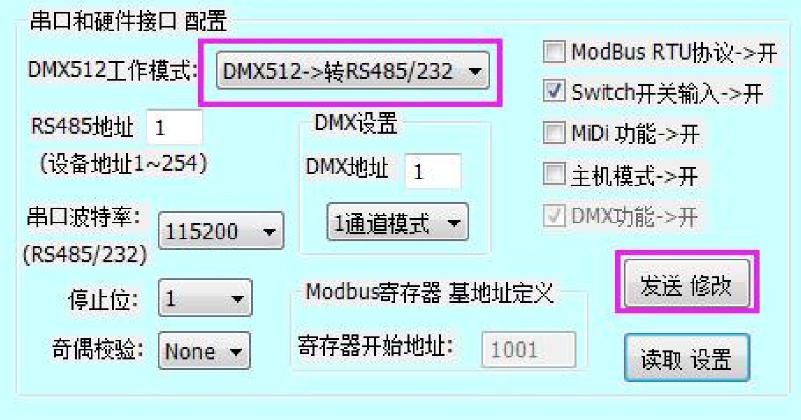
(7) Output recorded scene instructions:Cmd=0x1F **(Added in V1.3)**

Field function description:

Function Fields Field	Function Name Name	Byte length Size Byte	Bit Bit	Function description and hexadecimal code Description
1	SOP: Synchronization header	2	-	0xA5, 0xAA
2	Addr: Device address	1	-	Device default address = 0x01
3	Cmd: command operation code	1	-	0x1F
4	Flag: Control Flag	1		Control Flag =? (verification mode, answer no)
5	Len:Data length	2	-	Len=1
6	Data[1]: data	1	-	Scene ID = 1~6
7	CS: Check word	2	-	=No check/16-bit check/CRC16 check?

1.4 DMX512-->convert to RS485/232 command:

Set the working mode interface:



串口和硬件接口 配置

DMX512工作模式: DMX512->转RS485/232

RS485地址 1
(设备地址1~254)

串口波特率:
(RS485/232) 115200

停止位: 1

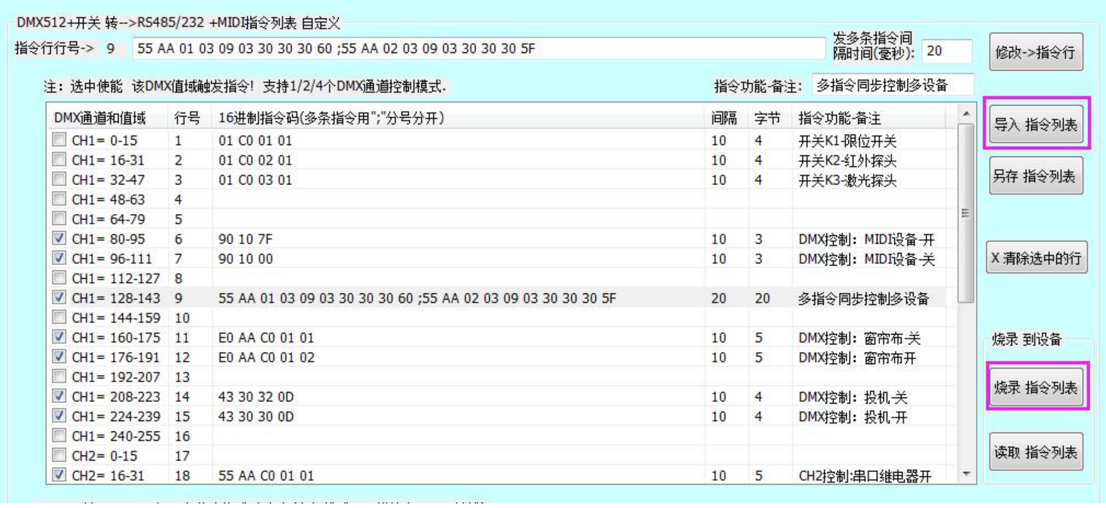
奇偶校验: None

DMX设置
DMX地址 1
1通道模式

Modbus寄存器 基地址定义
寄存器开始地址: 1001

□ ModBus RTU协议->开
☑ Switch开关输入->开
□ MiDi 功能->开
□ 主机模式->开
☑ DMX功能->开

（1）DMX512Channel value triggers customRS485/232instruction(Available"DMX512+Uart+MIDI Command configuration tool.exe"Software Toolsto edit and download custom instructions, as shown below).



DMX512+开关 转-->RS485/232 +MIDI指令列表 自定义

指令行行号-> 9 55 AA 01 03 09 03 30 30 30 60 ;55 AA 02 03 09 03 30 30 30 5F

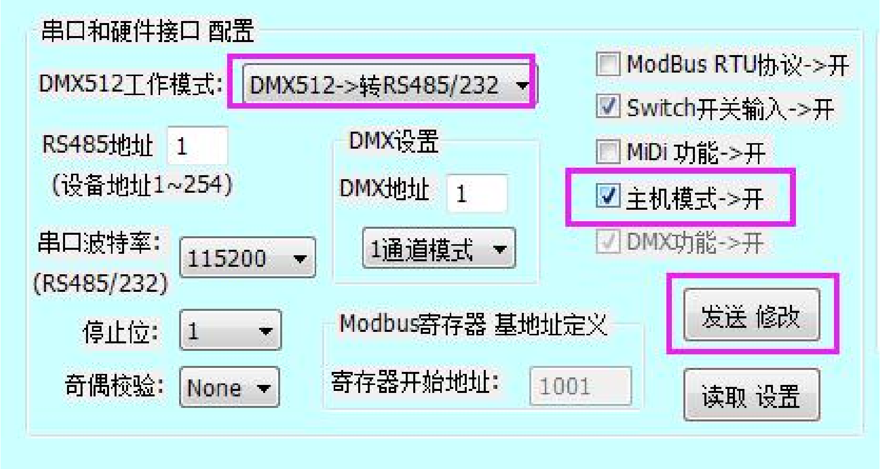
发多条指令间隔时间(毫秒): 20

注：选中使能 该DMX值域触发指令！ 支持1/2/4个DMX通道控制模式.

