Serial communication protocol specification

v1.2 (increase UART_TYPE)

1. Overview

This paper describes the specific data structure of the serial communication protocol.

2. Design requirements

2.1 Application scenarios

Data interaction between light embedded devices and module sensors.

2.2 functional requirements

1> The embedded device can correctly read the data generated by the module sensor through the serial port.

2> The embedded device can read the data of the module sensor at random (any time point).

3> The embedded device can actively request to read the data of the module sensor at a certain frequency.

3. Design environment

3.1 hardware platform

C3 control card, single chip sensor

3.2 software platform

C++ linux

4. Serial communication protocol

Description: The following structure definitions are in big endian

4.1 Protocol carrier package structure

4.2

Frame header(FH)	version number(V)	Data segment length (L)	Data segment(D)	Checksum(C)
2 byte	2 bytes	2 bytes	(6 ~ 1024) bytes	4 bytes

a. Frame header (FH): The value 0x0a0d. is used to identify the beginning of the packet.

b. Version number (V): Identifies the version of the current protocol. The current version number is 0x1000. After the protocol is upgraded, the corresponding changes will be

made. The version needs to be compatible, and the received packet is found to be higher Than the current version.

c. Data segment length (L): indicates the total number of bytes of the data segment (D).

d. Data segment (D): sensor packet

e. Checksum (C): Calculate the checksum using CRC-32. (Formula reference to Appendix 5.2)

4.3 Sensor packet structure

Length (L)	sensor type (T)	Command (C)	Data (D)
2 bytes	2 bytes	2 bytes	(0 ~1018) bytes

Description:

1> Length (L) L = L + T + C + D The number of bytes.

2> Sensor type (T) identifies different sensor types.

```
Const unsigned short LUMINANCE_TYPE = 1;              //Brightness sensor type
Const unsigned short TEMPERATURE_TYPE = 2;   //Temperature sensor type
Const unsigned short HUMIDITY_TYPE = 3;          //Humidity sensor type
Const unsigned short TELECONTROLLER_TYPE = 4;      //Remote control type
Const unsigned short TEMPERATURE2_TYPE = 5;  //Temperature 2 sensor type
Const unsigned short UART_TYPE = 6, // Serial port mapping type
Const unsigned short SCREEN_SWITCH_TYPE = 7, //Timer switch screen
Const unsigned short CMD_ASK_PM10 = 8, //PM10 concentration
Const unsigned short CMD_ASK_ PM2_5 = 9, //PM2.5 concentration
Const unsigned short CMD_ASK_ PM0_3_PM2_5 = 10, //PM0.3~PM2.5 concentration
Const unsigned short CMD_ASK_ PM2_5_PM10 = 11, //PM2.5~PM10 concentration
Const unsigned short RF_TYPE = 12, //RF module
Const unishged short SWITCH_PROGRAM_TYPE = 18, // Program switching
… …
```

3> Command (C) is defined by different sensor types

4> Data (D) is defined by different sensor types.

4.3.1 亮度(Luminance)

1> CMD Definition

```
Const unsigned short CMD_ASK_LUMINANCE = 0;
Const unsigned short CMD_ANSWER_LUMINANCE = 1;
```

2> data pack

a. The embedded device requests the current brightness value of the sensor..

Field	Length (L)	sensor type (T)	Command (C)
Value	6	LUMINANCE_TYPE	CMD_ASK_LUMINANCE

b. The sensor feeds back the current brightness value to the embedded device.

Field	Length (L)	sensor type (T)	Command (C)	Data (D)

Value	10	LUMINANCE_TYPE	CMD_ANSWER_LUMINANCE	Value(int32) lumen

Description: Value(int 32) 4 bytes, the upper 16 bits are the integer value of the luminance value, the lower 16 bits are the fractional part of the luminance value, and the illegal value of the brightness is 0x0.

4.3.2 Temperature

1> CMD Definition

Const unsigned short CMD_ASK_TEMPERATURE = 0;

Const unsigned short CMD_ANSWER_ TEMPERATURE = 1;

2> data pack

a. The embedded device requests the current temperature value of the sensor.

