UVC Functionality Development Guide

Foreword

This document applies to the development of UVC functionality in modules, and contains all implementation methods and suggestions for supporting UVC extension unit 2.0 protocol communication . Its purpose is to provide relevant guidance and requirements for developers on both the device and host sides .

Revision history

Version	Change description	date	Change person

Table of contents

Terms and Definitions

the term	describe
USB	Universal Serial Bus.
UVC	USB Video Class.
Device	USB peripherals. This document specifically refers to the HIK temperature measurement module .
Host	The computer system on which the host controller is installed. This document specifically refers to the user host device .
VC	Video Control. An interface used for controlling video functions.
VS	Video Streaming. An interface used for video streaming.
XU	Extension Unit . Extended unit.
Endpoint	Endpoint . A data source or receiver on a USB device.
Entity	A Unit, Terminal, or Interface. Each entity can contain controls.
Interface	An Entity represents an entity that is a collection of zero or more endpoints that provide functionality to a Host .
Terminal	An Entity represents the source (input terminal) or destination (output terminal) of data flowing into or out of a video function .
Unit	An Entity. This refers to the entity that performs the data transformation that flows through a video function .
UD	Unit Descriptor .
Descriptor	A data structure used to describe the functions or characteristics of a USB device.

1. UVC Extension Unit 2.0 Protocol Design

Compared to the custom extension unit 1.0 protocol, the extension unit 2.0 protocol establishes a completely new communication and interaction mechanism between the device and the host . The device side uniformly receives and sends data in little-endian byte order .

1.1 Protocol Basics

Use UVC Class - Specific Requests .

One-way request : The host initiates the request, and the device is responsible for the response.

1.1.1 Message Format

bmRequestType	bRequest	wValue	wIndex	wLength	Data
00100001	SET_CUR	CS	Extension Unit ID and Interface	Length of parameter block	Parameter block
10100001	GET_CUR GET_MIN GET_MAX GET_RES GET_DEF GET_LEN GET_INFO				

Field names are preceded by : b - single byte, w - double byte.

1.1.1.1 bmRequestType field

The direction of data transmission after this request is executed .

0 = Host to Device (SET Request)

1 = Device to Host (GET request)

D6-D5: Command Types

00 = Standard Request Command

01 = Specific class request command (used by this extension protocol)

10 = Custom request command

11 = Reserved

D4-D0: Type of command receiver

00000 = Equipment

00001 = Interface (VC or VS interface)

00010 = Endpoint (Endpoint of VS interface)

00011 = Other

Other undefined values are reserved.

1.1.1.2 bRequest field

Request command types, including SET and GET.

process of this extended unit protocol mainly uses the following three request commands :

SET_CUR(0x01) - set up

GET_CUR(0x81) - Retrieve

GET_LEN(0x85) - Get the data length

Class-Specific Request Code	Value
RC_UNDEFINED	0x00
SET_CUR	0x01
SET_CUR_ALL	0x11
GET_CUR	0x81
GET_MIN	0x82
GET_MAX	0x83
GET_RES	0x84
GET_LEN	0x85
GET_INFO	0x86
GET_DEF	0x87
GET_CUR_ALL	0x91
GET_MIN_ALL	0x92
GET_MAX_ALL	0x93
GET_RES_ALL	0x94
GET_DEF_ALL	0x97

1.1.1.3 wValue field

The value is limited by the Entity ID in the wIndex field. The two bytes of the wValue field have slightly different meanings depending on the entity being addressed . For extended units, the high byte indicates the control ID (CS_ID later in this document, the quantity of which is specified by the extended unit descriptor) , and the low byte is limited to 0.

XU_CS_ID	definition	illustrate	Notes
0x00	XU_CS_ID_UNDEFINED	reserve	
0x01	XU_CS_ID_SYSTEM	System Management	

0x02	XU_CS_ID_IMAGE	Video image management	
0x03	XU_CS_ID_THERMAL	Thermal imaging temperature measurement management	
0 x04	XU_CS_ID_PROTOCOL_VER	Extended Protocol Version Detection	2.0 protocol adds
0 x05	XU_CS_ID_COMMAND_SWITCH	Function command switching	2.0 protocol adds
0x06	XU_CS_ID_ERROR_CODE	Error code retrieval	2.0 protocol adds

1.1.1.4 wIndex field

Low byte: Indicates the ID of the addressed Interface or Endpoint.

High byte : Indicates the ID of the addressed Entity (Unit, Terminal, or Interface) or 0.

For this Extended Unit Protocol , the Unit ID is set to 0x0A .

1.1.1.5 wLenth field

This field indicates the size of the data transmitted during the control data transmission phase ; its specific meaning may vary depending on the request .

SET_CUR request: Indicates the length of the Data field . (No response message)

GET_*** request: without parameters , the wLenth field specifies the maximum length of the Data field in the response message .

Note : The wLenth field of the extended unit GET_LEN request is limited to 2.

1.1.2 Error code

Error codes are set by the device and can only be obtained by the host .

Error code	Error code description
0x00	normal
0x01	The device has not yet completed its previous request; the host must wait for it to finish.
0x02	If the device is in a state where certain requests are not allowed, the host needs to initiate other specific requests to change this state. 1) Making a request with an inconsistent CS_ID before and after without

	performing a feature switch. 2) ROI Highest Temperature Query: GET_CUR without sending parameters via SET_CUR 3) The current temperature measurement mode does not meet the requirements for executing a specific request. 4) Upgrading or formatting in progress; upgrade not allowed at this time.
0x03	The actual power mode on the device side is insufficient to fulfill the request.
0 x04	The parameters set in the SET_CUR request are outside the allowed range.
0 x05	Unsupported Unit ID
0 x06	Unsupported CS ID
0 x07	Unsupported request command types
0 x08	The SET_CUR request sets parameters within the specified range, but the set value is invalid.
0 x09	Unsupported sub-functions
0 x0A	Device -side function execution error
0 x0B	Device -side internal protocol process anomaly
0 x0C	Big data transmission process anomaly 1) Request command type not supported 2) Data type mismatch (Host→Device) 3) Total data length anomaly (Host→Device) 4) Packet sequence number error (Host→Device) 5) The actual received length is inconsistent with the total data length (Host→ Device)
0 x0D	wL ength field in the SET_CUR request message is abnormal.
0 x0E - 0xFE	Reserved for future use
0xFF	Unknown

1.2 communication process

An initialization process has been added to distinguish which version of the extended protocol is being used for communication . The protocol interaction process has been redefined , with the length obtained using GET_LEN before each configuration request . Function switching is implemented through a specific CS_ID ; the device will remain in the previous functional state unless a function switch is received (it is in a non-functional state by default after power -on initialization). When the length of the Data field exceeds the maximum length that a single frame can hold,

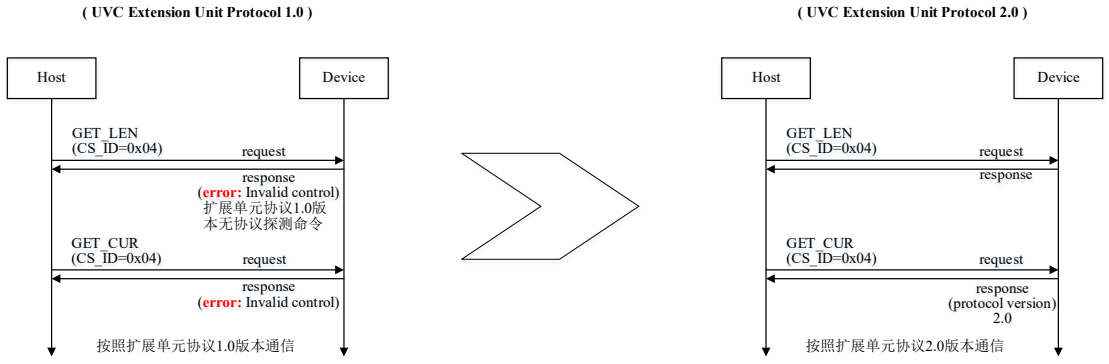
it will be transmitted according to the large data transmission process .

Obtaining the length via GET_LEN : Before sending each SET_CUR or GET_*** request , the host must first obtain the value of the wLenth field in the request message via GET_LEN . For GET_*** requests, the length value obtained via GET_LEN is the maximum length of the Data field in the GET_*** request response message — currently, except for the upgrade process where GET_LEN returns the actual length of the Data field in the response message before obtaining the upgrade status via GET_CUR , all other requests return the actual length of the response message .

Note: Except for SET requests, the Data field of other request messages does not include parameters!

1.2.1 Initialization process (protocol version query)

formal communication, the host must actively obtain the extended unit protocol version from the device using a GET_CUR request . Only when the device responds normally and returns a protocol version of 2.0 can the host communicate with the device according to the protocol defined in this document. If a restart occurs during communication , whether on the host or device side, the host must resend the protocol version retrieval request after the restart .



a request message:

	Length	Phase	Data		Description
Step1:		CTL	a1 85 00 04 00 0a 02 00	...	GET LEN
	2	IN	04 00	...	..
Step2:		CTL	a1 81 00 04 00 0a 04 00	...	GET CUR
	4	IN	32 2e 30 00	...	2.0.

Message reading and interpretation: CTL - UVC request message 8-byte header .

IN - DATA field of the response message ; OUT - DATA field of the request message .

Parsing the Data field of the response message:

Offset	Field	Size	Value type	Description
0	Extension unit protocol version	4	String	2.0

1.2.2 Function switching process

Since the function ID is divided into a main function ID (corresponding to CS_ID) and a sub-function ID, and the high byte of the wValue field in the request header sent by the host is the CS_ID, the sub-function ID cannot be transmitted to the device along with the request message . To inform the device of the current complete function request type , a function switching mechanism is defined. The host can use the following two switching schemes:

Option 1: Before each request is sent, the host must first switch the function.

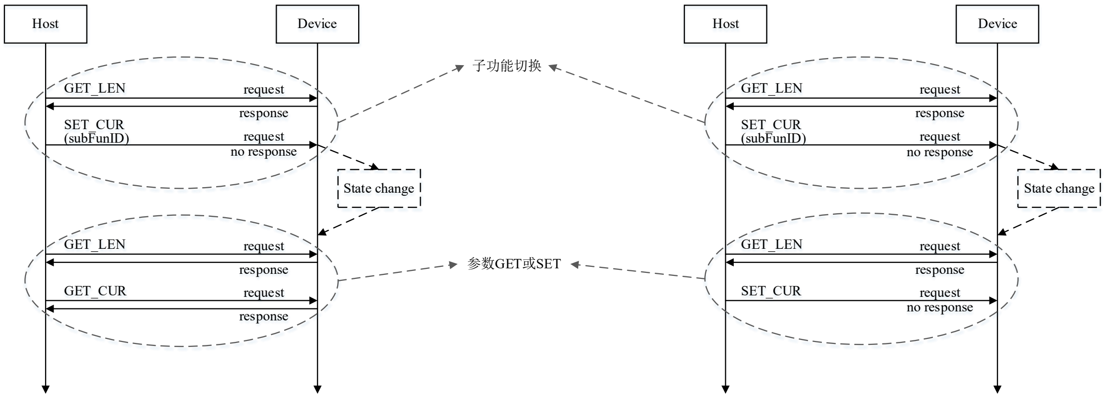
Option 2 : The host records the functional communication status . Before sending each request, it checks whether the current request is for the same function as the previous request . If the functions are different, a function switch request should be sent first. (Option 2 is recommended.)

Note: Regardless of which of the above methods is used on the host side , the following requirements must be met :

(1) Function switching is not allowed during big data transmission .

(2) Since the ROI highest temperature query is divided into two steps: first, the parameters are set through the SET_CUR request , and then the required information is obtained through the GET_CUR request . Therefore, the function switching is only allowed to occur before the SET_CUR request .

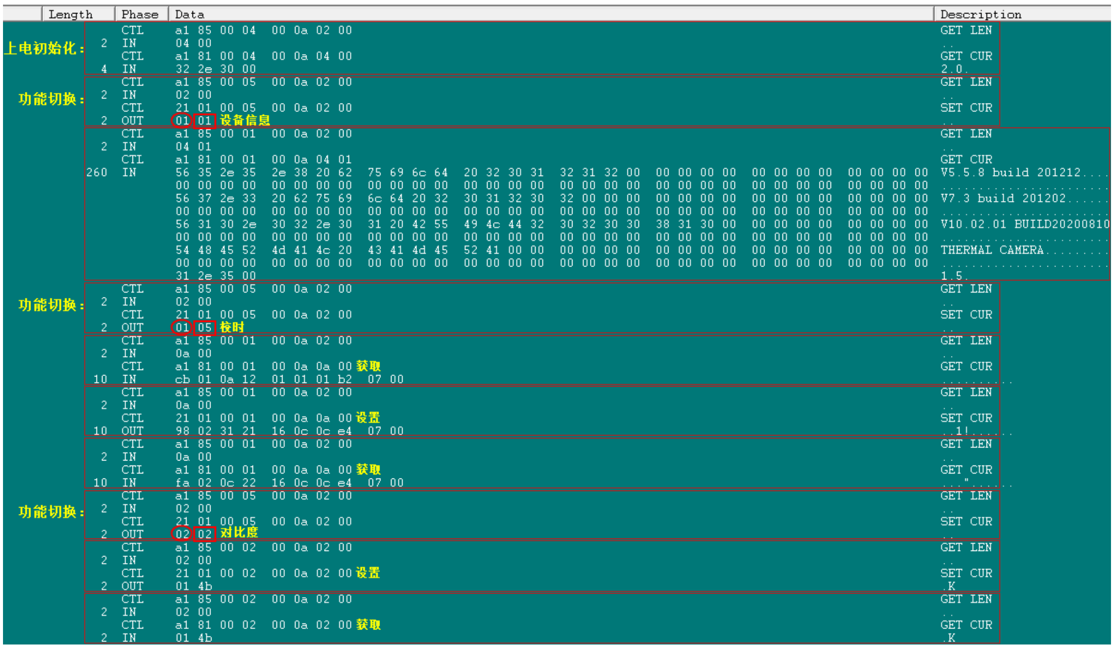
(3) After power-on initialization, the Device is in a no-function state by default . Before the Host sends a function request for the first time, it must first send a function switching request.