指令功能-备注: 多指令同步控制多设备

DMX通道和值域	行号	16进制指令码(多条指令用";"分号分开)	间隔	字节	指令功能-备注
□ CH1= 0-15	1	01 C0 01 01	10	4	开关K1-限位开关
□ CH1= 16-31	2	01 C0 02 01	10	4	开关K2-红外探头
□ CH1= 32-47	3	01 C0 03 01	10	4	开关K3-激光探头
□ CH1= 48-63	4				
□ CH1= 64-79	5				
☑ CH1= 80-95	6	90 10 7F	10	3	DMX控制：MIDI设备-开
☑ CH1= 96-111	7	90 10 00	10	3	DMX控制：MIDI设备-关
□ CH1= 112-127	8				
☑ CH1= 128-143	9	55 AA 01 03 09 03 30 30 30 60 ;55 AA 02 03 09 03 30 30 30 5F	20	20	多指令同步控制多设备
□ CH1= 144-159	10				
☑ CH1= 160-175	11	E0 AA C0 01 01	10	5	DMX控制：窗帘布-关
☑ CH1= 176-191	12	E0 AA C0 01 02	10	5	DMX控制：窗帘布开
□ CH1= 192-207	13				
☑ CH1= 208-223	14	43 30 32 0D	10	4	DMX控制：投机-关
☑ CH1= 224-239	15	43 30 30 0D	10	4	DMX控制：投机-开
□ CH1= 240-255	16				
□ CH2= 0-15	17				
☑ CH2= 16-31	18	55 AA C0 01 01	10	5	CH2控制:串口继电器开

(2)Configure as DMX512 ---" Convert to RS485/232 DMX512 channel value byte stream output mode（Configure as host mode), the output DMX512 channel byte stream format can be customized.



串口和硬件接口 配置

DMX512工作模式: DMX512->转RS485/232

RS485地址 1
(设备地址1~254)

串口波特率: 115200
(RS485/232)

停止位: 1

奇偶校验: None

DMX设置

DMX地址 1

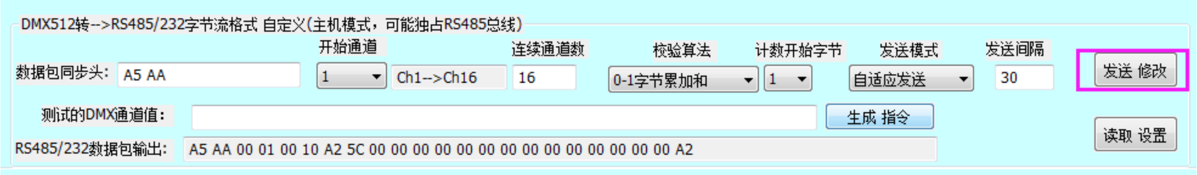
1通道模式

Modbus寄存器 基地址定义

寄存器开始地址: 1001

□ ModBus RTU协议->开
☑ Switch开关输入->开
□ MiDi 功能->开
☑ 主机模式->开
☑ DMX功能->开

Configure custom format DMX channel byte stream interface:



DMX512转-->RS485/232字节流格式 自定义(主机模式，可能独占RS485总线)

数据包同步头: A5 AA

开始通道: 1 Ch1-->Ch16

连续通道数: 16

校验算法: 0-1字节累加和

计数开始字节: 1

发送模式: 自适应发送

发送间隔: 30

测试的DMX通道值:

RS485/232数据包输出: A5 AA 00 01 00 10 A2 5C 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 A2

（3）ReadDMX512Channel value instructions:Cmd=0x21

Field function description:

Function Fields Field	Function Name Name	Byte length Size Byte	Bit Bit	Function description and hexadecimal code Description
1	SOP: Synchronization header	2	-	0xA5, 0xAA
2	Addr: Device address	1	-	Device default address = 0x01
3	Cmd: command operation code	1	-	0x21
4	Flag: Control Flag	1		Control Flag =? (verification mode, answer no)
5	Len:Data length	2	-	Len=5
6	Data[5]: data	n	-	Specify DMX domain u = 0x00 (1 Byte)
				Starting channel = 1 to 512 (2 Byte)
				Read the number of consecutive channels n (2 Byte)
7	CS: Check word	2	-	=No check/16-bit check/CRC16 check?

As from1Channel starts, reads8A continuousDMXChannel value,Device address = 1,Request for response, 16-bit accumulation and checksum

Send command:A5 AA 01 21 81 00 05 00 00 01 00 08 00 B1

Response Packet:A5 AA 01 214100 0D 00 00 01 00 0800 00 00 00 00 00 00 00 00 79

(return8indivualDMXChannel Value)

(4) QueryDMX512Input interface signal status command:Cmd=0x20

Field function description:

Function Fields Field	Function Name Name	Byte length Size Byte	Bit Bit	Function description and hexadecimal code Description
1	SOP: Synchronization header	2	-	0xA5, 0xAA
2	Addr: Device address	1	-	Device default address = 0x01
3	Cmd: command operation code	1	-	0x20
4	Flag: Control Flag	1		Control Flag =? (verification mode, answer no)
5	Len:Data length	2	-	Len=0
6	Data[0]: data	n	-	
7	CS: Check word	2	-	=No check/16-bit check/CRC16 check?

Device address = 1,Request for response, 16-bit accumulation and checksum

Send command:A5 AA 012081 00 00 00 A2

Response Packet:A5 AA 01 204100 010000 63:(00=DMXchangeRS485Input mode, nonedmxSignal)

answerField function description:

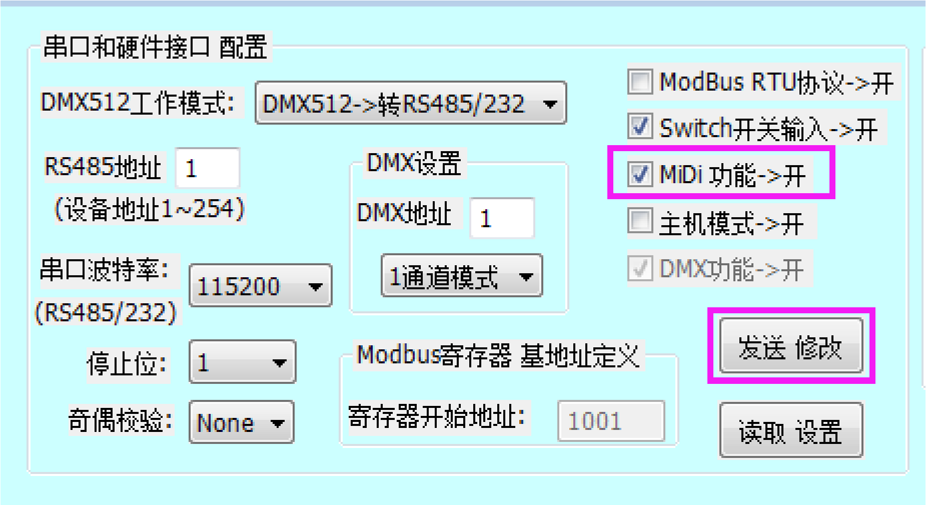
Function Fields Field	Function Name Name	Byte length Size Byte	Bit Bit	Function description and hexadecimal code Description
1	SOP: Synchronization header	2	-	0xA5, 0xAA
2	Addr: Device address	1	-	0x01
3	Cmd: command operation code	1	-	0x20
4	Flag: Control Flag	1		0x41 (Response packet, 16-bit accumulation and checksum)
5	Len:Data length	2	-	Len=0x0001

6	Data[1]: data	1	Bit7
			.
			.
			.
			Bit0
				Bit7 = DMX output/input mode: 0 = input; 1 = output
				Bit6=unused
				Bit5=unused
				Bit4=unused
				Bit3=unused
				Bit2=unused
				Bit1=unused
				Bit0 = DMX512 input status: 1 = signal; 0 = no signal

7	CS: Check word	2	-	=No check/16-bit check/CRC16 check?

1.5 RS485 to MIDI related command codes:

Set the MIDI function switch interface:



串口和硬件接口 配置

DMX512工作模式: DMX512->转RS485/232

RS485地址 1

(设备地址1~254)

串口波特率: 115200

(RS485/232)

停止位: 1

奇偶校验: None

DMX设置

DMX地址 1

1通道模式

Modbus寄存器 基地址定义

寄存器开始地址: 1001

☐ ModBus RTU协议->开

☑ Switch开关输入->开

☑ MiDi 功能->开

☐ 主机模式->开

☑ DMX功能->开

(1)RS485changeMIDIOutput instructions:Cmd=0x31

Function Fields Field	Function Name Name	Byte length Size Byte	Bit Bit	Function description and hexadecimal code Description
1	SOP: Synchronization header	2	-	0xA5, 0xAA
2	Addr: Device address	1	-	Device default address = 0x01
3	Cmd: command operation code	1	-	0x31
4	Flag: Control Flag	1		Control Flag =? (verification mode, answer no)
5	Len:Data length	2	-	Len=n bytes of MIDI instructions
6	Data[n]: data	n	-	n Byte
7	CS: Check word	2	-	=No check/16-bit check/CRC16 check?

For example, sending MIDI command 3 bytes: Node ON: 90 01 7E;Device address = 1, no verification, response required **Send command:A5 AA 01 318000 03 90 01 7E** answer: A5 AA 01 314000 01 **00 00**

00 00 73

Device address = 1,Request for response, 16-bit accumulation and checksum **The command to send is:A5 AA 01 318100 03 90 01 7E 01 C5 answer:A5 AA 01 31 41 00 01 00 00 74**

Device address = 1, CRC16 Modbus checksum, response required:

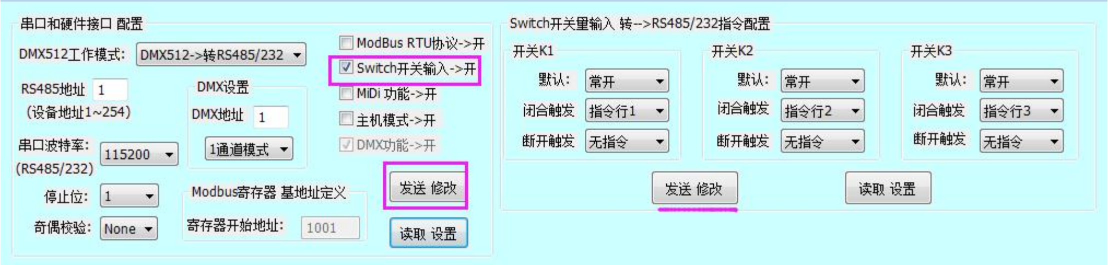
The command to send is:A5 AA 01 318200 03 90 01 7E 3E

answer:A5 AA 01 31 42 00 01 00 BB E9

Function Fields Field	Function Name Name	Byte length Size Byte	Bit Bit	Function description and hexadecimal code Description
1	SOP: Synchronization header	2	-	0xA5, 0xAA
2	Addr: Device address	1	-	Device default address = 0x01
3	Cmd: command operation code	1	-	0x28
4	Flag: Control Flag	1		Control Flag =? (verification mode, answer no)
5	Len:Data length	2	-	Len=0x00
6	Data[n]: data	0	-	0 Byte
7	CS: Check word	2	-	=No check/16-bit check/CRC16 check?

1.6 Read switch signal input instructions:Cmd=0x28

Set the switch function switch interface:



串口和硬件接口 配置

DMX512工作模式: DMX512->转RS485/232

RS485地址 1
(设备地址1~254)

串口波特率: (RS485/232) 115200

停止位: 1

奇偶校验: None

DMX设置

DMX地址 1

1通道模式

Modbus寄存器 基地址定义

寄存器开始地址: 1001

□ ModBus RTU协议->开
☑ Switch开关输入->开
□ MiDi 功能->开
□ 主机模式->开
☑ DMX功能->开

Switch开关量输入 转-->RS485/232指令配置

开关K1
默认: 常开
闭合触发 指令行1
断开触发 无指令

开关K2
默认: 常开
闭合触发 指令行2
断开触发 无指令

开关K3
默认: 常开
闭合触发 指令行3
断开触发 无指令

Field function description:

16-bit byte sum check, response required:

Send a read command:A5 AA 01 28 81 00 00 00 AA

answer:A5 AA 012841 00 0100 The reply fields 00 6B are as follows:

Function Fields Field	Function Name Name	Byte length Size Byte	Bit Bit	Function description and hexadecimal code Description
1	SOP: Synchronization header	2	-	0xA5, 0xAA
2	Addr: Device address	1	-	0x01
3	Cmd: command operation code	1	-	0x28
4	Flag: Control Flag	1		0x41 (Response packet, 16-bit accumulation and checksum)
5	Len:Data length	2	-	Len=0x0001

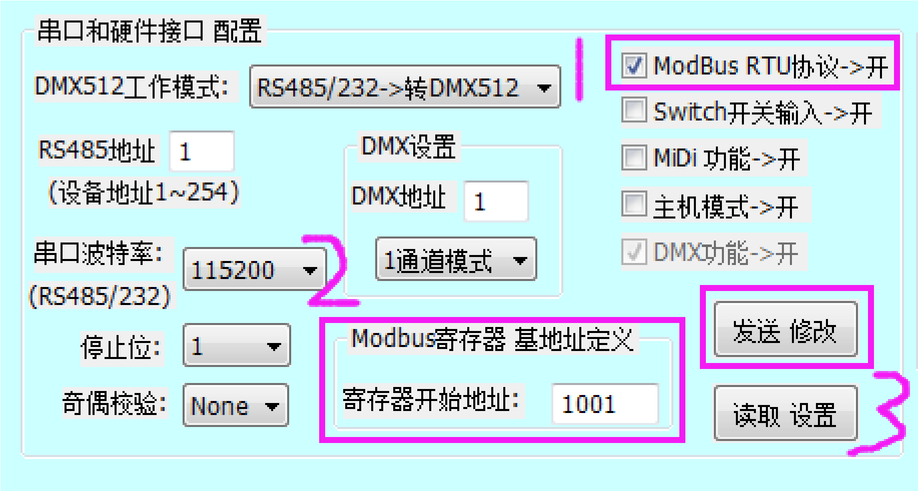
6	Data[1]: data	0	Bit7
			.
			.
			.
			Bit0
				Bit7=Unused
				Bit6=unused
				Bit5=unused
				Bit4=unused
				Bit3=unused
				Bit2 = K3 status: 1 = closed, 0 = open
				Bit1 = K2 status: 1 = closed, 0 = open
				Bit0 = K1 status: 1 = closed, 0 = open

7	CS: Check word	2	-	=No check/16-bit check/CRC16 check?

Four,ModBus protocol part

ModBus protocol function introduction

If it is a model with ModBus function, you need to enable the Modbus protocol, as shown in the following figure:



串口和硬件接口 配置

DMX512工作模式: RS485/232->转DMX512

RS485地址 1
(设备地址1~254)

串口波特率: 115200
(RS485/232)

停止位: 1

奇偶校验: None

DMX设置

DMX地址 1

1通道模式

☑ ModBus RTU协议->开

☐ Switch开关输入->开

☐ MiDi 功能->开

☐ 主机模式->开

☑ DMX功能->开

Modbus寄存器 基地址定义

寄存器开始地址: 1001

2.1 Important register address definition description

ModBus of this deviceBase Address(BaseAddr is used instead below) is customizable and configurable.

Device DefaultBase address BaseAddr = 1001(customizable settings), that is DMX512Channel 1 Corresponding register address. Each DMX channel value corresponds to a 16-bit ModBus register;

This device reserves the following address starting from the base address BaseAddr: AddrOffSet=30001024/2048 channels) for backup expansion of DMX channels.Function registerAddress from (BaseAddr+AddrOffSet=1001+3000=4001) starts to defineOffset Address.

(1) If the base address BaseAddr = 1001:

The register addresses corresponding to the 1st to 512th DMX channels are: 1001 to 1512 If the device has 1024 channels, the register addresses corresponding to the 1st to 512th DMX channels of the second port are: 1513 to 2024, and so on.

(2) Register offset address for other functionsFunAddr（3）The absolute register address formula of other functions is:

Register base address + offset address + function address offset =BaseAddr+AddrOffSet+ FunAddr =1001+3000 (fixed)+FunAddr =**4001+FunAddr**

2.2 Function register offset address and supported function code list:

sequence Number	Operational functions	Base Address BaseAddr Custom	Offset Address AddrOffSet =3000	Function address offset **FunAddr**	Supported Function code	Read and Write Function
1	RS485/232--> Direct DMX512	BaseAddr	0	0	0x10=16 0x06=6	Write
2	DMX512--> Direct transfer RS485/232	BaseAddr	0	0	0x03=3	read
3	Receive DMX512 signal status state	BaseAddr	AddrOffSet	**FunAddr=** 0 (Register 0)	0x03=3 0x04=4	read
4.	Switch signal status	BaseAddr	AddrOffSet	**FunAddr=** 10 (Register 10)	0x02=2	read
5	RS485/232--》 Convert MIDI	BaseAddr	AddrOffSet	**FunAddr=** 20 (Register 20~99)	0x10=16	Write
6	Set up 512 DMX Channel Value	BaseAddr	AddrOffSet	**FunAddr=** 100 (Registers 100 to 109)	0x10=16	Write
7	Set by lamp model DMX channel value	BaseAddr	AddrOffSet	**FunAddr=** 110 (Registers 110 to 149)	0x10=16	Write
8	Set regular discontinuity DMX channel value	BaseAddr	AddrOffSet	**FunAddr=** 150 (Registers 150~189)	0x10=16	Write
9	RGB light point control (RGB light strip only)	BaseAddr	AddrOffSet	**FunAddr=** 190 (Registers 190 to 199)	0x10=16	Write
10	RGB Increment Control (RGB light strip only)	BaseAddr	AddrOffSet	**FunAddr=** 200 (Register 200~209)	0x10=16	Write
11	Call scene number (Added in V1.3)	BaseAddr	AddrOffSet	**FunAddr=** 210 (Registers 210~219)	0x10=16 0x06=6	Write

2.3 Example of controlling DMX512 through Modbus protocol register reading and writing

(1) RS485/232 --> Direct to DMX512

Register address = BaseAddr + AddrOffSet +**FunAddr=** 1001+0+0=1001 Register address corresponding to channel 1=BaseAddr The 512th channel corresponds toRegister Address =BaseAddr+511

The register addresses corresponding to channels 1 to 512 are: BaseAddr~(BaseAddr+511) = 1001~1512

For example, for DMX channels 1 to 8Write value, then use 0x10=16 function code to write 8 register values: 8 channel values are: FF 02 00 00 00 00 00 08