.

Field	Length (L)	sensor type (T)	Command (C)
Value	6	TEMPERATURE_TYPE	CMD_ASK_TEMPERATURE

b. The sensor feeds back the current temperature value to the embedded device..

Field 字段	Length (L)	sensor type (T)	Command (C)	Data (D)
Value 取值	10	TEMPERATURE_TYPE	CMD_ANSWER_ TEMPERATURE	Value(int32) Celsius

Description: Value(int 32) 4 bytes, the upper 16 bits are the temperature value integer, the lower 16 bits are the fractional part of the temperature value, and the illegal temperature value is 0x0.

4.3.3 Humidity

1> CMD Definition

Const unsigned short CMD_ASK_HUMIDITY = 0;

Const unsigned short CMD_ANSWER_HUMIDITY = 1;

Illegal value, decimal representation.

2> data pack

a. The embedded device requests the current humidity value of the sensor.

Field	Length (L)	sensor type (T)	Command (C)
Value	6	HUMIDITY_TYPE	CMD_ASK_HUMIDITY

b. The sensor feeds back the current humidity value to the embedded device.

Field	Length (L)	sensor type (T)	Command (C)	Data (D)
Value	10	HUMIDITY_TYPE	CMD_ANSWER_HUMIDITY	Value(int32) Relative humidity

Description: Value(int 32) 4 bytes, the upper 16 bits are the humidity value integer, the lower 16 bits are the decimal part of the humidity value, and the illegal humidity value is 0x0.

4.3.4 Telecontroller)

1> CMD Definition

The code value of the single chip decoding (refer to Appendix 5.1)

2> data pack

Fiel	Length (L)	sensor type (T)	Command (C)
Value	8	TELECONTROLLER_TYPE	MCU decoding code value

4.3.5 Temperature2

1> CMD Definition

Const unsigned short CMD_ASK_TEMPERATURE2 = 0;

Const unsigned short CMD_ANSWER_ TEMPERATURE2 = 1;

2> data pack

c. The embedded device requests the current temperature value of the sensor.

Field	Length (L)	sensor type (T)	Command (C)
Value	6	TEMPERATURE2_TYPE	CMD_ASK_TEMPERATURE2

d. The sensor feeds back the current temperature value to the embedded device.

Field	Length (L)	sensor type (T)	Command (C)	Data (D)
Value	10	TEMPERATURE2_TYPE	CMD_ANSWER_ TEMPERATURE2	Value(int32) Celsius

Description: Value(int 32) 4 bytes, the upper 16 bits are the temperature value integer, the lower 16 bits are the fractional part of the temperature value, and the illegal temperature value is 0x7f0000.

4.3.6 Serial port

1> CMD Definition

Const unsigned short CMD_DEFAULT = 0;

2> data pack

c. send data

Field	Length (L)	sensor type (T)	Command (C)	Data (d)
Value	indefinite	UART_TYPE	CMD_DEFAULT	Forward this data to the corresponding serial port

d. Feedback data

Field	Length (L)	sensor type (T)	Command (C)	Data (d)
Value	indefinite	UART_TYPE	CMD_DEFAULT	Return the data returned by the

				corresponding serial port back

4.3.7 Time switch screen (SCREEN_SWITCH_TYPE)

1> CMD Definition

```
Const unsigned short CMD_SET_TIME = 0;      //Set time
Const unsigned short CMD_GET_TIME = 1;      //Get Time
Const unsigned short CMD_SET_ON = 2;        //Set the screen to open
Const unsigned short CMD_SET_OFF = 3;       //Set off screen
```

2> Packet definition

A. Set the small board time

Field	Length (L)	sensor type (T)	Command (C)	Data (D)
Value	10	SCREEN_SWITCH_TYPE	CMD_SET_TIME	Value(int32)

Note: value is in seconds and is the total number of seconds at the current time distance of 00:00:00. For example, if the current time is set to 1 point, then the value is 3600.