the function switch request message:

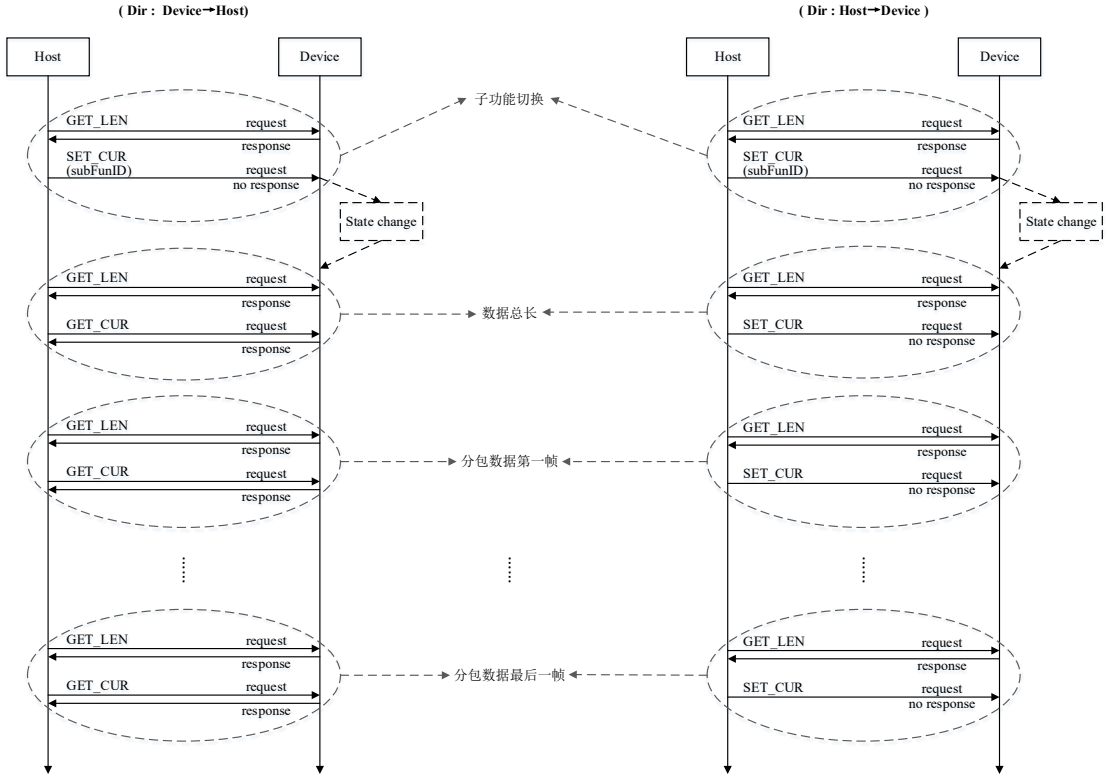
O ffset	Fi el d	Size	Value type	Description
0	CS_ID	1	Number	0x05
1	Sub-function ID	1	Number	See Appendix A for details.

function switching:



1.2.3 Big data transmission process

the length of the Data field exceeds the maximum length that a single frame can hold , the data will be transmitted in packets . When the receiver is a Device, a SET_CUR request is used ; when the receiver is a Host, a GET_CUR request is used .



The transmission process consists of the following two steps (see Section 1.3.4 for a typical example of expert temperature measurement rule configuration):

Step 1 : The data receiver obtains the total data length (excluding the 5-byte frame header of each packet data frame).

Message direction	Sender→Receiver			
Data				
Offset	Field	Size	Value type	Description
0	Data type	1	Number	Data type: 0x0 1 - Total data length
1	Total length	4	Number	data length

Step 2 : The data receiver obtains the sub-packet data.

Message direction	Sender→Receiver			
Data				
Offset	Field	Size	Value type	Description

0	Data type	1	Number	Data type: 0x02 - Packet Data
1	Packet sequence number	4	Number	Package number, incrementing from 1
5	Packet data	Length	Byte	Subcontracted data

current agreement adopts a pre-agreed approach, and data transmission according to big data processes has been pre - defined , including :

(1) Upgrade

(2) Capture heat map (with full-screen temperature measurement data)

(3) Obtain the highest temperature information of ROI

(4) Export diagnostic files

(5) Export/import calibration files

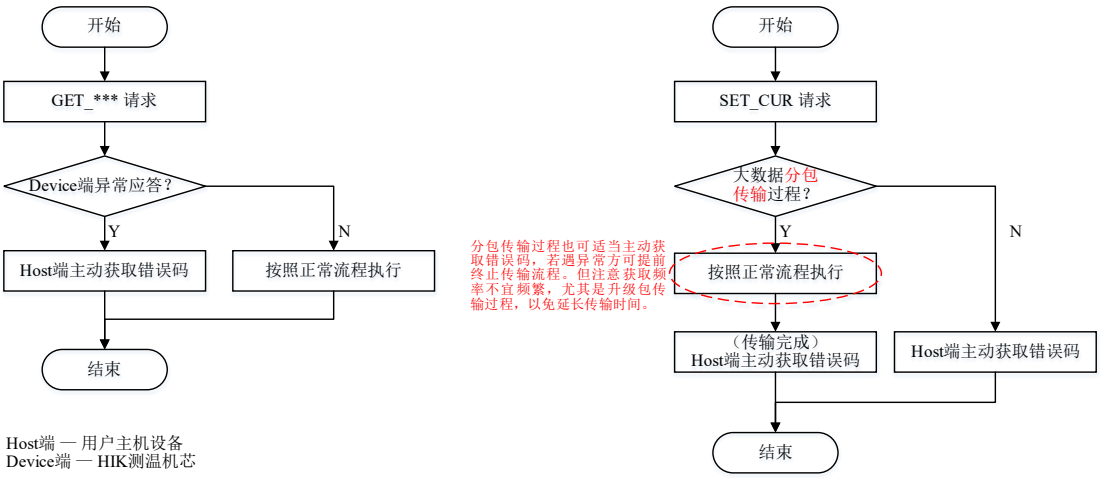
(6) Obtain/set expert temperature measurement rules

1.2.4 Error code retrieval process

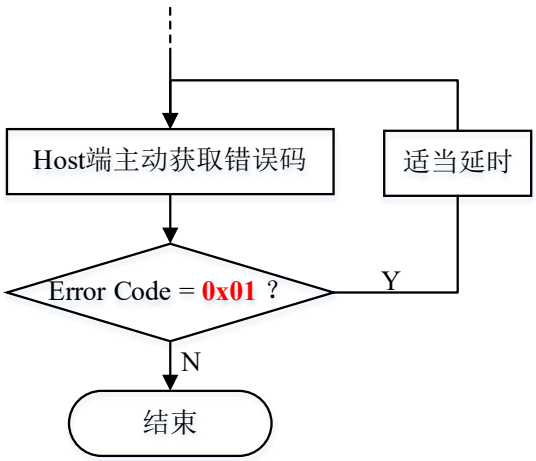
The UVC standard specification includes an error code retrieval mechanism . For this extended unit protocol, in addition to supporting the standard error code retrieval mechanism , a custom error code retrieval mechanism is also defined (Windows, Linux x86 , and Android users using our USBSDK can currently only retrieve error codes using the custom mechanism).

Currently, the Device side has expanded to include 5 additional error codes on top of the existing 9 , as detailed in Section 1.1.2, Error Code Definition . The Device 's error code update mechanism is as follows: For every request from the Host (excluding error code retrieval requests) , the Device updates the error code after execution . For all GET requests , the Host can directly detect the exception when the Device sets an error code . However, for SET_CUR requests , the Host cannot directly detect whether an exception has occurred on the Device , therefore it needs to actively request and retrieve the error code. The Host ultimately determines whether an exception has occurred on the Device based on the retrieved error code value (a non-zero value indicates an exception). The decision on when to retrieve the error code rests with the Host. This extended unit protocol does not fixate on any process ; the Host can refer to

the process shown in the diagram below to design its error code retrieval logic .



For SET requests, the Device receives the data completely before parsing and verifying it. Furthermore, functions such as restoring default settings , writing calibration files, setting expert temperature measurement rules , and setting expert temperature measurement correction parameters require a certain amount of time to execute . Therefore, if the Host immediately retrieves the error code after data transmission , the Device may not have had time to update the error code, potentially leading to request blocking. Thus , for these scenarios, the Device first sets the error code to 0x01 after receiving the data, and then updates the error code based on the execution result after the function is completed. Therefore, if the Host receives an error code of 0x01, it needs to continuously poll and wait : for restoring default settings and setting expert temperature measurement correction parameters, polling can be done at 1-second intervals ; for other function requests, it is recommended to poll at intervals of at least 10ms .



Request message format for obtaining error codes according to UVC standard :

Offset	Field	Size	Value	Description
0	**bmRequestType**	1	0xA1	D7: 1 = Device To host D6..5: 01 = Class request D4..0: 00001 = Recipient is interface
1	**bRequest**	1	0x81 0x86	GET_CUR GET_INFO
2	**wValue**	2	0x0200	Get the error code control (02) of the interface.
4	**wIndex**	2	0x0000	Only send request to the VideoControl interface (interface ID 00)
6	**wLength**	2	0x0001	Parameter block length

Request message format when retrieving error codes using a custom mechanism :

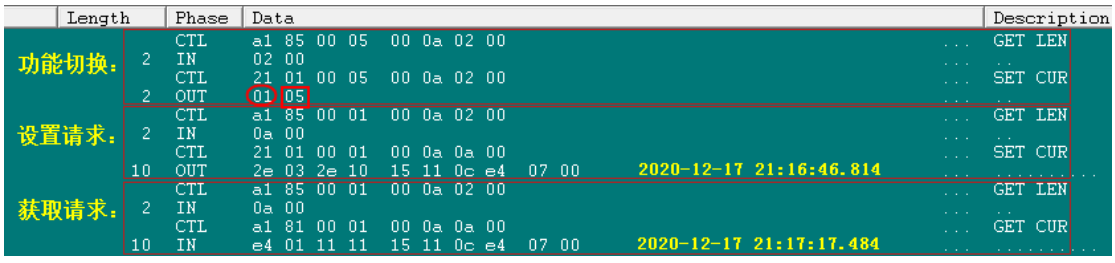
Offset	Field	Size	Value	Description
0	bmRequestType	1	0 xA 1	Refer to UVC standard protocol
1	bRequest	1	0 x81	GET_CUR
2	wValue	2	0x 0600	Extended unit custom CS_ID
4	w Index	2	0x0A0 0	Thermal Imaging Extended Unit ID
6	wLength	2	0x0001	Refer to UVC standard protocol

1.3 Typical Communication Scenarios

The examples in this section demonstrate that the function switching follows Scheme 2 in the function switching process described in Section 1.2.2 . For detailed parameter parsing of the Data field in each request and response message, please refer to Appendix A.

1.3.1 Time synchronization configuration

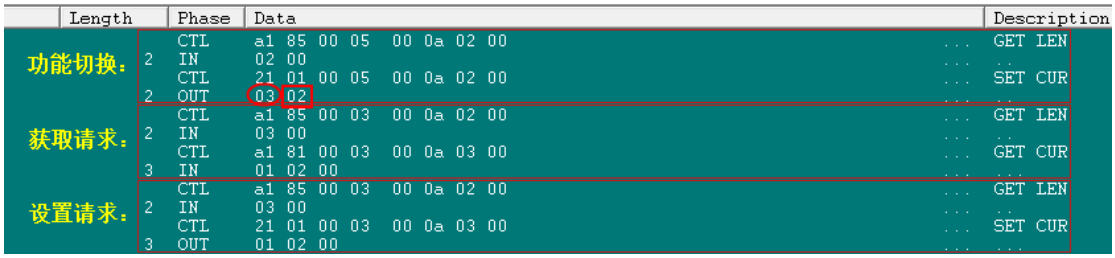
a get/set request message :



1.3.2 Temperature measurement mode configuration

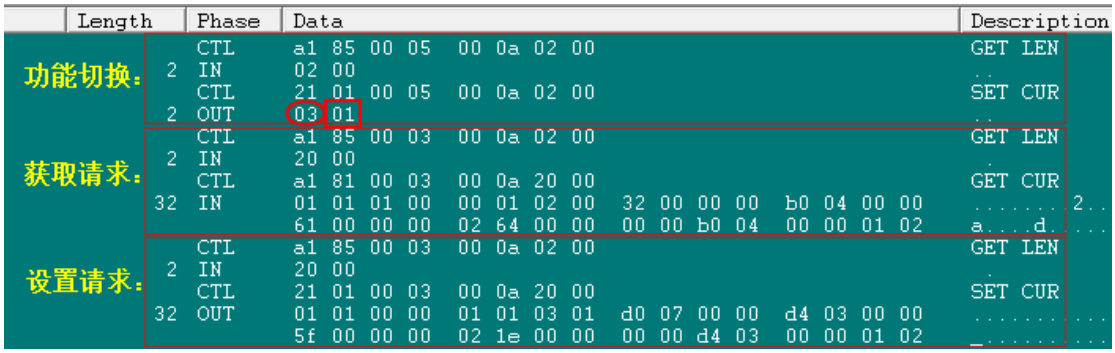
setting temperature measurement parameters such as temperature measurement zone rules and basic temperature measurement parameters, you must ensure that the current temperature measurement mode on the device meets expectations .

get/set request message (industrial model):



1.3.3 Temperature measurement basic parameter configuration

a get/set request message :



1.3.4 Expert temperature measurement rule configuration

The configuration settings and retrieval requests for expert temperature measurement rules follow the big data transmission process outlined in Section 1.2.3 . This means that the total data length is transmitted first , followed by the segmented data . Retrieval uses a GET_CUR request , and setting uses a SET_CUR request .