The ModBus command to send is: 011003 E9 00 08 1000 FF 00 02 00 00 00 00 00 00 00 00 00 00 00 08D9 D4 (Start register address = 1001) answer:01 10 03 E9 00 08 10 7F

(2) DMX512--》 Direct to RS485/232

Register address = BaseAddr + AddrOffSet +**FunAddr=** 1001+0+0=1001

Register address corresponding to channel 1=BaseAddr The 512th

channel corresponds toRegister Address =BaseAddr+511

The register addresses corresponding to channels 1 to 512 are: BaseAddr~(BaseAddr+511) = 1001~1512

For example, read the DMX channels from 1 to 8value, then use 0x03=3 function code to read 8 register values:

Send ModBus command:01 03 03 E9 00 08 95 BC (Start register address = 1001)

Response: 01 03 1000 FF 00 02 00 00 00 00 00 00 00 00 00 00 00 085D 9D

8 channel values: FF 02 00 00 00 00 00 08

(3) Read the receiving DMX512 signal status

Register address = BaseAddr + AddrOffSet +**FunAddr=** 1001+3000+0=4001

Support function codes 0x03 and

0x04: Use function code 03 to read:

Send modbus command:01 03 0F A1 00 01 D6 FC

Answer: 01 03 0200 0179 84

(Note: 01 = indicates that there is a DMX signal in the input DMX mode; 00 = indicates that there is a DMX signal in the input DMX mode; 80 = indicates that it is the output DMX mode)

Use 04 function code to read:

Send modbus command:01 04 0F A1 00 01 63 3C

Answer: 01 04 020001 78 F0

(4) Read the switch signal status

Register address = BaseAddr + AddrOffSet +**FunAddr=** 1001+3000+10=4011 Support function code 0x02, read discrete signals:

Send modbus command:01 02 0F AB 00 03 4A FF

Response: 01 0201 00A1 88

(K3, K2, K1 correspond to Bit2, Bit1, Bit0 = 0x0001: indicating K1 is closed, K2, K3 is open)

(5) RS485/232--》 Convert to MIDI

Support function code 0x10=16;

Register address = BaseAddr + AddrOffSet +**FunAddr=** 1001+3000+20=4021

(Register 20~99, up to 80 bytes) For example, to send the Note on command: 90 01 7E; a total of 3 bytes are sent

Send modbus command: 01 10 03 E9 00 03 0600 90 00 01 00 7E2C 40

answer:01 10 03 E9 00 03 51 B8

(6) Set the channel values of 512 DMX

Register address = BaseAddr + AddrOffSet +**FunAddr=** 1001+3000+100=4101 Use function code 0x10=16, 2 registers:

Function Fields Field	Function Name Name	register Offset	Bit Bit	Function description and hexadecimal code Description
1	Specifying a DMX Domain	0	-	0x0000=0
2	DMX channel value	1	-	0~255

For example, if you set all 512 DMX channels to 255 or 0xFF,

send the modbus command:01 10 10 05 00 02 04 00 00 00 FF BE 10

answer:01 10 10 05 00 02 55 09

For example, if you want to set all 512 DMX channels to 0 or 0x00

(black field), send the modbus command:01 10 10 05 00 02 04 00 00 00 00 FE 50

answer:01 10 10 05 00 02 55 09

(7) Set DMX channel value according to the lighting model

Register address = BaseAddr + AddrOffSet +**FunAddr=** 1001+3000+110=4111

(Registers 110~149: control lights with up to 36 channels)

Function Fields Field	Function Name Name	register Offset	Bit Bit	Functional Description Description
1	Specifying a DMX Domain	0	-	0
2	Start channel, i.e., the light ground Address code	1	-	1~512
3	Continuous lights: c	2		1~512
4	Number of lamp channels: n	3		1~36
5	n DMX channel values	4~ n-1		n registers correspond to n DMX channel values of the lamp

For example, if you want to control 6 8-channel par lights to light up red, the address code of the first par light is 1, and the values of the 8 DMX channels are:FF FF 00 00 00 00 00 00

(Par light 8 channel functions: total brightness, R, G, B, W, strobe, speed, macro effect)

Send modbus command:

01 10 10 0F 00 0C 18 00 00 00 01 00 06 00 08 00 FF 00 FF 00 00 00 00 00 00 00 00 00 00 00 00 7D E6 Response code:01 10 10 0F 00 0C F4 C F

(8) Set regular discontinuous DMX channel values

Register address = BaseAddr + AddrOffSet +**FunAddr=** 1001+3000+150=4151

(Register 150~189): Control lights with up to 36 channels)

Function Fields Field	Function Name Name	register Offset	Bit Bit	Functional Description Description
1	Specifying a DMX Domain	0	-	0
2	Start channel, i.e., the light ground Address code	1	-	1~512
3	Continuous operands	2		1~512
4	Number of interval channels	3		1~512
5	Set the number of channels: n	4		1~35
6	n DMX channel values	5~ n-1		n registers correspond to n DMX channel values of the lamp

For example, 170 RGB lights start to light up from channel 1, that is, address code = 1; every 3 lights occupy 3 DMX channels. Now only the first 50 even-numbered lights are controlled to light up red:

That is to start from the second light, that is, the 4th channel, because each light occupies 3 channels, and the number of channels for even-numbered lights is 5, so only one red channel is needed;

Send modbus command:

01 10 10 37 00 06 0C 00 00 00 04 00 32 00 05 00 01 00 FF AD B7

Response code:01 10 10 37 00 06 F5 05

(9) RGB light point control (for RGB light strip only)

(Registers 190 to 199)

Register address = BaseAddr + AddrOffSet +**FunAddr=** 1001+3000+190=4191

Function Fields Field	Function Name Name	register Offset	Bit Bit	Functional Description Description
1	Specifying a DMX Domain	0	-	0
2	Specify the nth point control	1	-	1~170
3	Red R channel	2		0~255
4	Green G channel	3		0~255
5	Blue B channel	4		0~255

If the specified control10indivualRGBLight up red

Send modbus command:

01 10 10 5F 00 05 0A 00 00 00 0A00 FF 00 00 00 00F0 BF reply code:01 10 10 5F 00 05 34 D8

(10) RGB light strip increment control (for RGB light strip only)