B. Get small board time (reserved)

C. Set the screen opening time

Length (L)	sensor type (T)	Command (C)	Data (D)
10	SCREEN_SWITCH_TYPE	CMD_SET_ON	Value(int32)

Note: value is in seconds and is the total number of seconds at the current time distance of 00:00:00. For example, if the screen opening time is 1 point, then the value is 3600.

D. Set the screen off moment

Field	Length (L)	sensor type (T)	Command (C)	Data (D)
Value	10	SCREEN_SWITCH_TYPE	CMD_SET_OFF	Value(int32)

Note: value is in seconds and is the total number of seconds at the current time distance of 00:00:00. For example, if the set screen time is 1 point, then the value is 3600.

E. CMD_SET_TIME Feedback package

Field	Length (L)	sensor type (T)	Command (C)	Data (D)
Value	10	SCREEN_SWITCH_TYPE	CMD_SET_TIME	Value(int32)

Note: The value takes a value of 0. (reserved for use)

F. CMD_SET_ON Feedback package

Field	Length (L)	sensor type (T)	Command (C)	Data (D)
Value	10	SCREEN_SWITCH_TYPE	CMD_SET_ON	Value(int32)

Note: The value takes a value of 0. (reserved for use)

G. CMD_SET_OFF Feedback package

Field	Length (L)	sensor type (T)	Command (C)	Data (D)
Value	10	SCREEN_SWITCH_TYPE	CMD_SET_OFF	Value(int32)

Note: The value takes a value of 0. (reserved for use)

4.3.8 PM10

1> CMD Definition

Const unsigned short CMD_ASK_PM10 = 0;

Const unsigned short CMD_ANSWER_PM10 = 1;

2> data pack

c. The embedded device requests the current humidity value of the sensor

Field	Length (L)	sensor type (T)	Command (C)
Value	6	PM10_TYPE	CMD_ASK_PM10

d. The sensor feeds back the current humidity value to the embedded device.

Field	Length (L)	sensor type (T)	Command (C)	Data (D)
Value	10	PM10_TYPE	CMD_ANSWER_PM10	Value(int32) PM10Concentration

Description: Value(int 32) 4 bytes, the illegal value is 0xffffffff.

4.3.9 PM2.5 concentration

1> CMD Definition

Const unsigned short CMD_ASK_PM2_5 = 0;

Const unsigned short CMD_ANSWER_ PM2_5 = 1;

2> data pack

e. The embedded device requests the current humidity value of the sensor.

Field	Length (L)	sensor type (T)	Command (C)
Value	6	PM2_5_TYPE	CMD_ASK_ PM2_5

f. The sensor feeds back the current humidity value to the embedded device.

Field	Length (L)	sensor type (T)	Command (C)	Data (D)
Value	10	PM2_5_TYPE	CMD_ANSWER_ PM2_5	Value(int32) PM2.5Concentration

Description: Value(int 32) 4 bytes, the illegal value is 0xffffffff.

4.3.10 PM0.3~PM2.5 concentration

1> CMD Definition

Const unsigned short CMD_ASK_PM0_3_PM2_5 = 0;

Const unsigned short CMD_ANSWER_ PM0_3_PM2_5 = 1;

2> data pack

g. The embedded device requests the current humidity value of the sensor.

Field	Length (L)	sensor type (T)	Command (C)
Value	6	PM0_3_PM2_5_TYPE	CMD_ASK_ PM0_3_PM2_5

h. The sensor feeds back the current humidity value to the embedded device.

Field	length (L)	sensor type (T)	Command (C)	Data (D)
Value	10	PM0_3_PM2_5_TYPE	CMD_ANSWER_ PM0_3_PM2_5	Value(int32) PM0.3~PM2.5Concentration

Description: Value(int 32) 4 bytes, the illegal value is 0xffffffff.

4.3.11 PM2.5~PM10 concentration

1> CMD Definition

Const unsigned short CMD_ASK_ PM2_5_PM10_ = 0;

Const unsigned short CMD_ANSWER_ PM2_5_PM10 = 1;

2> data pack

i. The embedded device requests the current humidity value of the sensor

Field	Length (L)	sensor type (T)	Command (C)
Value	6	PM2_5_PM10_TYPE	CMD_ASK_ PM2_5_PM10

j. The sensor feeds back the current humidity value to the embedded device

Field	Length (L)	sensor type (T)	Command (C)	Data (D)
Value	10	PM2_5_PM10_TYPE	CMD_ANSWER_ PM2_5_PM10	Value(int32) PM2.5~PM10Concentration

Description: Value(int 32) 4 bytes, the illegal value is 0xffffffff.