Note: The temperature measurement mode needs to be set to expert temperature measurement mode in advance .

obtaining a request message :

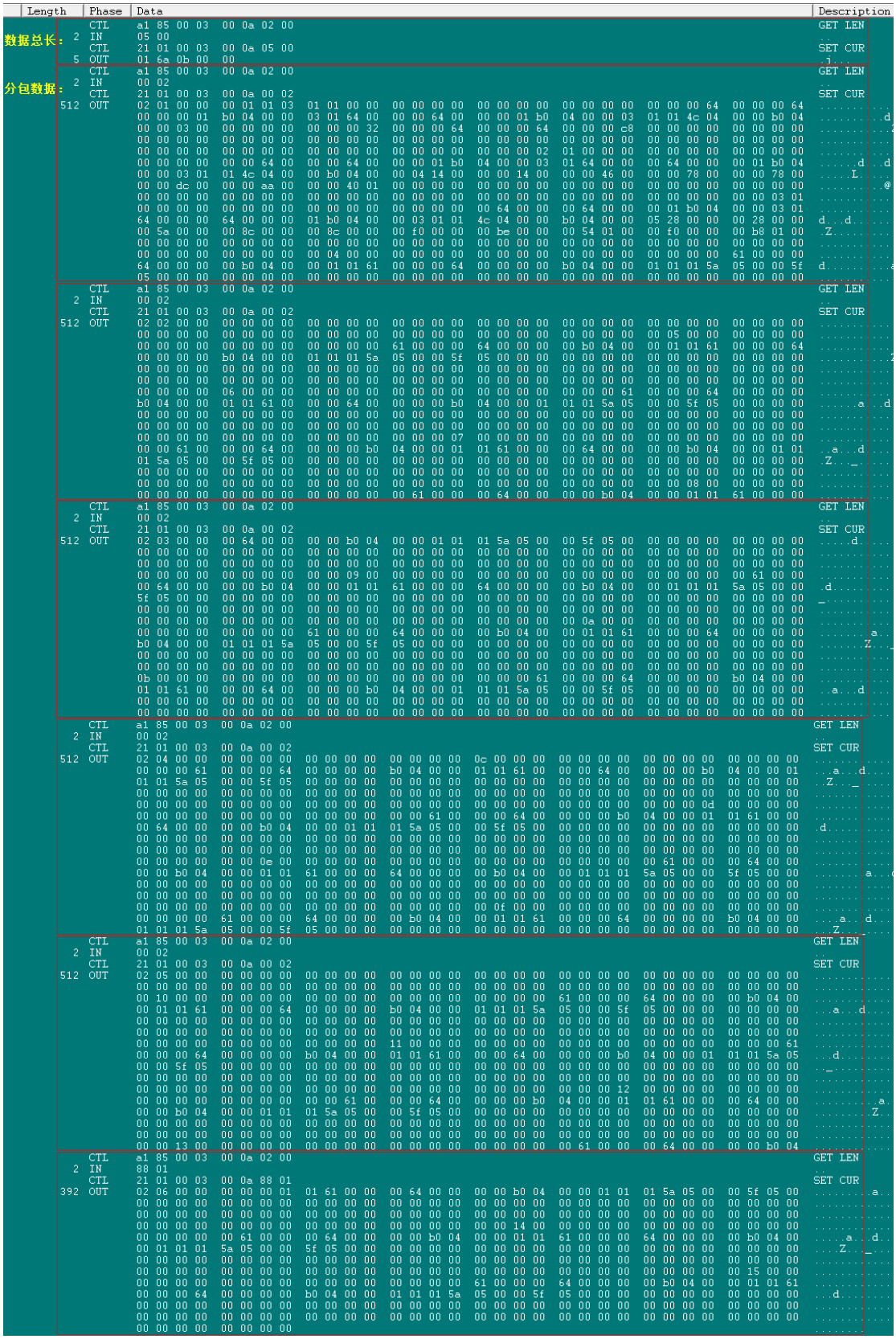
功能切换：

数据总长：

分包数据：

Length	Phase	Data	Description
	CTL	a1 85 00 05 00 0a 02 00	GET LEN
2	IN	02 00	..
	CTL	21 01 00 05 00 0a 02 00	SET CUR
2	OUT	03 0f	..
	CTL	a1 85 00 03 00 0a 02 00	GET LEN
2	IN	05 00	..
	CTL	a1 81 00 03 00 0a 05 00	GET CUR
5	IN	01 6a 0b 00 00	.j...
	CTL	a1 85 00 03 00 0a 02 00	GET LEN
2	IN	00 02	..
	CTL	a1 81 00 03 00 0a 00 02	GET CUR
512	IN	02 01 00 00 00 01 01 03 01 01 01 01 64 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00	.....
		00 00 00 00 00 00 00 00 00 00 64 00 00 00 64 00 00 00 01 b0 04 00 00 03 01 01 4c 04 00 00 b0 04	.....d
		00 00 03 00 00 00 00 00 00 00 00 32 00 00 00 64 00 00 00 64 00 00 00 c8 00 00 00 00 00 00 00 00	2
		00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00	.....
		00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 02 01 01 01 64 00 00 00 00 00 00 00 00	.....
		00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 64 00 00 00 64 00 00 00 01 b0 04	.....
		00 00 03 01 01 4c 04 00 00 b0 04 00 00 04 14 00 00 00 14 00 00 00 46 00 00 00 78 00 00 00 78 00	.....L.
		00 00 dc 00 00 00 aa 00 00 00 40 01 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00	.....@.
		00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 03 01	.....
		01 01 64 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00	..d...
		64 00 00 00 64 00 00 00 01 b0 04 00 00 03 01 01 4c 04 00 00 b0 04 00 00 05 28 00 00 00 28 00 00	d...d.
		00 5a 00 00 00 8c 00 00 00 8c 00 00 00 f0 00 00 00 be 00 00 00 54 01 00 00 f0 00 00 00 b8 01 00	.Z.....
		00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00	.....
		00 00 00 00 00 00 00 00 00 00 04 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00	.....
		00 00 00 00 00 00 00 00 00 00 00 00 61 00 00 00 64 00 00 00 b0 04 00 00 01 01 01 5a 05 00 00 5f	.....a
		05 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00	.....
	CTL	a1 85 00 03 00 0a 02 00	GET LEN
2	IN	00 02	..
	CTL	a1 81 00 03 00 0a 00 02	GET CUR
512	IN	02 02 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00	.....
		00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 05 00 00 00 00 00 00 00	.....
		00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 61 00 00 64	.....d
		00 00 00 00 b0 04 00 00 01 01 01 5a 05 00 00 5f 05 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00	.....Z.
		00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00	.....
		00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00	.....
		00 00 00 00 00 00 00 00 06 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00	.....
		00 00 00 00 00 00 61 00 00 00 64 00 00 00 00 b0 04 00 00 01 01 01 5a 05 00 00 5f 05 00 00 00 00	.....a...d
		00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00	.....
		00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00	.....
		00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 07 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00	.....
		00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 61 00 00 00 64 00 00 00 b0 04 00 00 01 01	.....a...d.....
		01 5a 05 00 00 5f 05 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00	.Z..._
		00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00	.....
		00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00	.....
		00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 61 00 00 00	.....a...
	CTL	a1 85 00 03 00 0a 02 00	GET LEN
2	IN	00 02	..
	CTL	a1 81 00 03 00 0a 00 02	GET CUR
512	IN	02 03 00 00 00 64 00 00 00 00 b0 04 00 00 01 01 01 5a 05 00 00 5f 05 00 00 00 00 00 00 00 00	.....d.....
		00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00	.....
		00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00	.....
		00 00 00 00 00 00 00 00 00 00 00 09 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00	.....
		00 00 00 00 00 00 00 00 00 00 00 00 61 00 00 00 64 00 00 00 00 b0 04 00 00 01 01 01 5a 05 00 00	.....a...d.....Z...
		5f 05 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00	_.....
		00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00	.....
		00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00	.....
		00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 61 00 00 00	.....a...
		b0 04 00 00 01 01 01 5a 05 00 00 5f 05 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00	.....Z..._
		00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00	.....
		00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00	.....
		0b 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00	.....
		00 00 61 00 00 00 64 00 00 00 00 b0 04 00 00 01 01 01 5a 05 00 00 5f 05 00 00 00 00 00 00 00 00	..a...d.....Z..._
		00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00	.....
		00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00	.....
	CTL	a1 85 00 03 00 0a 02 00	GET LEN
2	IN	00 02	..
	CTL	a1 81 00 03 00 0a 00 02	GET CUR
512	IN	02 04 00 00 00 00 00 00 00 00 00 00 00 00 00 00 0c 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00	.....
		00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 61 00 00 00 64 00 00 00 00 b0 04 00 00 01	.....a...d.....
		01 01 5a 05 00 00 5f 05 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00	..Z..._
		00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00	.....
		00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 0d	.....
		00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 61 00	.....a.
		00 64 00 00 00 b0 04 00 00 01 01 01 5a 05 00 00 5f 05 00 00 00 00 00 00 00 00 00 00 00 00 00 00	.d.....Z..._
		00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00	.....
		00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00	.....
		00 00 00 00 00 00 00 0e 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00	.....
		00 00 00 00 00 00 00 00 61 00 00 00 64 00 00 00 00 b0 04 00 00 01 01 01 5a 05 00 00 5f 05 00 00	.....a...d.....Z..._
		00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00	.....
		00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00	.....
		00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 0f	.....
		00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 61 00 00 00 64 00 00 00 b0 04 00 00 01	.....a...d.....
		01 01 5a 05 00 00 5f 05 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00	..Z..._
	CTL	a1 85 00 03 00 0a 02 00	GET LEN
2	IN	00 02	..
	CTL	a1 81 00 03 00 0a 00 02	GET CUR
512	IN	02 05 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00	.....
		00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00	.....
		00 10 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00	.....
		00 00 00 61 00 00 00 64 00 00 00 00 b0 04 00 00 01 01 01 5a 05 00 00 5f 05 00 00 00 00 00 00 00	...a...d.....Z..._
		00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00	.....
		00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00	.....
		00 00 00 00 00 00 00 00 00 00 00 00 11 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00	.....
		00 00 00 00 00 00 00 00 00 00 00 00 00 00 61 00 00 00 64 00 00 00 b0 04 00 00 01 01 01 5a 05 00	.....a...d.....Z..
		00 00 5f 05 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00	.._.....
		00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00	.....
		00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00	.....
		00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 12	.....
		00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 61 00 00 64	.....a...d
		00 00 b0 04 00 00 01 01 01 5a 05 00 00 5f 05 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00	.....Z..._
		00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00	.....
		00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00	.....
		00 00 13 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00	.....
	CTL	a1 85 00 03 00 0a 02 00	GET LEN
2	IN	88 01	..
	CTL	a1 81 00 03 00 0a 88 01	GET CUR
392	IN	02 06 00 00 00 00 00 00 00 61 00 00 00 64 00 00 00 00 b0 04 00 00 01 01 01 5a 05 00 00 5f 05 00	.....a...d.....Z..._.
		00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00	.....
		00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00	.....
		00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 14 00 00 00 00 00 00 00 00 00 00 00 00	.....
		00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 61 00 00 00 64 00 00 00 00 b0 04 00	.....a...d.....
		00 01 01 01 5a 05 00 00 5f 05 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00	....Z..._.
		00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00	.....
		00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 15 00 00	.....
		00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 61	.....a
		00 00 00 64 00 00 00 00 b0 04 00 00 01 01 01 5a 05 00 00 5f 05 00 00 00 00 00 00 00 00 00 00 00	...d.....Z..._
		00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00	.....
		00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00	.....
		00 00 00 00 00 00 00 00	.....

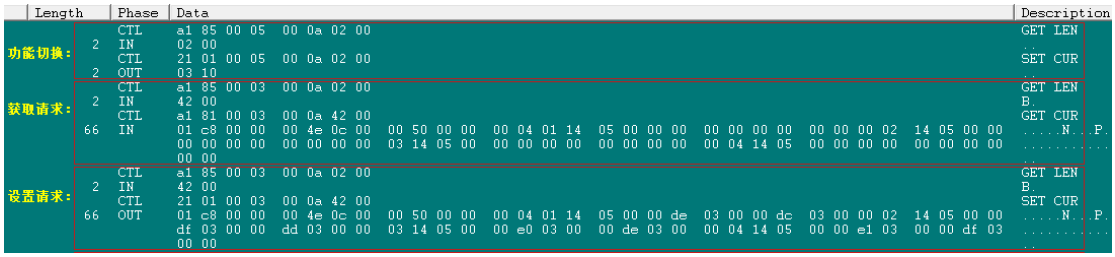
a configuration request message :