(Register 200~209)

Register address = BaseAddr + AddrOffSet +**FunAddr=** 1001+3000+200=4201

Function Fields Field	Function Name Name	register Offset	Bit Bit	Functional Description Description
1	Specifying a DMX Domain	0	-	0
2	Number of control points n	1	-	1~170
3	direction	2		0~1
4	Foreground Red R	3		0~255
5	Foreground Green G	4		0~255
6	Foreground blue B	5		0~255
7	Background red	6		0~255
8	Background Green G	7		0~255
9	Background blue B	8		0~255

Before designation50The light is red, and the background color of other lights is dark purple.RGB=FF,0,0; Dark purple RGB=0F,00,0F

Send modbus command:

01 10 10 69 00 09 12 00 00 00 32 00 0000 FF 00 00 00 00 00 0F 00 00 00 0FE9 23

Response code:01 10 10 69 00 09 D4 D3

(11) Exporting saved scene effects(Added in V1.3)

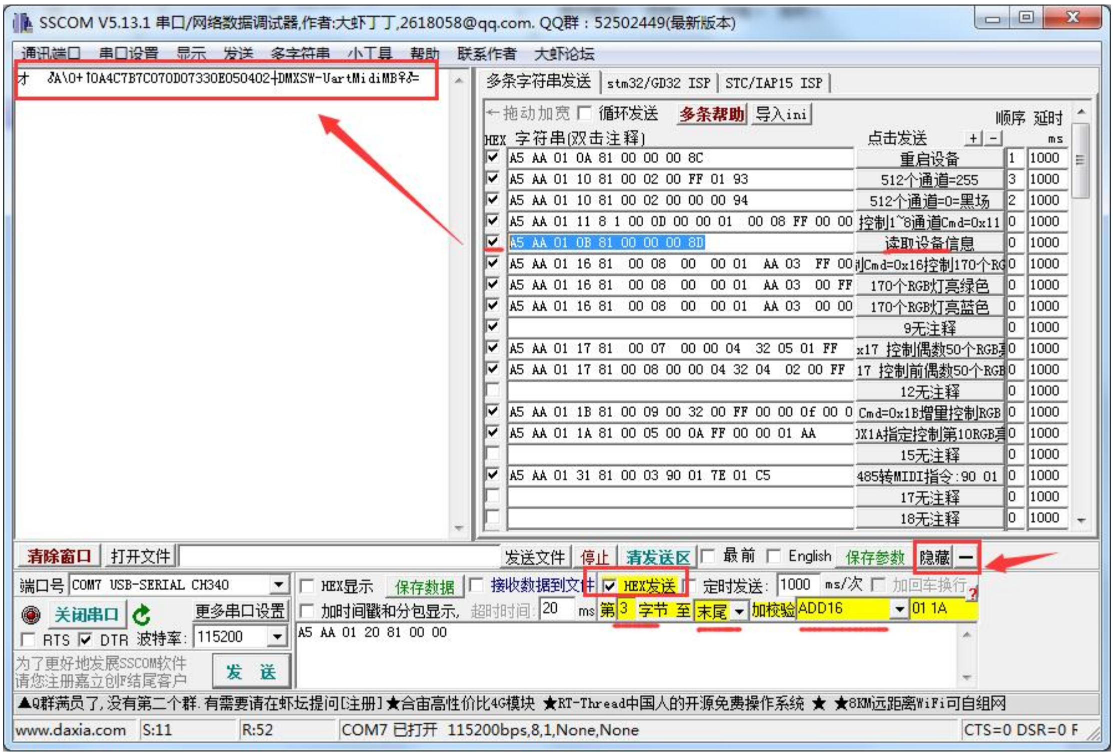
(Registers 210~219)

Register address = BaseAddr + AddrOffSet +**FunAddr=** 1001+3000+210=4211

Function Fields Field	Function Name Name	register Offset	Bit Bit	Functional Description Description
1	Scene number	0	-	Scene ID = 1~6

5. Refer to the attached picture of the debugging tool software interface

Refer to Figure 1 of the tool debugging assistant:



SSCOM V5.13.1 串口/网络数据调试器,作者:大虾丁丁,2618058@qq.com. QQ群：52502449(最新版本)

通讯端口 串口设置 显示 发送 多字符串 小工具 帮助 联系作者 大虾论坛

```
才  ♁A\0+↑0A4C7B7C070D07330E050402┼DMXSW-UartMidiMB♀♁=
```

多条字符串发送 | stm32/GD32 ISP | STC/IAP15 ISP
←拖动加宽 □ 循环发送 多条帮助 导入ini

HEX	字符串(双击注释)	点击发送	顺序	延时 ms
☑	A5 AA 01 0A 81 00 00 00 8C	重启设备	1	1000
☑	A5 AA 01 10 81 00 02 00 FF 01 93	512个通道=255	3	1000
☑	A5 AA 01 10 81 00 02 00 00 00 94	512个通道=0=黑场	2	1000
☑	A5 AA 01 11 8 1 00 0D 00 00 01 00 08 FF 00 00	控制1~8通道Cmd=0x11	0	1000
☑	A5 AA 01 0B 81 00 00 00 8D	读取设备信息	0	1000
☑	A5 AA 01 16 81 00 08 00 00 01 AA 03 FF 00	刂Cmd=0x16控制170个RG	0	1000
☑	A5 AA 01 16 81 00 08 00 00 01 AA 03 00 FF	170个RGB灯亮绿色	0	1000
☑	A5 AA 01 16 81 00 08 00 00 01 AA 03 00 00	170个RGB灯亮蓝色	0	1000
☑		9无注释	0	1000
☑	A5 AA 01 17 81 00 07 00 00 04 32 05 01 FF	x17 控制偶数50个RGB亮	0	1000
☑	A5 AA 01 17 81 00 08 00 00 04 32 04 02 00 FF	17 控制前偶数50个RGB	0	1000
□		12无注释	0	1000
☑	A5 AA 01 1B 81 00 09 00 32 00 FF 00 00 0f 00 0	Cmd=0x1B增量控制RGB	0	1000
☑	A5 AA 01 1A 81 00 05 00 0A FF 00 00 01 AA	0X1A指定控制第10RGB亮	0	1000
□		15无注释	0	1000
☑	A5 AA 01 31 81 00 03 90 01 7E 01 C5	485转MIDI指令:90 01	0	1000
□		17无注释	0	1000
□		18无注释	0	1000

