5. Bluetooth Ymodel Service

5.1 General

The Ymodel Bluetooth communication protocol (hereinafter referred to as this Agreement) is used for file transfer between apps and devices. We define the transmission service and three data transmission channels, and the smart device communicates and transmits data with the Xinger APP by implementing the necessary services and channels.

5.2 Bluetooth Services and Features

This Agreement is based on an open service channel, and the UUID of the service channel is: 0x6E400001-B5A3-F393-E0A9-E50E24DCCA9E Under this service, we define three basic characteristics to complete data communication and command control.

Characteristic name	UUID	description	attribute
command	0x6E400004-B5A3-F393-E0A9-E50E24DCCA9E	Command features, which are used to send and receive commands between the app and the hardware	Read&Write
Acceptance of data	0x6E400003-B5A3-F393-E0A9-E50E24DCCA9E	It is used for APP to accept hardware data	Notify
Send data	0x6E400002-B5A3-F393-E0A9-E50E24DCCA9E	It is used by the APP to send data to the intelligent hardware	Write

5.3 Control Commands

5.3.1 Command Definitions

Command UUID 0x6E400004-B5A3-F393-E0A9-E50E24DCCA9E, the communication between mobile phone and smart device commands, used to initiate transmission requests, control and status representation, and the attributes include NOTIFY and WRITE.

- The maximum length is 20 bytes
- Includes checksums, contents, and check sum values (XORs)
- Check XOR Value (XOR)
- This structure is used regardless of whether the phone is Write to the device or the device is Notify to the phone

Opcode	Concent	Checksum（XOR）
1Byte	n Byte (n <= 18)	1Byte

Command structure table

5.3.2 Command Table

The name of the command	Code	description
Idle state command	0x04	IDLE
File fetch command	0x05	The APP sends a file to the smart hardware request
The file gets the status in transit	0x06	The smart hardware is transferring files to the APP
File send command	0x07	The smart hardware requests the app to send files
The status of the file being sent in transit	0x08	The APP is transferring files to the smart hardware
Stop the transfer command	0x1F	APP actively stops the file transfer command
Error status	0x11	Device error: Command format is incorrect/Command validation error/Unsupported command
Error status	0x12	Device error: This file is not available
Error status	0x13	Device error: Out of memory
Error status	0x14	Device error: Non-idle state
Error status	0x15	Device error: File parsing failed
Error status	0x1F	If the timeout device does not respond, the command is sent for 10 seconds, or the command is sent after a timeout
Status commands	0xFF	Get the current status of a smart hardware device

Content: The content is usually the file name, such as panel.json, **.json *** .txt. If it is empty, make up one byte 0x00.

- Note: Send panel.json to get the watch face layout configuration.

Checksum: The XOR check value (XOR) of the command and content from 1 to n-1.