	Length	Phase	Data	Description
		CTL	a1 85 00 03 00 0a 02 00	GET LEN
数据总长：	2	IN	05 00	
		CTL	21 01 00 03 00 0a 05 00	SET CUR
	5	OUT	01 6a 0b 00 00	
		CTL	a1 85 00 03 00 0a 02 00	GET LEN
分包数据：	2	IN	00 02	
		CTL	21 01 00 03 00 0a 00 02	SET CUR
	512	OUT	02 01 00 00 00 01 01 03 01 01 00 00 00 00 00 00 00 00 00 00 00 00 00 00 64 00 00 00 64	
			00 00 00 01 b0 04 00 00 03 01 64 00 00 00 64 00 00 00 01 b0 04 00 00 03 01 01 4c 04 00 00 b0 04	
			00 00 03 00 00 00 00 00 00 00 00 32 00 00 00 64 00 00 00 64 00 00 00 c8 00 00 00 00 00 00 00 00	
			00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00	
			00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 02 01 00 00 00 00 00 00 00 00 00 00 00	
			00 00 00 00 00 00 64 00 00 00 64 00 00 00 01 b0 04 00 00 03 01 64 00 00 00 64 00 00 01 b0 04	
			00 00 03 01 01 4c 04 00 00 b0 04 00 00 04 14 00 00 00 14 00 00 00 46 00 00 00 78 00 00 00 78 00	
			00 00 dc 00 00 00 aa 00 00 00 40 01 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00	
			00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 03 01	
			00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 64 00 00 00 64 00 00 00 01 b0 04 00 00 03 01	
			64 00 00 00 64 00 00 00 01 b0 04 00 00 03 01 01 4c 04 00 00 b0 04 00 00 05 28 00 00 00 28 00 00	
			00 5a 00 00 00 8c 00 00 00 8c 00 00 00 f0 00 00 00 be 00 00 00 54 01 00 00 f0 00 00 00 b8 01 00	
			00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00	
			00 00 00 00 00 00 00 00 00 04 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 61 00 00 00 00	
			64 00 00 00 00 b0 04 00 00 01 01 61 00 00 00 64 00 00 00 00 b0 04 00 00 01 01 01 5a 05 00 00 5f	
			05 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00	
		CTL	a1 85 00 03 00 0a 02 00	GET LEN
	2	IN	00 02	
		CTL	21 01 00 03 00 0a 00 02	SET CUR
	512	OUT	02 02 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00	
			00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 05 00 00 00 00 00 00	
			00 00 00 00 00 00 00 00 00 00 00 00 61 00 00 00 64 00 00 00 00 b0 04 00 00 01 01 61 00 00 00 64	
			00 00 00 00 b0 04 00 00 01 01 01 5a 05 00 00 5f 05 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00	
			00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00	
			00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00	
			00 00 00 00 06 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 61 00 00 00 64 00 00 00	
			b0 04 00 00 01 01 61 00 00 00 64 00 00 00 b0 04 00 00 01 01 01 5a 05 00 00 5f 05 00 00 00 00 00	
			00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00	
			00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00	
			00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 07 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00	
			00 00 61 00 00 00 64 00 00 00 b0 04 00 00 01 01 61 00 00 00 64 00 00 00 b0 04 00 00 01 01	
			01 5a 05 00 00 5f 05 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00	
			00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00	
			00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 08 00 00 00 00	
			00 00 00 00 00 00 00 00 00 00 00 00 00 61 00 00 00 64 00 00 00 b0 04 00 00 01 01 61 00 00 00	
		CTL	a1 85 00 03 00 0a 02 00	GET LEN
	2	IN	00 02	
		CTL	21 01 00 03 00 0a 00 02	SET CUR
	512	OUT	02 03 00 00 00 64 00 00 00 00 b0 04 00 00 01 01 01 5a 05 00 00 5f 05 00 00 00 00 00 00 00 00	
			00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00	
			00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00	
			00 00 00 00 00 00 00 00 00 09 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 61 00 00 00 00	
			00 64 00 00 00 00 b0 04 00 00 01 01 61 00 00 00 64 00 00 00 00 b0 04 00 00 01 01 01 5a 05 00 00	
			5f 05 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00	
			00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00	
			00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 0a 00 00 00 00 00 00 00 00	
			00 00 00 00 00 00 61 00 00 00 64 00 00 00 00 b0 04 00 00 01 01 61 00 00 00 64 00 00 00 00 00 00	
			b0 04 00 00 01 01 01 5a 05 00 00 5f 05 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00	
			00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00	
			00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00	
			0b 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00	
			01 01 61 00 00 00 64 00 00 00 b0 04 00 00 01 01 01 5a 05 00 00 5f 05 00 00 00 00 00 00 00 00	
			00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00	
			00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00	
		CTL	a1 85 00 03 00 0a 02 00	GET LEN
	2	IN	00 02	
		CTL	21 01 00 03 00 0a 00 02	SET CUR
	512	OUT	02 04 00 00 00 00 00 00 00 00 00 00 00 00 00 00 0c 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00	
			00 00 00 61 00 00 00 64 00 00 00 00 b0 04 00 00 01 01 61 00 00 00 64 00 00 00 00 b0 04 00 00 01	
			01 01 5a 05 00 00 5f 05 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00	
			00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00	
			00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 0d	
			00 00 00 00 00 00 00 00 00 00 00 00 00 61 00 00 00 64 00 00 00 00 b0 04 00 00 01 01 61 00 00 00	
			00 64 00 00 00 00 b0 04 00 00 01 01 01 5a 05 00 00 5f 05 00 00 00 00 00 00 00 00 00 00 00 00 00	
			00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00	
			00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00	
			00 00 00 00 00 0e 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 61 00 00 00 64	
			00 00 b0 04 00 00 01 01 61 00 00 00 64 00 00 00 00 b0 04 00 00 01 01 01 5a 05 00 00 5f 05 00 00	
			00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00	
			00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00	
			00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00	
			00 00 00 00 61 00 00 00 64 00 00 00 00 b0 04 00 00 01 01 61 00 00 00 64 00 00 00 00 b0 04 00 00	
			01 01 01 5a 05 00 00 5f 05 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00	
		CTL	a1 85 00 03 00 0a 02 00	GET LEN
	2	IN	00 02	
		CTL	21 01 00 03 00 0a 00 02	SET CUR
	512	OUT	02 05 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00	
			00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00	
			00 10 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 61 00 00 00 64 00 00 00	
			00 01 01 61 00 00 00 64 00 00 00 00 b0 04 00 00 01 01 01 5a 05 00 00 5f 05 00 00 00 00 00 00 00	
			00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00	
			00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00	
			00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 11 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 61	
			00 00 00 64 00 00 00 00 b0 04 00 00 01 01 61 00 00 00 64 00 00 00 00 b0 04 00 00 01 01 01 5a 05	
			00 00 5f 05 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00	
			00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00	
			00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 12 00 00 00 00	
			00 00 00 00 00 00 00 00 00 61 00 00 00 64 00 00 00 00 b0 04 00 00 01 01 61 00 00 00 64 00 00 00	
			00 00 b0 04 00 00 01 01 01 5a 05 00 00 5f 05 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00	
			00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00	
			00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00	
			00 00 13 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 61 00 00 00 64 00 00 00 00 b0 04	
		CTL	a1 85 00 03 00 0a 02 00	GET LEN
	2	IN	88 01	
		CTL	21 01 00 03 00 0a 88 01	SET CUR
	392	OUT	02 06 00 00 00 00 00 01 01 61 00 00 00 64 00 00 00 00 b0 04 00 00 01 01 01 5a 05 00 00 5f 05 00	
			00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00	
			00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00	
			00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 14 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00	
			00 00 00 00 00 61 00 00 00 64 00 00 00 00 b0 04 00 00 01 01 61 00 00 00 64 00 00 00 00 b0 04 00	
			00 01 01 01 5a 05 00 00 5f 05 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00	
			00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00	
			00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 15 00 00 00 00	
			00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 61 00 00 00 64 00 00 00 00 b0 04 00 00 01 01 61	
			00 00 00 64 00 00 00 00 b0 04 00 00 01 01 01 5a 05 00 00 5f 05 00 00 00 00 00 00 00 00 00 00 00	
			00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00	
			00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00	
			00 00 00 00	

1.3.5 Expert temperature measurement calibration parameter configuration

Example of a get/set request message:



	Length	Phase	Data	Description
功能切换：		CTL	a1 85 00 05 00 0a 02 00	GET LEN
	2	IN	02 00	..
		CTL	21 01 00 05 00 0a 02 00	SET CUR
	2	OUT	03 10	..
获取请求：		CTL	a1 85 00 03 00 0a 02 00	GET LEN
	2	IN	42 00	B.
		CTL	a1 81 00 03 00 0a 42 00	GET CUR
	66	IN	01 c8 00 00 00 4e 0c 00 00 50 00 00 00 04 01 14 05 00 00 00 00 00 00 00 00 00 00 02 14 05 00 00 00 00 00 00 03 14 05 00 00 00 00 00 00 00 00 00 00 04 14 05 00 00 00 00 00 00 00 00 00 00	.....N...P.
设置请求：		CTL	a1 85 00 03 00 0a 02 00	GET LEN
	2	IN	42 00	B.
		CTL	21 01 00 03 00 0a 42 00	SET CUR
	66	OUT	01 c8 00 00 00 4e 0c 00 00 50 00 00 00 04 01 14 05 00 00 de 03 00 00 dc 03 00 00 02 14 05 00 00 df 03 00 00 dd 03 00 00 03 14 05 00 00 e0 03 00 00 de 03 00 00 04 14 05 00 00 e1 03 00 00 df 03 00 00	.....N...P.

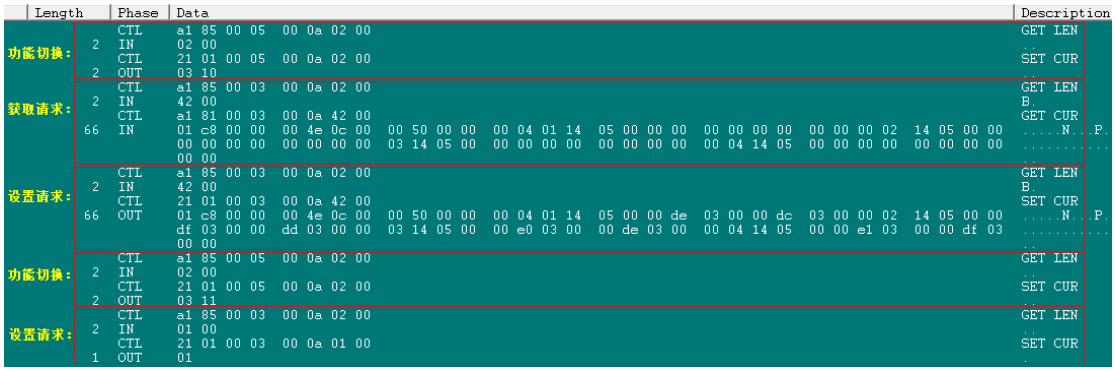
Notice:

(1) The normalization parameters are no longer the common [0,1000], but X [13,993] and Y [17,991] .

(2) Set four temperature points sequentially or simultaneously (recommended), which will take about 1 minute in total .

1.3.6 Expert temperature measurement and calibration begins

a get/set request message :



	Length	Phase	Data	Description
功能切换：		CTL	a1 85 00 05 00 0a 02 00	GET LEN
	2	IN	02 00	..
		CTL	21 01 00 05 00 0a 02 00	SET CUR
	2	OUT	03 10	..
获取请求：		CTL	a1 85 00 03 00 0a 02 00	GET LEN
	2	IN	42 00	B.
		CTL	a1 81 00 03 00 0a 42 00	GET CUR
	66	IN	01 c8 00 00 00 4e 0c 00 00 50 00 00 00 04 01 14 05 00 00 00 00 00 00 00 00 00 00 02 14 05 00 00 00 00 00 00 03 14 05 00 00 00 00 00 00 00 00 00 00 04 14 05 00 00 00 00 00 00 00 00 00 00	.....N...P.
设置请求：		CTL	a1 85 00 03 00 0a 02 00	GET LEN
	2	IN	42 00	B.
		CTL	21 01 00 03 00 0a 42 00	SET CUR
	66	OUT	01 c8 00 00 00 4e 0c 00 00 50 00 00 00 04 01 14 05 00 00 de 03 00 00 dc 03 00 00 02 14 05 00 00 df 03 00 00 dd 03 00 00 03 14 05 00 00 e0 03 00 00 de 03 00 00 04 14 05 00 00 e1 03 00 00 df 03 00 00	.....N...P.
功能切换：		CTL	a1 85 00 05 00 0a 02 00	GET LEN
	2	IN	02 00	..
		CTL	21 01 00 05 00 0a 02 00	SET CUR
	2	OUT	03 11	..
设置请求：		CTL	a1 85 00 03 00 0a 02 00	GET LEN
	2	IN	01 00	..
		CTL	21 01 00 03 00 0a 01 00	SET CUR
	1	OUT	01	.

Note: The calibration parameters must be reset before each calibration begins!

1.3.7 Real-time upload bitrate type configuration

a get/set request message :

	Length	Phase	Data		Description
功能切换：		CTL	a1 85 00 05 00 0a 02 00	...	GET LEN
	2	IN	02 00	...	..
		CTL	21 01 00 05 00 0a 02 00	...	SET CUR
	2	OUT	03 05	...	..
获取请求：		CTL	a1 85 00 03 00 0a 02 00	...	GET LEN
	2	IN	02 00	...	..
		CTL	a1 81 00 03 00 0a 02 00	...	GET CUR
	2	IN	01 06	...	..
设置请求：		CTL	a1 85 00 03 00 0a 02 00	...	GET LEN
	2	IN	02 00	...	..
		CTL	21 01 00 03 00 0a 02 00	...	SET CUR
	2	OUT	01 05	...	..

Note : For a list of supported real-time streaming types, see Appendix A.3.5 .

2. Temperature data acquisition scheme

This chapter will introduce how to acquire temperature data when the user's host device integrates a temperature measurement module . A detailed thermal imaging preview solution will be provided in the next chapter ; this chapter does not cover previewing .