清除窗口 打开文件 发送文件 停止 清发送区 □ 最前 □ English 保存参数 隐藏 —

端口号 COM7 USB-SERIAL CH340 □ HEX显示 保存数据 □ 接收数据到文件 ☑ HEX发送 □ 定时发送: 1000 ms/次 □ 加回车换行

关闭串口 更多串口设置 □ 加时间戳和分包显示, 超时时间: 20 ms 第 3 字节 至 末尾 加校验 ADD16 01 1A

□ RTS ☑ DTR 波特率: 115200

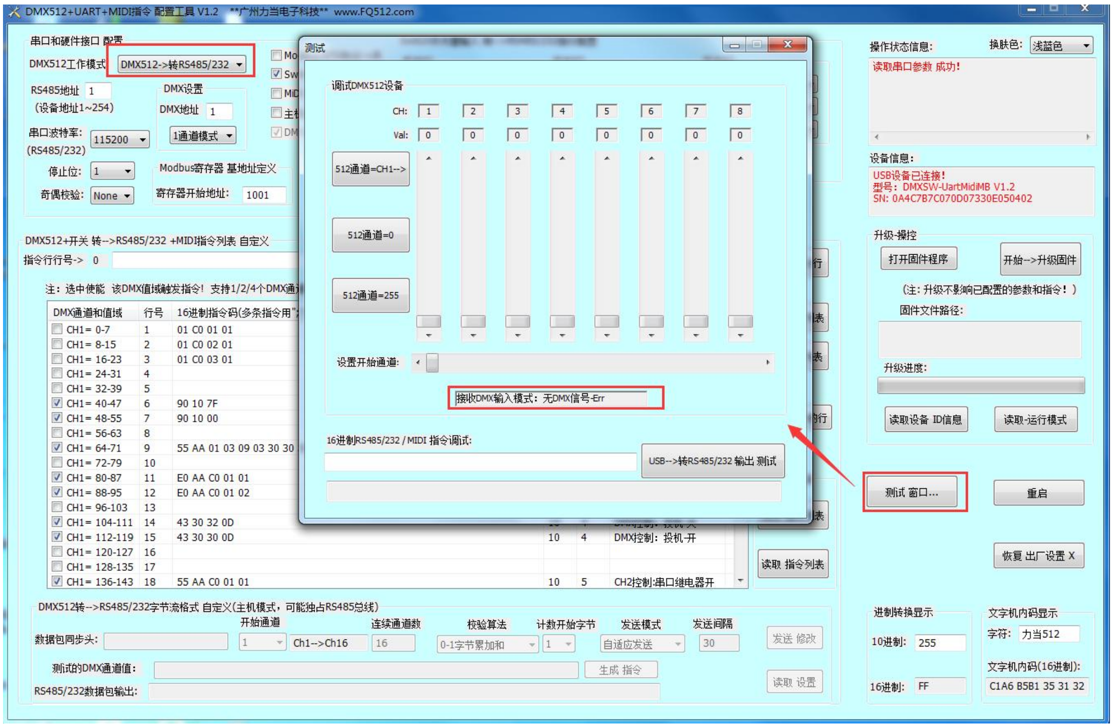
```
A5 AA 01 20 81 00 00
```

为了更好地发展SSCOM软件
请您注册嘉立创F结尾客户

▲Q群满员了,没有第二个群.有需要请在虾坛提问[注册]★合宙高性价比4G模块 ★RT-Thread中国人的开源免费操作系统 ★ ★8KM远距离WiFi可自组网

www.daxia.com S:11 R:52 COM7 已打开 115200bps,8,1,None,None CTS=0 DSR=0 F

DMX512--》Convert to RS485/232 Figure 2 of USB communication auxiliary monitoring interface:



DMX512+UART+MIDI指令 配置工具 V1.2 **广州力当电子科技** www.FQ512.com

串口和硬件接口 配置

DMX512工作模式 DMX512->转RS485/232

RS485地址 1 (设备地址1~254)

DMX设置 DMX地址 1 1通道模式

串口波特率: 115200 (RS485/232)

停止位: 1

奇偶校验: None

Modbus寄存器 基地址定义 寄存器开始地址: 1001

DMX512+开关 转-->RS485/232 +MIDI指令列表 自定义

指令行行号-> 0

注：选中使能 该DMX值域触发指令！ 支持1/2/4个DMX通

DMX通道和值域	行号	16进制指令码(多条指令用"			
□ CH1= 0-7	1	01 C0 01 01			
□ CH1= 8-15	2	01 C0 02 01			
□ CH1= 16-23	3	01 C0 03 01			
□ CH1= 24-31	4				
□ CH1= 32-39	5				
☑ CH1= 40-47	6	90 10 7F			
☑ CH1= 48-55	7	90 10 00			
□ CH1= 56-63	8				
☑ CH1= 64-71	9	55 AA 01 03 09 03 30 30			
□ CH1= 72-79	10				
☑ CH1= 80-87	11	E0 AA C0 01 01			
☑ CH1= 88-95	12	E0 AA C0 01 02			
□ CH1= 96-103	13				
☑ CH1= 104-111	14	43 30 32 0D			
☑ CH1= 112-119	15	43 30 30 0D	10	4	DMX控制：投机-开
□ CH1= 120-127	16				
□ CH1= 128-135	17				
☑ CH1= 136-143	18	55 AA C0 01 01	10	5	CH2控制:串口继电器开

DMX512转-->RS485/232字节流格式 自定义(主机模式，可能独占RS485总线)

数据包同步头: 开始通道 1 Ch1-->Ch16 连续通道数 16 校验算法 0-1字节累加和 计数开始字节 1 发送模式 自适应发送 发送间隔 30

测试的DMX通道值:

RS485/232数据包输出:

测试

调试DMX512设备

CH:	1	2	3	4	5	6	7	8
Val:	0	0	0	0	0	0	0	0

设置开始通道:

接收DMX输入模式：无DMX信号-Err

16进制RS485/232 / MIDI 指令调试:

操作状态信息: 换肤色: 浅蓝色

读取串口参数 成功!