4.3.12 RF module

1> CMD Definition

Const unsigned short CMD_ASK_STATUS = 0;

Const unsigned short CMD_ANSWER_STATUS = 1;

Const unsigned short CMD_SET_ADDRESS = 2;

Const unsigned short CMD_SEND_DATA = 3;

Const unsigned short CMD_RECV_DATA = 4;

Const unsigned short CMD_SET_CHANNEL = 5;

2> data pack

a. The embedded device requests to obtain the access status of the RF module.

Field	Length (L)	sensor type (T)	Command (C)
Value	6	RF_TYPE	CMD_ASK_STATUS

b. Sensor feedback RF module access status

Field 字段	Length (L)	sensor type (T)	Command (C)	DATA1	DATA2	DATA3
Value	10	RF_TYPE	CMD_ANSWER_STATUS			

Description：

i. DATA1(1 byte): 0 Indicates that the RF module is connected; 1 indicates that the RF module is not connected.

ii. DATA2(2 bytes): Effective when DATA1 is 0, indicating the address of the RF module

iii. DATA3(1 byte): Indicates the channel currently used by the RF module

c. The address of the RF module is set by the embedded device.

Field	Length (L)	sensor type (T)	Command (C)	DATA1
Value	8	RF_TYPE	CMD_SET_ADDRESS	

Description 说明：

DATA1(2 bytes): Indicates the address of the RF module

d. The embedded device sends data to the RF module

Field	Length (L)	sensor type (T)	Command (C)	DATA1	DATA2
Value	9-1024	RF_TYPE	CMD_SEND_DATA		

Description 说明：

i. DATA1(2 bytes): Indicates the address of the target RF module

ii. DATA2(1-1016 bytes): data

e. The sensor forwards the data of the RF module to the embedded device

Field	Length (L)	sensor type (T)	Command (C)	DATA1	DATA2
Value	9-1024	RF_TYPE	CMD_RECV_DATA		

Description 说明：

i. DATA1(2 bytes): Indicates the address of the data source RF module

ii. DATA2(1-1016 bytes): data

f. Embedded device sets the channel of the RF module

Field	Length (L)	sensor type (T)	Command (C)	DATA1
Value	7	RF_TYPE	CMD_SET_CHANNEL	

Description：

DATA1(1 byte): Channel representing the RF module

4.3.13 Program switching

1> data pack

Field	Length (L)	sensor type (T)	Command (C)
	8	SWITCH_PROGRAM_TYPE	Program index value

4.4 Embedded device gets sensor data

Disclaimer: Embedded device (A), sensor (B), open the serial port parameter settings (baud rate 9600, data bit 8, stop bit 1, check digit none)

4.4.1 brightness

A Get B Current brightness, A package sensor type (LUMINANCE_TYPE), command type (CMD_ASK_LUMINANCE) packet is written to the serial port for the brightness value in request B, B package sensor type (LUMINANCE_TYPE), command type (CMD_ANSWER_LUMINANCE) packet Write to the serial port for feedback.

4.4.2 temperature

A Get B current temperature, A package sensor type (TEMPERATURE_TYPE), command type (CMD_ASK_TEMPERATURE) packet is written to the serial port for the temperature value in request B, B package sensor type (TEMPERATURE_TYPE), command type (CMD_ANSWER_TEMPERATURE) packet Write to the serial port for feedback.

4.4.3 humidity

A Get B Current humidity, A package sensor type (HUMIDITY_TYPE), command type (CMD_ASK_HUMIDITY) packet is written to the serial port for the humidity value in request B, B package sensor type (HUMIDITY_TYPE), command type (CMD_ ANSWER _HUMIDITY) Packet is written to the serial port for feedback

4.4.4 remote control

B When receiving the button of the remote control, the active package sensor type (TELECONTROLLER_TYPE), the command type (the key value definition corresponding to the 5.1 appendix) is written into the serial port, and A can check whether there is readable data in the serial port. Check if the user has a button.