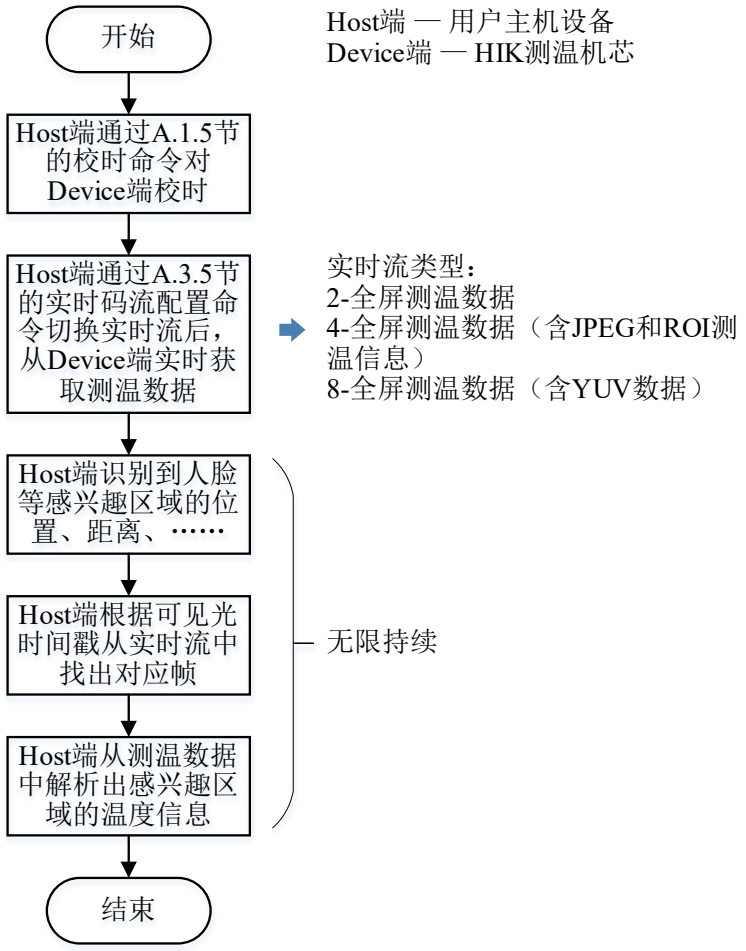
2.1 Full-screen real-time temperature measurement solution

Currently, the Device can provide three full-screen real-time temperature measurement streams via the VS interface: 2-full-screen temperature measurement matrix data , 4-full-screen temperature measurement matrix data + heatmap data, and 8-full-screen temperature measurement matrix data + YUV data . For data format and parsing details , please refer to the corresponding section in Appendix B.1 . The Device defaults to setting the real-time stream type to 5-MJ PEG real-time stream upon each power-on . The Host can switch between these types by referring to the real-time upload stream type configuration in Section 1.3.7 .

2.1.1 Applicable Scenarios

Primarily designed for scenarios with high real-time requirements, the host can acquire the temperature matrix data of all pixels on the screen of the temperature measurement module in real time , then cache it. Based on the region of interest, such as a face, detected by visible light at a certain moment , the corresponding temperature information can be found in the cached full -screen temperature measurement data.

2.1.2 Solution Flowchart



【illustrate】

(1) The temperature in the full-screen temperature measurement matrix is recommended to be the average temperature of 3*3 pixels, and the temperature of the face is the forehead area.

3. Thermal Imaging Preview Solution

This chapter will introduce how to acquire thermal imaging media data for preview when a user host integrates a temperature measurement module device . The solutions presented are based on common scenarios and are concise . Any omissions can be expanded upon and applied in practice by combining an understanding of the protocol .

3.1 Real-time streaming solution

It is primarily designed for scenarios with high real-time requirements and multiple temperature measurement targets in the scene. The host can acquire media data in real time , or simultaneously acquire media data and temperature matrix data of all pixels on the screen .

According to the real-time upload stream types listed in Appendix A.3.5 , the Device currently provides four real-time streams for preview via the VS interface: 4-Full-screen temperature measurement matrix data + heatmap data, 5-MJ PEG real-time stream, 6-YUV real-time stream , and 8-Full-screen temperature measurement matrix data + YUV data . For data format and parsing details, please refer to the corresponding chapters in Appendix B. The Device defaults to setting the real-time stream type to 5-MJ PEG real-time stream upon each power-on . The Host can switch the real-time stream type by referring to section 1.3.7 for real - time upload stream type configuration .

3.1.1 Full-screen temperature measurement + real-time thermal image streaming solution

If the host uses this scheme for previewing, the processing logic in Section 3.2 can be similar , continuously refreshing the heatmap of the preview area using the heatmap returned by the device , thus achieving dynamic preview functionality. The protocol defines the output of the rectangular area temperature measurement results in the header fields , including the highest temperature, average temperature, lowest temperature , and their coordinates for the entire screen. If these requirements are not met , the host can parse the temperature data of the area of interest from the full-screen temperature

measurement matrix data . For details on the data format and parsing of this type of real-time stream , please refer to Appendix B.1.2 .

【illustrate】

(1) The returned JPEG heatmap does not overlay temperature and rule information .

(2) Rotation and mirroring are off by default , and the resolution is 160*120. It can be configured through the video adjustment request in Appendix A.2.6 .

(3) The temperature in the full-screen temperature measurement matrix is recommended to be the average temperature of 3*3 pixels, and the temperature of the face is the forehead area.

3.1.2 Full-screen temperature measurement + Y UV real-time flow solution

This scheme is similar to the scheme shown in Section 3.3.2 , except for the media data type . The temperature data parsing and streaming are similar. The real-time temperature measurement result output defined in the header field is configured according to the temperature measurement rules in Section A.3.3 or A.3.13 , depending on the device model . For details on data format and parsing, please refer to Appendix B.1.3 .

【illustrate】

(1) The returned YUV data format is YUV422, without superimposed temperature and rule information .

(2) Rotation and mirroring are off by default , and the resolution is 160*120. It can be configured through the video adjustment request in Appendix A.2.6 .

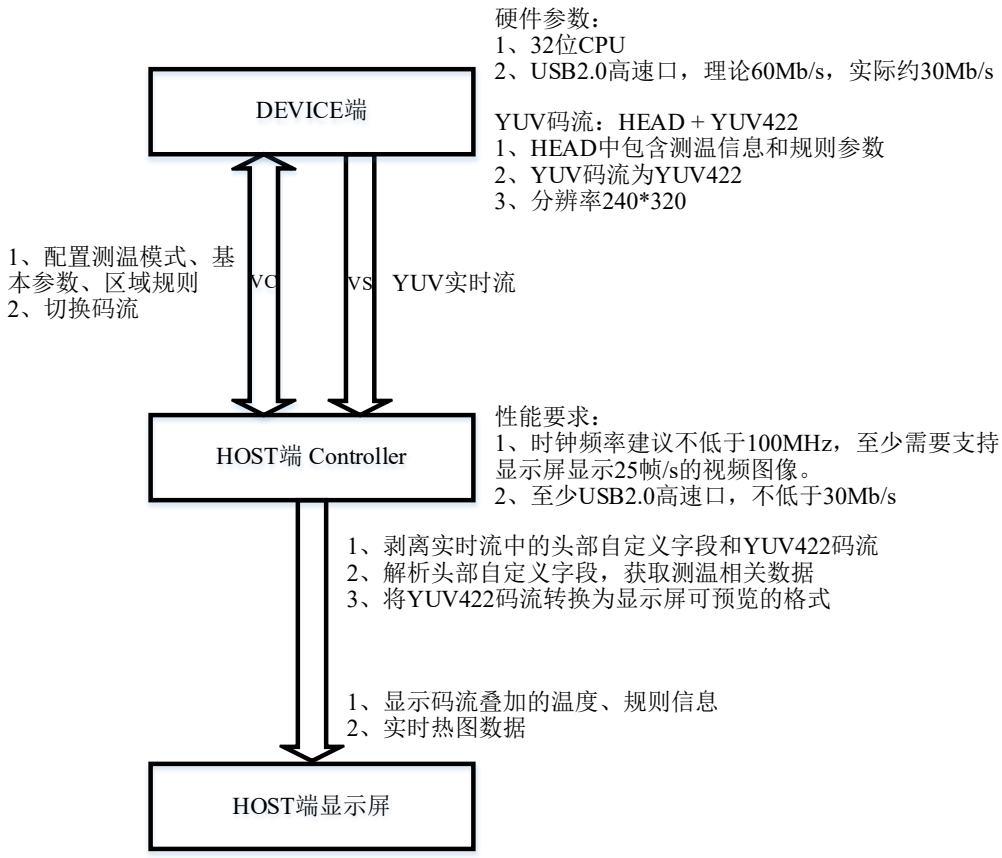
(3) The temperature in the full-screen temperature measurement matrix is recommended to be the average temperature of 3*3 pixels, and the temperature of the face is the forehead area.

3.1.3 [Industrial] YUV Real-Time Stream Acquisition Solution

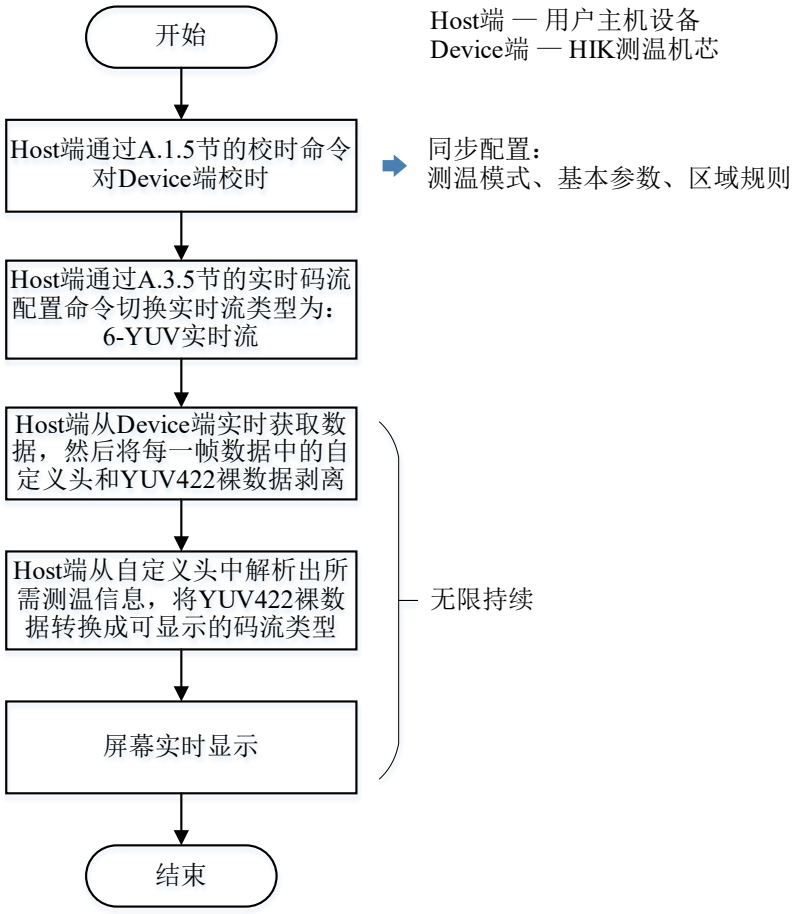
This scheme is similar to the one shown in Section 3.3.1 , where the device end

overlays temperature and zone rules , among other temperature measurement information , onto the bitstream . The host end can also choose to parse custom header fields of the real-time stream and then automatically overlay temperature and other information to suit the screen display. The temperature measurement rule configuration interface corresponding to Section A.3.13 can be used to control the output of real-time temperature information in the custom header fields. This configuration interface allows for different temperature measurement parameters to be configured for each temperature measurement target. For details on the data format and parsing of this type of real-time stream, please refer to Appendix B.2 . (Note : This scheme is only applicable to industrial models.)

3.1.3.1 Overall block diagram of the solution



3.1.3.2 Solution Flowchart



【illustrate】

(1) This type of real-time stream does not return full-screen temperature measurement matrix data .

(2) The returned YUV data format is YUV422. Whether to overlay temperature, temperature measurement rules and other information is controlled by the temperature information overlay field in the basic temperature measurement parameter configuration section of Appendix A.3.1 .

(3) Rotation and mirroring are off by default , and the resolution is 320 * 240. It can be configured through the video adjustment request

in Appendix A.2.6 .

Appendix A Function Request/Response Parameters

This protocol document only includes the industrial model . The two models may support different functions or some parameters ; please refer to the annotations in this section for details. The parameter changes between the Extension Unit 2.0 protocol and the 1.0 protocol are highlighted in red in the various lists . Please ensure compatibility when switching protocols . The channel number remains limited to 1.

No fields are reserved for each protocol in this chapter . Considering the extensibility of the protocol , the host needs to determine whether the corresponding protocol has been extended by checking the parameter length returned by the GET_LEN request before each request to obtain and send parameters . New fields for each protocol will only appear after the original parameter fields .

For parameters sent via the SET_CUR request , the host must validate each parameter according to the constraints outlined in this section before sending it . If the device fails to validate the parameters , it will not execute the request and will set error code 0x04 .

A.1 System Management (CS ID= 0x01)

Sub-function ID	definition	illustrate
0x01	SYSTEM_DEVICE_INFO	Equipment Information Acquisition
0x02	SYSTEM_REBOOT	Device restart
0x03	SYSTEM_ RESET	Restore Default
0x04	SYSTEM_HARDWARE_SERVER	Hardware service parameter configuration
0x05	SYSTEM_LOCALTIME	School Time
0x0 6	SYSTEM_UPDATE_FIRMWARE	Equipment upgrade
0 x07	SYSTEM_DIAGNOSED_DATA	Diagnostic information export

A.1.1 Device information retrieval (Sub-function ID = 0x01)

Request	GET_CUR			
Data				
Offset	Field	Size	Value type	Description
0	firmwareVersion	64	String	Main control program version
64	encoderVersion	64	String	Encoding version
128	hardwareVersion	64	String	Movement version
192	device Name	64	String	Equipment Name
256	protocolVersion	4	String	Protocol version information: " 2.0 "
260	serialNumber	64	String	Serial Number

A.1.2 Device restart (Sub-function ID = 0x02)

Request	SET_CUR			
Data				
Offset	Field	Size	Value type	Description
None				

A.1.3 Restore Default (Sub-function ID = 0x03)

Request	SET_CUR			
Data				
Offset	Field	Size	Value type	Description
None				

Note : Restore all values except network parameters. It is recommended to restart after restoring to default values .

A.1.4 Hardware Service Parameter Configuration (Sub-function ID = 0x04)

Request	GET_CUR			
Data				
O	Field	Size	Value type	Description

ffset				
0	usbMode	1	Number	USB mode switching: 1-UVC mode

Note : The F2 split type only supports UVC mode communication.