设备信息:

USB设备已连接!
型号：DMXSW-UartMidiMB V1.2
SN: 0A4C7B7C070D07330E050402

升级-操控

(注：升级不影响已配置的参数和指令！)

固件文件路径:

升级进度:

进制转换显示

10进制: 255

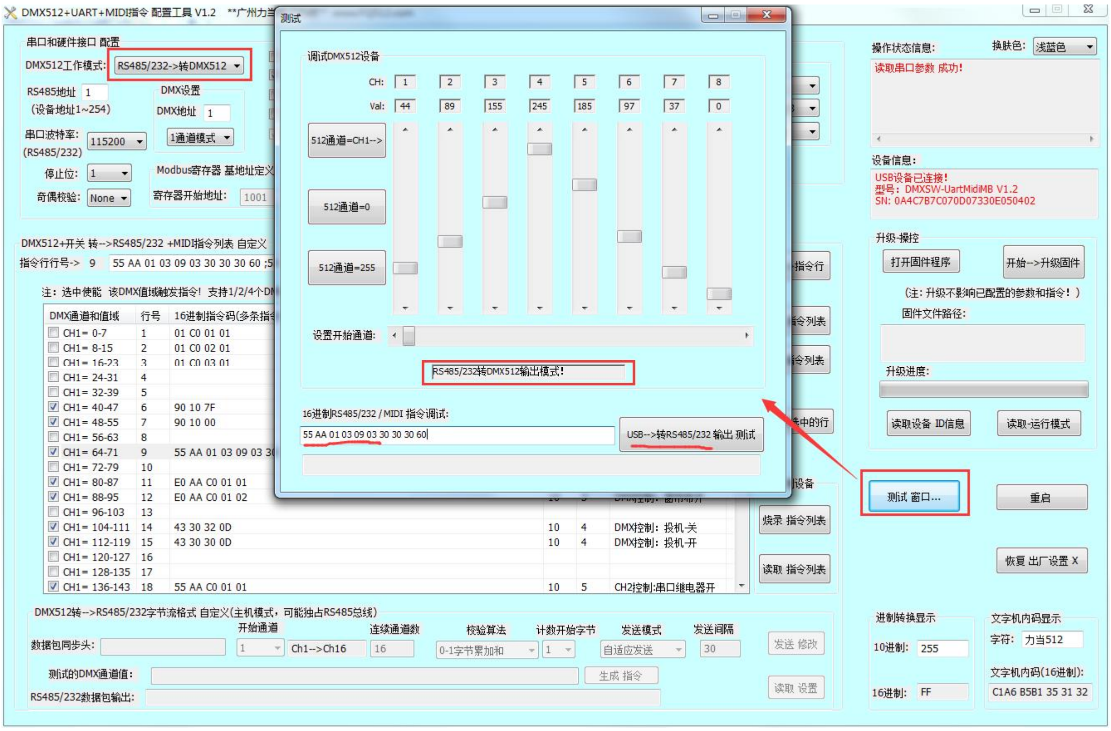
16进制: FF

文字机内码显示

字符: 力当512

文字机内码(16进制): C1A6 B5B1 35 31 32

DMX512 output control, serial port + MIDI send command, USB communication shop to help debug DMX512 and serial port + MIDI interface Figure 3:



DMX512+UART+MIDI指令 配置工具 V1.2 **广州力当

串口和硬件接口 配置

DMX512工作模式: RS485/232->转DMX512

RS485地址 1
(设备地址1~254)

DMX设置
DMX地址 1
1通道模式

串口波特率: (RS485/232) 115200

停止位: 1

奇偶校验: None

Modbus寄存器 基地址定义
寄存器开始地址: 1001

DMX512+开关 转-->RS485/232 +MIDI指令列表 自定义

指令行行号-> 9 55 AA 01 03 09 03 30 30 30 60 ;5

注: 选中使能 该DMX值域触发指令! 支持1/2/4个DM

DMX通道和值域	行号	16进制指令码(多条指令			
□ CH1= 0-7	1	01 C0 01 01			
□ CH1= 8-15	2	01 C0 02 01			
□ CH1= 16-23	3	01 C0 03 01			
□ CH1= 24-31	4				
□ CH1= 32-39	5				
☑ CH1= 40-47	6	90 10 7F			
☑ CH1= 48-55	7	90 10 00			
□ CH1= 56-63	8				
☑ CH1= 64-71	9	55 AA 01 03 09 03 30			
□ CH1= 72-79	10				
☑ CH1= 80-87	11	E0 AA C0 01 01			
☑ CH1= 88-95	12	E0 AA C0 01 02			
□ CH1= 96-103	13				
☑ CH1= 104-111	14	43 30 32 0D	10	4	DMX控制: 投机-关
☑ CH1= 112-119	15	43 30 30 0D	10	4	DMX控制: 投机-开
□ CH1= 120-127	16				
□ CH1= 128-135	17				
☑ CH1= 136-143	18	55 AA C0 01 01	10	5	CH2控制:串口继电器开

DMX512转-->RS485/232字节流格式 自定义(主机模式, 可能独占RS485总线)

数据包同步头:	开始通道		连续通道数	校验算法	计数开始字节	发送模式	发送间隔
	1	Ch1-->Ch16	16	0-1字节累加和	1	自适应发送	30

测试的DMX通道值:

RS485/232数据包输出:

测试

调试DMX512设备

CH:	1	2	3	4	5	6	7	8
Val:	44	89	155	245	185	97	37	0

设置开始通道:

RS485/232转DMX512输出模式!

16进制RS485/232 / MIDI 指令调试:

55 AA 01 03 09 03 30 30 30 60

操作状态信息: 换肤色: 浅蓝色

读取串口参数 成功!

设备信息:

USB设备已连接!
型号: DMXSW-UartMidiMB V1.2
SN: 0A4C7B7C070D07330E050402

升级-操控

(注: 升级不影响已配置的参数和指令!)

固件文件路径:

升级进度:

进制转换显示

10进制: 255

16进制: FF

文字机内码显示

字符: 力当512

文字机内码(16进制):

C1A6 B5B1 35 31 32

Technical service: Engineer Tan, Engineer Lai...

Tel: 189 0229 5001 /13418008536

Company website:www.FQ512.com

Guangzhou Lidang Electronic Technology Co., Ltd.