5.3.3 Check whether the device status is idle in the FF command process (command delivery instance).
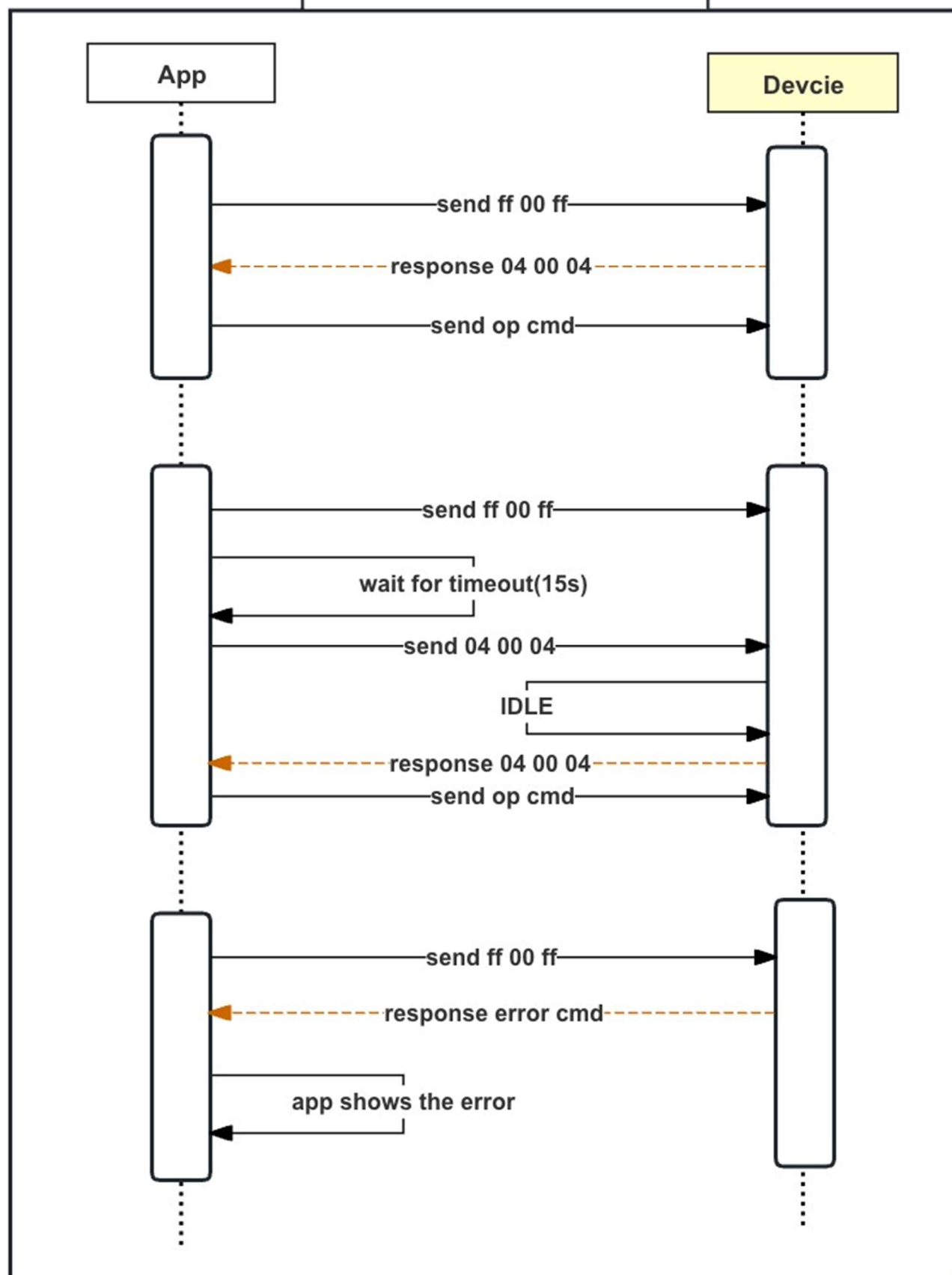
1. Normal process: The app issues a ff command, the device is idle, and the 04 command is replied, indicating that the device is idle and other commands can be executed, and the app continues to issue other commands

2. Timeout: The App issues a ff command, waiting for a timeout (15s), if it times out, the App continues to issue the 04 command, the device receives the 04 command, terminates the current transmission task, returns to the idle state, and then replies to 04, the App receives 04, and continues to execute other commands

3. Error: The App issues a ff command, the device replies with an incorrect command (0x11-0x15 as defined in the command table), and the App prompts an error

- Note: The ff command is a request device status command; 04 The command is IDLE, and the idle command is the idle command (when the App is delivered, it is to make the device idle); Error command 0x11-0x15 refer to Command Table 4.2

設備状態検査時序图

App | Devcie

send ff 00 ff

response 04 00 04

send op cmd

send ff 00 ff

wait for timeout(15s)

send 04 00 04

IDLE

response 04 00 04

send op cmd

send ff 00 ff

response error cmd

app shows the error

As shown in the figure, all interactions are enumerated and executed before the file transfer process

5.4 File Transfers

5.4.1 Transfer Process

In order to ensure the efficient and stable transmission of files, it is based on the Ymodem protocol, and the data area size is fixed at 128 bytes (expandable to 1024), and the specific packet format is as follows (because of 20 of BLE byte limit, so you need to do consolidation and unpacking processing when transmitting and receiving).