A.1. 5- time synchronization (Sub-function ID = 0x05)

Request	SET_CUR / GET_CUR			
Data				
O ffset	Fi el d	Size	Value type	Description
0	millisecond	2	Number	milliseconds 0-999
2	Second	1	Number	0-59 seconds
3	Minute	1	Number	minutes 0-59
4	Hour	1	Number	Hours 0-23
5	Day	1	Number	Japanese (not supported)
6	Month	1	Number	Month (not supported)
8	Year	2	Number	Year (not supported)
10	externalTimeSourceEnabled	1	Boolean	External time synchronization source enabled: (Not supported)

A.1.6 Device Upgrade (Sub -function ID = 0x06)

The upgrade package binary data transmission follows the big data transmission process described in Section 1.2.3 , using a SET_CUR request. The upgrade status can be retrieved via a GET_CUR request ; the Data field format of the response message for this request is as follows:

Request	GET_CUR			
Data				
O	Fi el d	Size	Value type	Description

ffset				
0	updateStatus	1	Number	Upgrade status : 0 - Reserved 1-Upgrading 2 - Upgrade successful 3 - Upgrade failed
1	percent	1	Number	Upgrade progress, value range: 0~100
2	errMsg	1	Number	upgrade failure : 0- Reserved 1- Upgrade type mismatch 2- Upgrade version incompatibility 3- Language mismatch during upgrade 0xff - Other Errors

Notice :

(1) The frequency of obtaining the upgrade status during the upgrade package transmission process should not be too frequent . It is recommended to obtain it at least once every 1 second.

(2) After the upgrade package is transmitted , it is still necessary to continue to obtain the upgrade status . The upgrade is completed based on the returned status.

(3) After the upgrade is successful , the temperature measuring core module device needs to be restarted to complete the switch of the upgrade package.

(4) GET_LEN returns 0x05, which is different from the protocol requirement. If there is a strong requirement, use ID=0x08.

A.1.7 Diagnostic Information Export (Sub-function ID = 0x07)

The diagnostic file binary data is exported using a GET_CUR request , following the big data transfer process outlined in Section 1.2.3 .

SET_CUR field has the following format:

Request	SET_CUR			
Data				
Offset	Field	Size	Value type	Description
0	dataType	1	Number	Location for exporting device data : 0-PSRAM 1-FLASH
2	address	4	Number	Export device data address:
3	length	4	Number	Exported device data length: Maximum support is 100K; exceeding this limit will result in the device not supporting it.

A.1.8 Get upgrade status (Sub -function ID = 0x08)

using Sub-function ID=0x06 to retrieve the upgrade status, GET_LEN returns 0x05, which deviates from the protocol requirements. The message length returned by this command is 0x03 . The Data field format of the response message for this retrieval request is as follows:

Request	GET_CUR			
Data				
Offset	Field	Size	Value type	Description
0	updateStatus	1	Number	Upgrade status : 0 - Reserved 1-Upgrading 2 - Upgrade

				successful 3 - Upgrade failed
1	percent	1	Number	Upgrade progress, value range: 0~100
2	errMsg	1	Number	upgrade failure : 0- Reserved 1- Upgrade type mismatch 2- Upgrade version incompatibility 3- Language mismatch during upgrade 0xff - Other Errors

Notice :

(1) The frequency of obtaining the upgrade status during the upgrade package transmission process should not be too frequent . It is recommended to obtain it at least once every 1 second.

(2) After the upgrade package is transmitted , it is still necessary to continue to obtain the upgrade status . The upgrade is completed based on the returned status.

(3) After the upgrade is successful , the temperature measuring core module device needs to be restarted to complete the switch of the upgrade package.

A.2 Image Management (CS ID= 0x02)

Sub-function ID	definition	illustrate
0x01	IMAGE _ BRIGHTNESS	Image brightness adjustment
0x02	IMAGE _ CONTRAST	Image contrast adjustment
0x03	IMAGE BACKGROUND CORRECT	One-click background correction
0x04	IMAGE_MANNUAL_CORRECT	One-click manual calibration
0x05	IMAGE _ ENHANCEMENT	Image enhancement
0x06	IMAGE _ VIDEO _ ADJUST	Video Adjustment

A. 2. 1 Image Brightness Adjustment (Sub-function ID = 0x01)

Request	SET_CUR / GET_CUR			
Data				
Offset	Field	Size	Value type	Description
0	channelID	1	Number	Channel number
1	brightness	1	Number	Image brightness : 0-100

A.2.2 Image Contrast Adjustment (Sub-function ID = 0x02)

Request	SET_CUR / GET_CUR			
Data				
Offset	Field	Size	Value type	Description
0	channelID	1	Number	Channel number
1	contrast	1	Number	Image contrast : 0-100

A. 2. 3 One -click background correction (Sub-function ID = 0x03)

Request	SET_CUR			
Data				
Offset	Field	Size	Value type	Description
0	channelID	1	Number	Channel number

Note : Please cover the lens before starting the calibration.

A. 2. 4 One -click manual calibration (Sub-function ID = 0x04)

Request	SET_CUR			
Data				
Offset	Field	Size	Value type	Description
0	channelID	1	Number	Channel number

A. 2. 5 Image Enhancement (Sub-function ID = 0x05)

Request	SET_CUR / GET_CUR			
Data				
Offset	Field	Size	Value type	Description
0	channelID	1	Number	Channel number
1	noiseReduceMode	1	Number	Digital noise reduction mode: 0 - Off 1-Normal Mode 2- Expert Mode
2	generalLevel	1	Number	Normal mode noise reduction level 0-100
3	frameNoiseReduceLevel	1	Number	Expert mode airspace noise reduction level: 0-100
4	interFrameNoiseReduceLevel	1	Number	Expert mode temporal noise reduction levels: 0-100
5	paletteMode	1	Number	Pseudo-color mode: White-hot 1 Black Hot 2 Fusion 1 10 Rainbow 1 1 Fusion 2 1 2 Iron Red 1 1 3 Iron Red 2 1 4 Dark brown 1 5 Color 1 1 6 Color 2 1 7 Ice and Fire 1 8 rain 1 9 Red Hot 20 Green Heat 2 1 Deep Blue 2 2
6	LSEDetailEnabled	1	Boolean	Image detail enhancement enabled 0 - Off

				1 - On
7	LSEDetailLevel	1	Number	Image detail enhancement level 0-100

A. 2. 6 Video Adjustment (Sub-function ID = 0x06)

Request	SET_CUR / GET_CUR			
Data				
Offset	Field	Size	Value type	Description
0	channelID	1	Number	Channel number
1	imageFlipStyle	1	Number	Mirror mode: 0 - Close 1 - Center 2 - Left/Right 3 - Up/Down
2	powerLineFrequencyMode	1	Number	Video format: 1-PAL (25 Hz) (Not compatible)
3	corridor	1	Boolean	Lens Corridor Mode (Rotation): 0 - Off 1 - On

Note : The video format is not configurable; the reserved field is for supporting existing protocols.

A. 3 Thermal Imaging Temperature Measurement Management (CS ID= 0x03)

Sub-function ID	definition	illustrate
0x01	THERMAL_THERMOMETRY_BASIC_PARAM	Temperature measurement basic parameter configuration
0x02	THERMAL_THERMOMETRY_MODE	Temperature measurement mode configuration
0x04	THERMAL_ALG_VERSION	Thermal imaging related algorithm versions
0x05	THERMAL_STREAM_PARAM	Real-time upload bitrate type configuration

0x0 E	THERMAL_THERMOMETRY_CALIBRATION_FILE	Temperature calibration file import/export
0x0 F	THERMAL_THERMOMETRY_EXPERT_REGIONS	Expert temperature measurement rule configuration
0x 10	THERMAL_THERMOMETRY_EXPERT_CORRECTION_PARAM	Expert temperature measurement calibration parameter configuration
0x 11	THERMAL_THERMOMETRY_EXPERT_CORRECTION_START	Expert temperature measurement and calibration begins

A. 3. 1 Basic Temperature Measurement Parameter Configuration (Sub-function ID = 0x01)

Request	SET_CUR / GET_CUR			
Data				
Offset	Field	Size	Value type	Description
0	channelID	1	Number	Channel number
1	enabled	1	Boolean	Enable temperature measurement function
2	displayMaxTempEnabled	1	Boolean	Display highest temperature: 0 - Off 1 - On
3	displayMinTempEnabled	1	Boolean	Display lowest temperature: 0 - Off 1 - On
4	displayAverageTempEnabled	1	Boolean	Average temperature display: 0 - Off 1 - On
5	temperature unit	1	Number	Temperature units: 1 - Celsius , 2 - Fahrenheit, 3 - Kelvin

				(The protocol specifies that temperature in Celsius is used as the unit of transmission.)
6	temperatureRange	1	Number	Temperature measurement range: 1- (30~45) 2- (-20~150) 3- (0~400)
7	calibrationCoefficientEnabled	1	Boolean	Enable calibration coefficient 0 - Off 1 - On
8	calibrationCoefficient	4	Number	Calibration factor: 0.00~30.00, actual value during transmission * 100, converted to an integer.
12	externalOpticsWindowCorrection	4	Number	External optical temperature: -40.0~80.0 ℃ during transmission (actual value +100) * 10 converted to a positive integer.
16	emissivity	4	Number	Emittance: 0.01~1 (accurate to two decimal places). The actual value during transmission will be multiplied by 100 and converted to an integer.
20	distanceUnit	1	Number	Distance units:

				1- meter 2- centimeter 3- foot
twenty one	distance	4	Number	distance: 0.3-2m (human body) 0.3-3 m (industrial) (Accurate to one decimal place. The protocol specifies cm as the unit of transmission.)
2 5	reflectiveEnable	1	Boolean	Reflection temperature enable: 0 - Off , 1 - On
2 6	reflectiveTemperature	4	Number	Reflection temperature: - 100.0~1000.0 ℃ (accurate to one decimal place). During transmission, (actual value + 100) * 10 is converted to a non-negative integer.
30	thermometryInfoDisplayposition	1	Number	Location of temperature measurement information display: 1 - Follow the rules 2 - Top left corner of the screen
31	thermometryStreamOverlay	1	Number	Temperature information superimposed on

				the bitstream 1 - No stacking 2 - Superposition
32	alert	4	Number	Warning temperature: -20 ℃ to 400 ℃ (actual value + 100) * 10 converted to a positive integer during transmission.
33	alarm	4	Number	Alarm temperature: -20 ℃ to 400 ℃ (when transmitting, the actual value is converted to a positive integer by multiplying by 10 by 10).
34	externalOpticsTransmit	4	Number	External optical transmittance: 0.01~1 (accurate to two decimal places) Convert the actual value to an integer

				by multiplying it by 100 during transmission.

Notice :

(1) External optical temperature setting and calibration coefficient enable functions are mutually exclusive.

(2) The reflected temperature is only for human body models. For industrial models, it is configured in the expert temperature measurement rules in section A.3.13 .

(3) The temperature measurement range is only the first level for human body models and only the second and third levels for industrial models .

(4) Distance : 0.3 to 2 meters for human body models and 0.3 to 3 meters for industrial models .

(5) Distance unit : The protocol stipulates that the distance is transmitted in centimeters. If the unit received by the Device is not centimeters, the distance value will be converted to centimeters according to the unit received .

(6) Whether the preview bitstream is overlaid with information such as temperature and temperature measurement rules is controlled by the last bitstream overlay field of the parameters .

A. 3. 2 Temperature Measurement Mode Configuration (Sub-function ID = 0x02)

Request	SET_CUR / GET_CUR			
Data				
Offset	Field	Size	Value type	Description
0	channelID	1	Number	Channel number
1	thermometryMode	1	Number	Temperature measurement mode 1 - Normal 2 - Expert
2	thermometryROIEnabled	1	Boolean	Temperature measurement ROI enable not

				supported) 0 - Off 1 - On

Note : Temperature measurement ROI enable is only available for human models; F2 devices do not support it .

A.3.4 Thermal Imaging Related Algorithm Version (Sub-function ID = 0x04)

Request	GET_CUR			
Data				
Offset	Field	Size	Value type	Description
0	thermometryAlgVer	64	String	Temperature measurement algorithm version information

A.3.5 Real - time upload stream type configuration (Sub-function ID = 0x05)

Request	SET_CUR / GET_CUR			
Data				
Offset	Field	Size	Value type	Description
0	channelID	1	Number	Channel number
1	stream Type	1	Number	Real-time upload stream type: 1-Raw thermal imaging data (not supported) 2-Full-screen temperature measurement data 3-14 bit real-time raw thermal imaging data 4- Full-screen temperature measurement data + JPEG and ROI temperature measurement information (not supported) 5-MJPEG live stream (not supported)

				6-YUV Real-time Streaming 7-PS encapsulated MJPEG real-time stream (not supported) 8-Full-screen temperature measurement data + YUV real-time stream 9-YUV+ raw data (new addition) 10 - YUV only, without temperature sensor (new) (default)

Note : This configuration controls the real-time stream type of the VS interface . For details on the real-time stream data format and parsing, please refer to Appendix B.

A. 3. 12 Temperature Measurement Calibration File Import/Export (Sub-function ID = 0x0E)

Temperature calibration files are imported into the Device using a SET_CUR request, and exported from the Device using a GET_CUR request. Both import and export processes follow the big data transfer workflow described in Section 1.2.3 .

Request	SET_CUR / GET_CUR			
Data				
Offset	Field	Size	Value type	Description
0	fileName	64	String	File Name
64	fileLength	4	Number	File length
68	calibrationFile	calibrationFile Length	Byte	Binary data of temperature calibration file

A. 3. 13 [Industrial] Expert Temperature Measurement Rule Configuration (Sub-function ID = 0x0F)

Setting expert temperature measurement rules uses a SET_CUR request, and retrieving expert temperature measurement rules uses a GET_CUR request. Both setting and retrieving follow the big data transmission process described in Section 1.2.3 .