4.4.5 Temperature 2

A Get B current temperature, A package sensor type (TEMPERATURE2_TYPE), command type (CMD_ASK_TEMPERATURE2) packet is written to the serial port for the temperature value in request B, B package sensor type (TEMPERATURE2_TYPE), command type (CMD_ANSWER_TEMPERATURE2) packet Write to the serial port for feedback.

4.4.6 Switch screen

a. Set current time

A will set the small board time data packet to the serial port. After receiving the data packet, B will set the time and write the CMD_SET_TIME feedback packet to the serial port for feedback.

b. Set the screen opening time

A will set the open time data packet to the serial port. After receiving the data packet, B will write the CMD_SET_ON feedback packet to the serial port for feedback, and open the open screen timer to process the operation at the specified time.

c. Set the screen closing time

A will write the off-screen time data packet to the serial port. After receiving the data packet, B will write the CMD_SET_OFF feedback packet to the serial port for feedback, and turn on the off-screen timer to handle the operation of closing the screen at the specified time.

5. appendix

5.1 remote control button function definition

Serial number	Remote control button	MCU decoding code value	Control card function
1	Red switch key	0xFE00 AF50	Screen switch
2	Mute button	0xFE00 4FB0	Mute switch
3	Number button	0xFE00 F708	Program selection, 0
4	Number button	0xFE00 7788	Program selection, 1
5	Number button	0xFE00 B748	Program selection, 2
6	Number button	0xFE00 37C8	Program selection, 3
7	Number button	0xFE00 D728	Program selection, 4
8	Number button	0xFE00 57A8	Program selection, 5
9	Number button	0xFE00 9768	Program selection, 6
10	Number button	0xFE00 17E8	Program selection, 7
11	Number button	0xFE00 E718	Program selection, 8
12	Number button	0xFE00 6798	Program selection, 9
13	EPG button	0xFE00 8778	All program loops
14	RECALL button	0xFE00 FD02	Single program loop
15	MENU button	0xFE00 A758	Unused
16	EXIT button	0xFE00 C738	Reset
17	CH Up button 上	0xFE00 FF00	Previous show
18	CH Down button	0xFE00 7F80	Next show
19	VOL Left button	0xFE00 3FC0	Decrease brightness

			by one
20	VOL Right button	0xFE00 BF40	Brightness plus one
21	OK button	0xFE00 07F8	Play/stop
22	STA button	0xFE00 DF20	Unused
23	FAV button	0xFE00 27D8	Unused
24	INFO button	0xFE00 2FD0	Unused
25	FIND button	0xFE00 CF30	Unused
26	SUB-T button	0xFE00 857A	Unused
27	TV/R button	0xFE00 47B8	Screen test
28	AUDIO button	0xFE00 8F70	Screen test white
29	TTX button	0xFE00 659A	Unused
30	CARD button	0xFE00 5FA0	Screen test red
31	MAILBOX button	0xFE00 6F90	Screen test green
32	MP button	0xFE00 3DC2	Screen test blue

5.2 CRC-32 Checksum formula.

```
#define ReverseUInt(value)  \
    ((value >> 24) | ((value >> 16) << 8) | ((value & 0xff00) << 8) | ((value
& 0xff) << 24))


unsigned int crc32(unsigned char *message, int len) {
    if ((message == 0) || (len < 1)) {
        return 0;
    }

    unsigned int byte, crc;
    crc = 0xFFFFFFFF;
    for (int i=0; i<len; i++) {
        byte = message[i];
        byte = ReverseUInt(byte);
        for (int j=0; j<8; j++) {
            if (((int)(crc ^ byte)) < 0) {
                crc = (crc << 1) ^ 0x04C11DB7;
            } else {
                crc = crc << 1;
            }
            byte = byte << 1;
        }
    }
```

```
return ReverseUInt(~crc);
 }
```