

HDMI 8x8 Matrix Switch RS-232 & IP Protocol

1. RS-232 Command:

Baudrate : 115200

Data width : 8bit
Parity: none
Stop: 1bit

Port switch command package length is 13byte:

[0xa5+0x5b+0x02+0x03+**input port(1~8)**+0x00+**output port(1~8)**+0x00+0x00+0x00+0x00+0x00+0x00+**checksum**]

All you need to change is just "input port", "output port", "checksum"

Checksum = 0x100 - (0xa5+0x5b+0x02+0x03+**input port**+0x00+**output port**+0x00+0x00+0x00+0x00)

For example: Set output 1 form input 8 command:

A5 5B 02 03 08 00 01 00 00 00 00 00 F2

All output switch to one input command package length is 13byte:

[0xa5+0x5b+0x02+0x02+**input port(1~8)**+0x00+0x00+0x00+0x00+0x00+0x00+0x00+**checksum**]

Port mapping status query package length is 13byte:

This is a query command which mean you must send query package and then receive an answer.

[0xa5+0x5b+0x02+0x01+**output port(1~8)**+0x00+0x00+0x00+0x00+0x00+0x00+0x00+**checksum**]

For example: Query output 1 input port (1~8)

Send package: A5 5B 02 01 **01** 00 00 00 00 00 00 00 FC

Receive package: A5 5B 02 01 01 00 **01** 00 00 00 00 00 FB

The red **01** mean the output port number, it should be 1~8.

The blue **01** mean the input port number, it should be 1~8.

Eddid set command package length is 13byte:

[0xa5+0x5b+0x03+0x02+**Eddid index(1~15)**+0x00+**input port(1~8)**+0x00+0x00+0x00+0x00+0x00+**checksum**]

Eddid index list:

SE_1080I_20	= 1
SE_1080I_51	= 2
SE_1080I_71	= 3
SE_1080P_20	= 4
SE_1080P_51	= 5
SE_1080P_71	= 6
SE_3D_20	= 7
SE_3D_51	= 8
SE_3D_71	= 9
SE_4K2K_20	= 10
SE_4K2K_51	= 11
SE_4K2K_71	= 12
SE_DVI_1024_768	= 13
SE_DVI_1920_1080	= 14
SE_DVI_1920_1200	= 15

Eddid copy command package length is 13byte:

[0xa5+0x5b+0x03+0x04+**output port (1~8)**+0x00+**input port(1~8)**+0x00+0x00+0x00+0x00+0x00+**checksum**]

Means: copy output port X edid to input port X

Power on/off command package length is 13byte:

[0xa5+0x5b+0x81+0x01+**power on/off(0x0f:ON; 0xf0:OFF)**+0x00+0x00+0x00+0x00+0x00+0x00+0x00+**checksum**]

Beep on/off command package length is 13byte:

[0xa5+0x5b+0x06+0x01+**Beep on/off(0x0f:ON; 0xf0:OFF)**+0x00+0x00+0x00+0x00+0x00+0x00+0x00+**checksum**]

2. IP Command:

Search Matrix MAC and IP:

1. Broadcast by UDP, the computer will broadcast a byte (any byte can be). the process port is 30600, broadcast IP is 255.255.255.255.
2. Matrix has received the computer's broadcast, will send own MAC, IP and some data to the computer, the computer will decode the data by the struct of Search.

Configure Matrix MAC and IP:

1. Broadcast by UDP, the computer will send 76 bytes (please reserve the next config struct). The process port is 30601, broadcast IP is 255.255.255.255.
2. Matrix received the computer broadcast, will send new MAC, IP which have been configured to the computer; the computer will decode the data by the next struct of config.

PC and Matrix control command:

1. if the PC connect to Matrix successfully, PC can send the command to Matrix (control command are the same as RS-232).

```
UDP_Search_port 30600
UDP_Config_port 30601
TCP_Data_port 8000
```

```
typedef struct
```

```
{
    BYTE MAC_Addr[6];                //for firmware to identify Matrix    6bytes
    //new IP address
    LWORd dwIPAddress ;              //4bytes

    //new gateway addr               //4bytes
    LWORd dwGWAddress ;

    //new netmask
    LWORd dwNMAddress ;              //4bytes

    //new dhcp
    BYTE dwDHCPEnable;               //1byte 1 = auto ip, 0 = Static ip

    //new mac address
    BYTE dwMacAddress[6];            //6bytes

    //1:set new ip address, 2:set new mac address, 4:reset , 0x08: set tftp server address,
    //0x10,set TAG
    BYTE dwModifyType;               //1byte
}
```

```

LWORD tftpServerIP ;           //4bytes
LWORD tftpIP ;                 //4bytes
LWORD tftpMask ;               //4bytes
LWORD tftpGate ;               //4bytes

BYTE Tag[32]; //TAG           //32bytes

BYTE receiveOK;                //1byte, 00=OK, FF=FAILED    msb 4bits: process
0=ok, 1=fail, 2=IP conflict; lsb 4bits: receive 0=ok, 1=fail.

BYTE checksum ;                //1byte, check sum, BYTE( sum all bytes )
}Config_ip_packet;

typedef struct
{
    BYTE MAC_Addr[6];           //Matrix MAC           6bytes
    LWORD IP_Address;           //Matrix IP             4bytes
    LWORD Gateway_address;      //Matrix gateway        4bytes
    LWORD netmask;              //Matrix netmask        4bytes
    WORD Data_Port;             //Matrix data port(30600) 2bytes
    WORD http_port;             //Matrix http port      2bytes
    BYTE Tag[32];               //TAG                   32bytes
    BYTE DHCP;                  //1byte
}Search_packet;

```

3. IR Code:

NEC CODE

#define SYSTEM_CODE 0x00

Function	Code
#define IR_KEY_POWER	0x14
#define IR_KEY_OUTPUT_1	0x47
#define IR_KEY_OUTPUT_2	0x07
#define IR_KEY_OUTPUT_3	0x40
#define IR_KEY_OUTPUT_4	0x02
#define IR_KEY_OUTPUT_5	0x18
#define IR_KEY_OUTPUT_6	0x44
#define IR_KEY_OUTPUT_7	0x0f
#define IR_KEY_OUTPUT_8	0x51
#define IR_KEY_OUTPUT_ALL	0x0a
#define IR_KEY_INPUT_1	0x19
#define IR_KEY_INPUT_2	0x1b
#define IR_KEY_INPUT_3	0x11
#define IR_KEY_INPUT_4	0x15
#define IR_KEY_INPUT_5	0x17
#define IR_KEY_INPUT_6	0x12
#define IR_KEY_INPUT_7	0x59
#define IR_KEY_INPUT_8	0x08
#define IR_KEY_INPUT_NEXT	0x48
#define IR_KEY_INPUT_PRE	0x55