Request	SET_CUR / GET_CUR			
Data				
Offset	Field	Size	Value type	Description
0	channelID	1	Number	Channel number

1	SID	1	Number	Scene ID, currently limited to 1
2	regionNum	1	Number	Total number of rule regions
rule zones, which are modules that appear in a loop and support a maximum of 21 zone configurations.				
3	regionID	1	Number	Region ID, incrementing from 1
4	e nabled	1	Boolean	Zone Enable 0 - Off 1 - On
5	name	32	String	Rule Name
37	emissivity	4	Number	Emittance: 0.01~1 (accurate to two decimal places). The actual value during transmission will be multiplied by 100 and converted to an integer.
41	distance	4	Number	Distance: 0.3-3 m (The protocol specifies that cm is used as the unit for transmission)
45	reflectiveEnable	1	Boolean	Reflection temperature enable: 0 - Off , 1 - On
46	reflectiveTemperature	4	Number	Reflection temperature: -100.0 ~1000.0 ℃ (accurate to one decimal place). During transmission, (actual value + 100) * 10 to convert to a non-negative integer .
50	type	1	Number	Rule labeling type: 1 point 2 -line 3 - box
51	s how AlarmColorEnabled	1	Boolean	Alarm color display

				enabled 0 - Off 1 - On
5 2	r ule	1	Number	Alarm temperature comparison method : When type is 1 - point: 1- Average temperature greater than 2 - Average temperature less than When type is 2-line : 1-High temperature greater than 2-High temperature less than 3- Low temperature greater than 4- Low temperature less than 5 - Average temperature greater than 6 - Average temperature less than When type is 3-box: 1 - High temperature greater than 2 - High temperature less than 3 - Low temperature greater than 4 - Low temperature less than 5 - Average temperature greater

				than 6 - Average temperature less than 7 - Temperature difference greater than 8 - Temperature difference less than
5 3	alert	4	Number	Warning temperature: -20 ℃ to 400 ℃ (actual value + 100) * 10 converted to a positive integer during transmission.
5 7	alarm	4	Number	Alarm temperature: -20 ℃ to 400 ℃ (when transmitting, the actual value is converted to a positive integer by multiplying by 10 by 10).
61	pointNum	1	Number	Total number of vertices in the region : When type is 1 , the number is 1. The number is 2 when type is 2- line . When type is 3-box: the number is 3-10 .
is a list of vertex coordinates for the region , with each vertex appearing cyclically, and the number of occurrences is pointNum.				
6 2	p ointX	4	Number	X-coordinate, normalized to 0-1000, with the origin at the top left.
6 6	p ointY	4	Number	Y-coordinate, normalized to 0-1000, with the origin at the top left.

A. 3. 14 [Industrial] Expert Temperature Measurement Calibration Parameter Configuration (Sub-function ID = 0x10)

Request	SET_CUR / GET_CUR			
Data				
O ffset	Fi el d	Size	Value type	Description
0	channelID	1	Number	Channel number
1	distance	4	Number	over a distance of 0.3-3 meters , the protocol uses centimeters as the unit.
5	envTemperature	4	Number	When transmitting at ambient temperatures ranging from -273.0 to -1000.0 , convert the actual value (plus 300) * 10 to an integer.
9	emissivity	4	Number	Transmit rate: 0.01-1.00, actual value during transmission * 100, converted to an integer.
13	pointNum	1	Number	Number of temperature points : 4
is a list of temperature point information , with the following module appearing cyclically, and the number of occurrences is pointNum.				
14	i d	1	Number	The serial number increases from 1.
15	presetTemperature	4	Number	When transmitting at a preset temperature of -40.0 to 650.0 ℃ , the actual value is converted to a

				positive integer by multiplying by 10 by 10 .
19	p ointX	4	Number	X-coordinate, normalized to 13 - 993 , with the origin at the top left.
twenty three	p ointY	4	Number	Y-coordinate, normalized to 17 - 991 , with the origin at the top left.

Note : Setting temperature points id = 1~3 takes at least 16 seconds to complete , and id = 4 takes at least 11 seconds , for a total time of approximately 1 minute. It is recommended to set four temperature points sequentially or simultaneously .

A. 3. 15 [Industrial] Expert temperature calibration initiated (Sub-function ID = 0x11)

Request	SET_CUR			
Data				
O ffset	Fi el d	Size	Value type	Description
0	channelID	1	Number	Channel number

Note : The calibration parameters must be reset before each calibration begins!

A. 3. 16 Overall temperature rise calibration (Sub-function ID = 0x12)

Request	SET_CUR/GET_CUR			
Data				
Offset	Field	Size	Value type	Description
0	channelID	1	Number	Channel number
1	enabled	1	Boolean	Enable temperature rise settings 0 - Off 1 - On
2	type	1	Number	Methods for obtaining temperature rise parameters

				0: Automatically obtain 1: Manual input
3	result	1	Number	Temperature rise parameter calculation status (read-only, automatic acquisition mode is valid) 0 - Success 1- Waiting 2- Failure
4	envTemperature	4	Number	Ambient temperature: -99.0~99.0 ℃ (accurate to two decimal places) During transmission, (actual value + 100) * 10 is converted to a positive integer.
8	coefficient	4	Number	Temperature rise coefficient, -10~10
12	maxTemperatureRise	4	Number	Maximum temperature rise, 2~20
16	coldStartRate	4	Number	Cold start temperature rise rate: 0.01~0.5 (accurate to two decimal places) Convert the actual value to an integer by multiplying it by 100 during transmission.
20	coldStartRise	4	Number	Cold start temperature rise: -3.0 to 3.0 (accurate to one decimal place)

				During transmission, (actual value + 100) * 10 is converted to a positive integer.

Appendix B Real-time stream data format

B.1 Full-screen real-time temperature measurement stream

Based on the full-screen temperature measurement matrix data , the device supports outputting the following based on different usage scenarios : 2 - Full-screen temperature measurement matrix data , 4 - Full- screen temperature measurement matrix data + heatmap data, and 8 - Full-screen temperature measurement matrix data + YUV data . For the latter two types of real-time full-screen temperature measurement streams , compared to the first type, firstly, they include heatmap and YUV data respectively ; secondly , the header fields define information such as the highest, average, and lowest temperatures and their coordinates . Their real- time temperature measurement rules are configured according to the temperature measurement rule commands in Section A.3.3 or A.3.13. However, currently, neither the heatmap nor the YUV data supports overlaying temperature and region rule information ; the host can overlay these manually.

In the full-screen temperature measurement matrix data , the temperature data of each pixel is transmitted using two or four bytes . When transmitting using four bytes, it can be directly forcibly converted to float .

fTmVal = *(float*)(pData + dataIndex)

When transmitting using two bytes , the data conversion relationship is as follows:

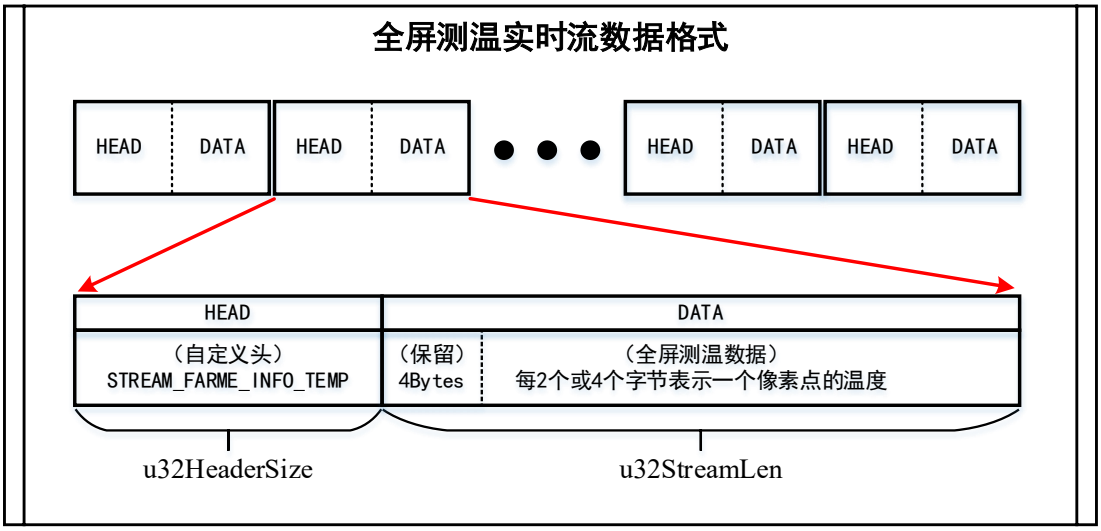
fTmVal = (*(unsigned short*)(pData + dataIndex)) / (float)u32TmScale +

u32TmOffset - 273.15

Among them: (1) p Data is a pointer to the full-screen temperature measurement data , and dataIndex is the pointer offset.

(2) u32TmScale and u32TmOffset are member variables of the STREAM_FS_SUPPLE_INFO_TEMP structure , which is used to add information to the real-time temperature data in the header fields of the real-time stream in subsequent sections . The u32TmDataMode member of this structure indicates the temperature data transmission format for each pixel .

B.1.1 Full - screen temperature measurement real-time streaming data format



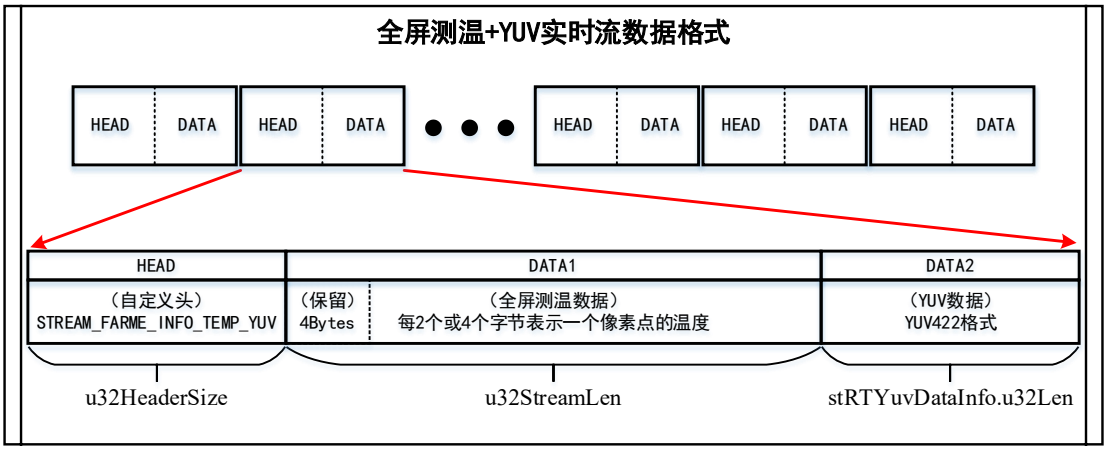
The header field data structure is as follows (132 bytes) :

Mode 1 - Full-screen temperature measurement real-time data header field data structure (132 bytes)					
Offset	Field	Size (bytes)	Value	Description	Source
0	u32MagicNo	4		0x70827773 ASCII code for "FRMI"	SOC
4	u32HeaderSize	4		HEAD length*/	SOC
8	u32StreamType	4		Data type RTData: see STREAM_TYPE_E*/	SOC
12	u32StreamLen	4		DATA length (including the first 4 bytes) */	SOC
16	u32RTDataType	4		1-14-bit raw data; 2-Full-screen temperature measurement results data; 3-YUV data*/	SOC
20	u32FrmNum	4		Frame sequence number:	logic

				The real-time uploaded frame sequence number is a continuously increasing even number, such as 1000, 1002, 1004, 1006.	
twenty four	u32StdStamp	4		DS relative timestamp (displayed as 0)	none
28	year	2		Year (displayed as 0)	none
30	month	2		Month (displayed as 0)	none
32	dayOfWeek	2		What day of the week (displayed as 0)	none
34	day	2		Day (displayed as 0)	none
36	hour	2		Hour*/	none
38	minute	2		minute*/	none
40	second	2		Second*/	none
42	milliSecond	2		millisecond*/	none
44	u32Width	4		Raw data width	SOC
48	u32Height	4		Raw data high	SOC
52	u32Len	4	98304	The length of the real-time temperature measurement matrix is in bytes, plus 4 bytes, such as a 4-byte float temperature measurement matrix: 256*192*2	SOC (Fixed)
56	u32Fps	4	25	Real-time upload frame rate - currently fixed at 25	SOC (Fixed)

60	u32Chan	4	0	Channel number, fixed at 0	SOC (Fixed)
64	u32TmDataMode	4	1	0 represents 4 bytes, 1 represents 2 bytes.	SOC (Fixed)
68	u32TmScale	4		Temperature measurement scaling	SOC (Fixed)
72	u32TmOffset	4	0	Temperature measurement offset is currently fixed at 0.	SOC (Fixed)
76	bylsFreezedata	4		Whether the data is frozen: 1 - frozen, 0 - not frozen	logic
124	res[12]	48	0	Reserved bytes	SOC (Fixed)
128	u32CrcVal	4		The structure checksum verifies the data preceding the structure.	SOC

B.1.3 Full-screen temperature measurement + Y UV real-time stream data format



The header field data structure is as follows (4636 bytes) :

	Field	Size (bytes)	Value	Description	Source
	u32MagicNo	4		0x70827773 ASCII code for "FRMI"	SOC
	u32HeaderSize	4		HEAD length*/	SOC
	u32StreamType	4		Data type RTData: See STREAM_TYPE_E	SOC
	u32StreamLen	4		DATA length (including the first 4 bytes) */	SOC
	bIFRYuv	4		Does it include YUV images?	SOC
STREAM_FS_SUPPLE_INFO_TEMP Real-time Temperature Data Additional Information Structure	u32TmDataMode	4	1	0 represents 4 bytes, 1 represents 2 bytes.	SOC (Fixed)
	u32TmScale	4		Temperature measurement scaling	SOC (Fixed)
	u32TmOffset	4	0	Temperature measurement offset is currently fixed at 0.	SOC (Fixed)
	bylsFreezedata	4		Whether the data is frozen: 1 - frozen, 0 - not frozen	logic
STREAM_RT_DATA_INFO_S Real-time	u32RTDataType	4		1-14-bit raw data; 2-Full-screen temperature measurement results data; 3-YUV data*/	SOC
	u32FrmNum	4		Frame sequence number:	logic

temperature data output structure				The real-time uploaded frame sequence number is a continuously increasing even number: 1000, 1002, 1004, 1006.	
	u32StdStamp	4	0	DS relative timestamp (displayed as 0)	none
	year	2	0	Year*/ (displayed as 0)	none
	month	2	0	Month*/ (displayed as 0)	none
	dayOfWeek	2	0	0: Sunday ~ 6: Saturday (displayed as 0)	none
	day	2	0	/ (displayed as 0)	none
	hour	2	0	Hour*/	none
	minute	2	0	minute*/	none
	second	2	0	Second*/	none
	milliSecond	2	0	millisecond*/	none
	u32Width	4		raw data width*/	SOC
	u32Height	4		Raw data high*/	SOC
	u32Len	4	98304	The length of the real-time temperature measurement matrix is in bytes, plus 4 bytes, such as a 4-byte float temperature measurement matrix: 256*192*2	SOC (Fixed)
	u32Fps	4	25	Real-time upload frame rate - currently fixed at 25	SOC (Fixed)
	u32Chan	4	1	Channel number, fixed at 1	SOC (Fixed)

RT_YUV_DATA_INFO_S is the output structure of real-time YUV.	u32FrmNum	4		Frame sequence number: The real-time uploaded frame sequence number is a continuously increasing even number: 1000, 1002, 1004, 1006.	logic
	u32Width	4		YUV data width	SOC
	u32Height	4		High YUV data	SOC
	u32Len	4		YUV data length (DATA2)	SOC
	u32StdStamp	4	0	DSP relative timestamp*/	none
	year	2	0	Year*/	none
	month	2	0	moon*/	none
	dayOfWeek	2	0	0: Sunday ~ 6: Saturday	none
	day	2	0	day*/	none
	hour	2	0	Hour*/	none
	minute	2	0	minute*/	none
	second	2	0	Second*/	none
	milliSecond	2	0	millisecond*/	none
The IFR_REALTIME_TM_OUTCOME_UPLOAD_INFO output structure for real-	tempUnit	4		Temperature unit, default is Celsius, mainly used for uploading.	SOC (Surveying Temperature Algorithm)
	refTempkey	1		Reflective temperature switch	SOC (Surveying

time temperature measurement information, supporting 21 rules for industrial applications and 10 rules for human health.					Temperature Algorithm)
	res[3]	3		reserve	SOC (Surveying Temperature Algorithm)
	f32Distance	4		Measuring distance	SOC (Surveying Temperature Algorithm)
	refTemp	4		Reflected temperature, based on the reflected temperature switch	SOC (Surveying Temperature Algorithm)
	emissionRate	4		Emittance, [0.00, 1.0]	SOC (Surveying Tempe

					rature Algorithm)
	envTemp	4		Ambient temperature; when there is no temperature sensor, it is obtained from the temperature measurement algorithm library.	TMP
	minTmp	4		Lowest temperature across the entire screen	OSD
	maxTmp	4		Full screen highest temperature	OSD
	avrTmp	4		Full screen average temperature	OSD
	IFR_POINT*3 (Stores the coordinates of the highest and lowest temperatures in the test results, normalized to 0-1000, array index: 0-highest temperature, 1-lowest temperature, 2-average temperature)				
	x	4			OSD (To be determined)
	y	4			OSD (To be determined)
	u32TempMode	4		0: Expert temperature measurement, 1: Regular	SOC

				temperature measurement	
	resv[5]	20		reserve	SOC
	pointNum	4		Maximum of 10 temperature measurement points.	SOC
	boxNum	4		Maximum number of temperature measuring units: 10	SOC
	lineNum	4		Maximum number of line temperature sensors: 1	SOC
	total	4		The sum of the above three	SOC
	reserved[8]	32		reserve	SOC
	IFR_OUTCOME_INFO*21 (Temperature measurement result)				
	enable	1		//<Enable: 0-No, 1-Yes>	SOC (Surveying Temperature Algorithm)
	regionalld	1		//<Region ID	SOC (Surveying Temperature Algorithm)
	reserved[2]	2		//<Reserved fields	SOC (Surve

					ying Tempe rature Algorit hm)
	reftemp	4			SOC (Surve ying Tempe rature Algorit hm)
	reservedex[2 0]	1		//<Reserved fields	SOC (Surve ying Tempe rature Algorit hm)
	f32Distance	4		//< distance	SOC (Surve ying Tempe rature Algorit hm)
	u8res	1		//<Reserved fields	SOC (Surve ying

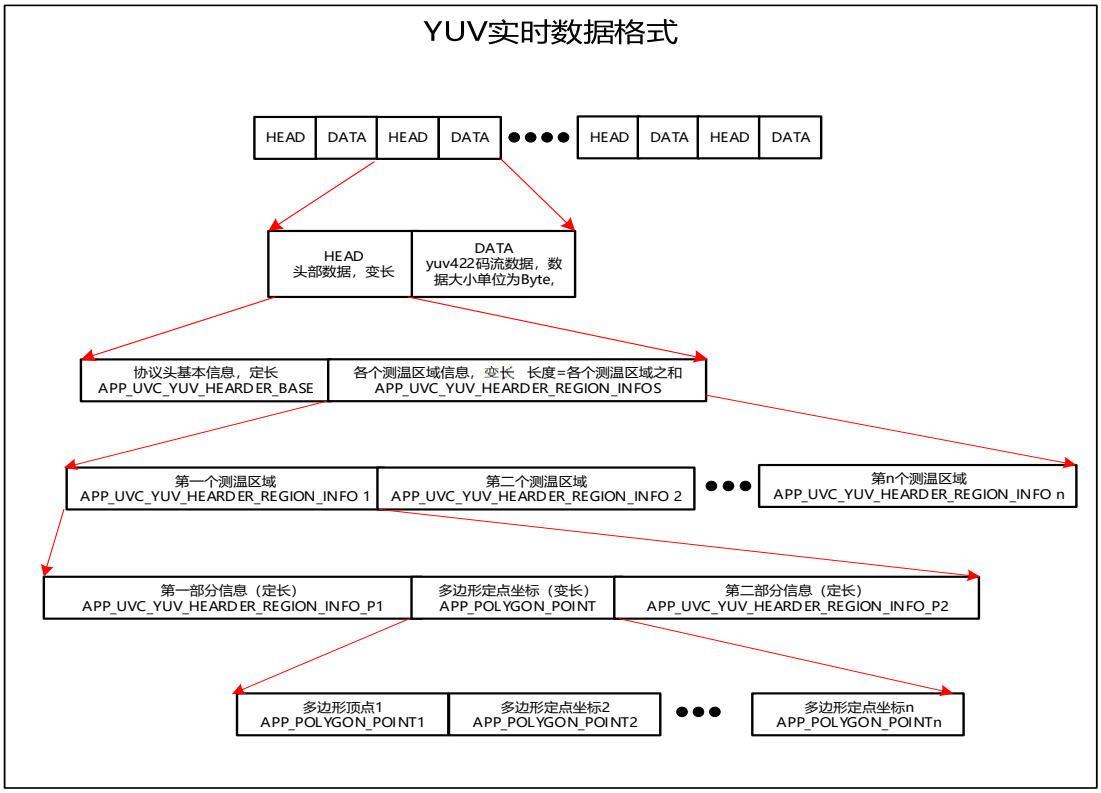
					Temperature Algorithm)
	u8maxTmpStat	1		//For the highest temperature status, see TM_TEMP_STAT_E	SOC (Surveying Temperature Algorithm)
	u8minTmpStat	1		//For the lowest temperature status, see TM_TEMP_STAT_E	SOC (Surveying Temperature Algorithm)
	u8avgTmpStat	1		//Average temperature status can be found in TM_TEMP_STAT_E	SOC (Surveying Temperature Algorithm)
	regiontype	4		Region Type	SOC (Surveying Tempe

					rature Algorithm)
	name[32]	32		Area Name	SOC (Surveying Temperature Algorithm)
	emission rate			//<Emittance: [0.00, 1.00]	SOC (Surveying Temperature Algorithm)
	minTmp			//<Minimum temperature: [-40.0, 1000.0], in degrees Celsius	OSD
	maxTmp			//<Maximum temperature: [-40.0, 1000.0], in degrees Celsius	OSD
	avrTmp			Average temperature: [-40.0, 1000.0], in degrees Celsius	OSD
	diffTmp			//<Temperature difference: [0.0, 1040.0], in degrees Celsius	OSD (To be determ

					ined)
	IFR_POINT*2 (saves the coordinates of the highest and lowest temperatures in the test results, normalized to 0-1000, array index: 0-highest temperature, 1-lowest temperature)				
	x	4			OSD
	y	4			OSD
	pointNum	4		The actual number of vertices of the polygon (polygon region)	SOC (temperature measurement algorithm) (to be determined)
	IFR_POINT*12 (polygon region) coordinates				
	x	4			SOC (temperature measurement algorithm) (to be determined)
	y	4			SOC (tempe

					rature measurement algorithm) (to be determined)
	uploadType	1	1	Real-time data upload types - Full-screen temperature measurement real-time upload data types: 0-JPEG; 1-YUV	SOC (Fixed)
	res[11]	44	0	Reserved	SOC
	u32CrcVal	4		The structure checksum verifies the data preceding the structure.	SOC

B. 2 YUV Real-Time Streaming Data Format



B. 2.1 Complete data structure

Order	Fi el d	Size	Value type	Description
0	YUV_Data_Magic_Head	4	Number	The head magic letter is specified as 0x050508e7
1	HeaderLen	4	Number	The header data length, in bytes, is currently set at 136, but may be increased in the future. It is recommended to obtain it dynamically.
2	TotalDataLen	4	Number	All data lengths, including header data, are in bytes; the length is dynamic and dynamically obtained.
3	channelID	1	Number	Channel number
4	reserved	3	Number	Reserve 3 bytes
5	Year	2	Number	Year (displayed as 0)
6	Month	2	Number	Months 1-12 (displayed as

				0)
7	Day of Week	2	Number	What day of the week? (Displayed as 0)
8	Day	2	Number	Day 1-31
9	Hour	2	Number	Hours 0-23
10	Minute	2	Number	minutes 0-59
11	Second	2	Number	0-59 seconds
12	millisecond	2	Number	milliseconds 0-999
13	TMInfoPosition	1	Bool	Location of temperature measurement information display: 0 - Follow the rule (default) 1-Top left corner display
14	bShowMaxTemp	1	Bool	Display the highest temperature: 0 - Do not display (default), 1 - Display
15	bShowMinTemp	1	Bool	Display the lowest temperature: 0 - Do not display (default), 1 - Display
16	bShowAvgTemp	1	Bool	Display average temperature: 0 - Do not display (default), 1 - Display
17	tempUnit	1	Number	temperature unit 0 degrees Celsius (default) 1- Fahrenheit 2- Kelvin
18	tempMode	1	Number	Temperature measurement mode: 0 - Expert temperature measurement, 1 - Standard temperature measurement
Number of temperature measurement areas				
19	pointTmpNum	1	Number	Maximum of 10 temperature measurement points.
20	boxTmpNum	1	Number	The maximum number of temperature measurement boxes is

				10.
twenty one	lineTmpNum	1	Number	Maximum number of line temperature sensors: 1
twenty two	total	1	Number	The sum of the above three elements currently supports a maximum of 21. For future scalability, this number should not be limited.
twenty three	reserved1	18	Number	18 bytes reserved
The temperature measurement results list appears in a loop, currently supporting a maximum of 21, the number of which is determined by the total above.				
twenty four	RegionID	1	Number	Expert temperature measurement area ID, starting from 1 and incrementing by 1.
25	RegionEnabled	1	Bool	Zone Enable 0 - Off 1 - On
26	regiontype	1	Number	Region Type 0 - No region type 1 - Polygonal Region Type 2 - Linear Region Type 3 - Point Area Type
27	Point_Num	1	Number	The number of valid vertices is limited to a maximum of 10 valid vertex coordinates.
Within a single temperature measurement rule, the vertex coordinates of the temperature measurement rule are determined here based on the Point_Num above, which specifies how many vertex coordinates to distribute.				
28	Point_X	4	Number	The current vertex's horizontal X-axis coordinate, normalized to 0-1000
29	Point_Y	4	Number	The current vertex's vertical Y-axis coordinate, normalized to 0-1000
30	emissionRate	4	Number	Emittance - Floating-point number: [0.00, 1.00]

31	distance	4	Number	distance 0.3-3m The protocol transmits data in centimeters, and users parse the data in centimeters.
32	reflectiveTemperature	4	Number	Reflection temperature: -100.0~1000.0 ℃ (accurate to one decimal place; during transmission, (actual value + 100) * 10 will be converted to a non-negative integer).
33	minTmp	4	Number	Minimum temperature: [-40.0, 1000.0], in degrees Celsius.
34	minTmp_X	4	Number	The lowest temperature horizontal X-axis coordinate, normalized to 0-1000
35	minTmp_Y	4	Number	The lowest temperature vertical Y-axis coordinate, normalized to 0-1000
36	maxTmp	4	Number	Maximum temperature: [-40.0, 1000.0], in degrees Celsius.
37	maxTmp_X	4	Number	The horizontal X-axis coordinate of the highest temperature is normalized to 0-1000.
38	maxTmp_Y	4	Number	The highest temperature vertical Y-axis coordinate, normalized to 0-1000
39	avrTmp	4	Number	Average temperature: [-40.0, 1000.0], in degrees Celsius.
40	YUVData		byte	YUV bitstream data, containing only the current frame