Packet starts	Serial number	Serial number complement	Data area	CRC high bytes	CRC low bytes
SOH	01	FE	....	...	....
1Byte	1Byte	1Byte	128Byte	1Byte	1Byte

A complete handshake process for protocol transmission

APP to get the smart device file

Step	Smart hardware	Phone APP
1		Command:0x05 2015.fit
2	Command:0x06 2015.fit	
3		Send(0002 Send Data): 'C'
4	Send(0003 send data): SOH 00 FF "2015.fit" "xxx" CRC CRC	
5		Send:ACK
6		Send:'C'
7	Send:SOH 01 FE Data[128] CRC CRC	
8		Send:ACK
	Send:SOH 02 FD Data[128] CRC CRC	
		Send:ACK
	Send:SOH 03 FC Data[128] CRC CRC	
		Send:ACK
	Send:SOH 01 FE Data[100] NUL[28] CRC CRC	
		Send:ACK
9	Send:EOT	
10		Send:NAK
11	Send:EOT	
12		Send:ACK
13	Command:0x04	

The app sends data packets to the smart device

Step	Phone APP	Smart hardware
1	Command(0001):0x07 "setting.json"	
2		Command(0001):0x08 "setting.json"
3		Send(send data 0003): 'C'
4	Send (send data 0002): SOH 00 FF "setting.json" "xxx" CRC CRC	
5		Send:ACK
6		Send:'C'
7	Send:SOH 01 FE Data[128] CRC CRC	
8		Send:ACK
	Send:SOH 02 FD Data[128] CRC CRC	
		Send:ACK
	Send:SOH 01 FC Data[128] CRC CRC	
		Send:ACK
	Send:SOH 01 FE Data[100] NUL[28] CRC CRC	
		Send:ACK
9	Send:EOT	
10		Send:NAK
11	Send:EOT	
12		Send:ACK
13		Command(0001):0x04 /0x15

File transfer process

Smart Hardware: Sender

Phone APP: Recipient

After sending the file that needs to be retrieved Command The first step is for the receiver to send a character 'C' to start the transmission After the sender receives 'C', it sends the first frame of the packet, which is as follows:

```
SOH 00 FF  "2015.fit"   "xxx"  CRC
```

Byte 1 SOH: indicates that the packet has a data area size of 128 bytes.

Byte 2 00: Numbered, the first packet is 00, the second packet is 01, and the third packet is 02 in turn. After reaching FF, the cycle continues to increase from 0.

Byte 3 FF: The inverse of the number. The number 00 corresponds to FF, 01 corresponds to FE, and so on.

Byte 4 to the last two bytes: Byte 1 is SOH means there are 128 bytes, and in STX it is 1024 bytes, which is divided into data zones. "2015.fit" filename, followed by file xxx, after the file name and file size, if it is less than 128 bytes, fill it with 0.

Last two bytes: Only the data part participates in the check CRC, excluding the header and encoding parts, the 16-bit CRC validation, the high byte first, and the low byte last.

After the receiver receives the first frame of the packet, it sends an ACK to reply correctly.

Then send a character 'C', above the receiver just received a file name, now officially start the file transfer, after the sender receives 'C', start to send the second frame, the data in the second frame is stored in the first packet of data of the file. After the receiver receives the data verification, it sends an ACK Then wait for the next packet to finish transmitting the data and continue with ACK answering. Until all data transfers are complete.

If the data verification fails, the NAK will be returned, and the sender will resend it, and if the number of retransmissions exceeds 5 times, it will directly exit the sending If the last packet of data is less than 128 bytes, the sender will send EOT after the data transmission is completed, and the first receiver will reply with NAK for a second confirmation. Once the sender receives the NAK, it resends it EOT, the second time the receiver receives a termination, it responds with an ACK.

The sender must wait for the receiver's ACK or NCK before sending the next packet, and if the sender does not receive the ACK/NCK within 5 seconds or retransmits more than 5 times, the sender will directly exit the transmission, and the sender will set the Command status to idle.

The state constants are defined as follows:

state	Code	description
SOH	0x01	start of 128-byte data packet
STX	0x02	start of 1024-byte data packet
EOT	0x04	end of transmission
ACK	0x06	acknowledge
NAK	0x15	negative acknowledge
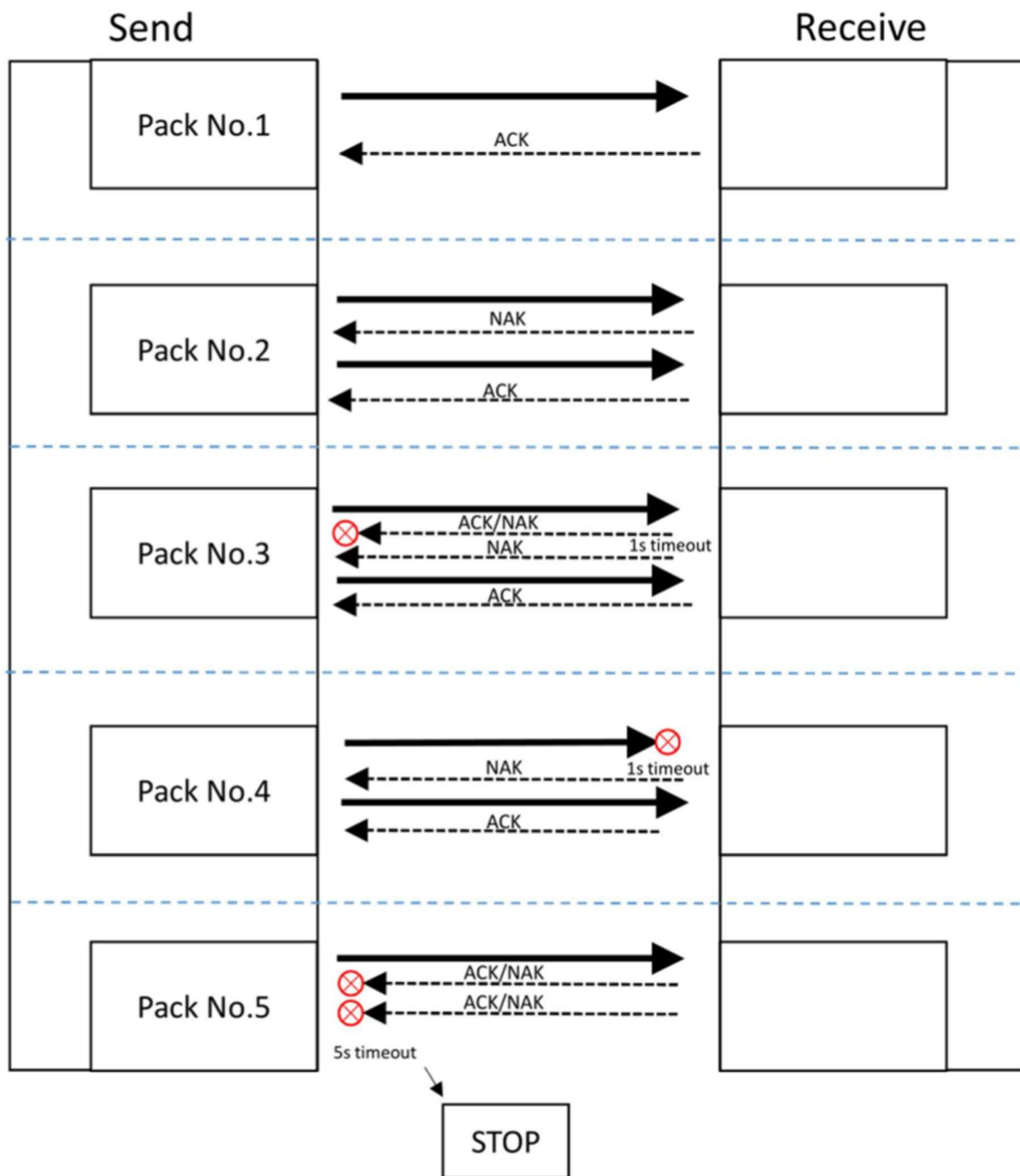
5.7 Bluetooth 4.2/5.0 extended support

Since 4.2, the Bluetooth organization supports the extension of the packet length, and the packet transmission up to 244MTU can be customized. This section uses new extensions to improve the number of bytes per packet to deliver data transfer efficiency.

- LE Data Packet Length Extension (BT v4.2)
- LE 2M PHY (BT v5.0)

Enable DLE

1. Both master and slave devices need to support Bluetooth 4.2 and above, and iPhone devices need iPhone 6 and above. Android depends on the device
2. The master-slave device supports the LE Data Packet Length Extension link-layer feature
3. The master-slave device initiates the DLE-supported data length update process by sending a LL_LENGTH_REQ command that completes the handshake process with a LL_LENGTH_RSP return. During the data length update process, the master and slave negotiate the following four values:

1. The number of payload bytes that connMaxTxOctets can send in a single link-layer packet

2. The length of time (in microseconds) that connMaxTxTime can actively transmit a single link-layer packet

3. The number of payload bytes that connMaxRxOctets can receive in a single link-layer packet

4. The length of time (in microseconds) that connMaxRxTime can actively receive a single link-layer packet

After the handshake, the hardware device will send the maximum number of MTUs that can respond to the mobile phone.
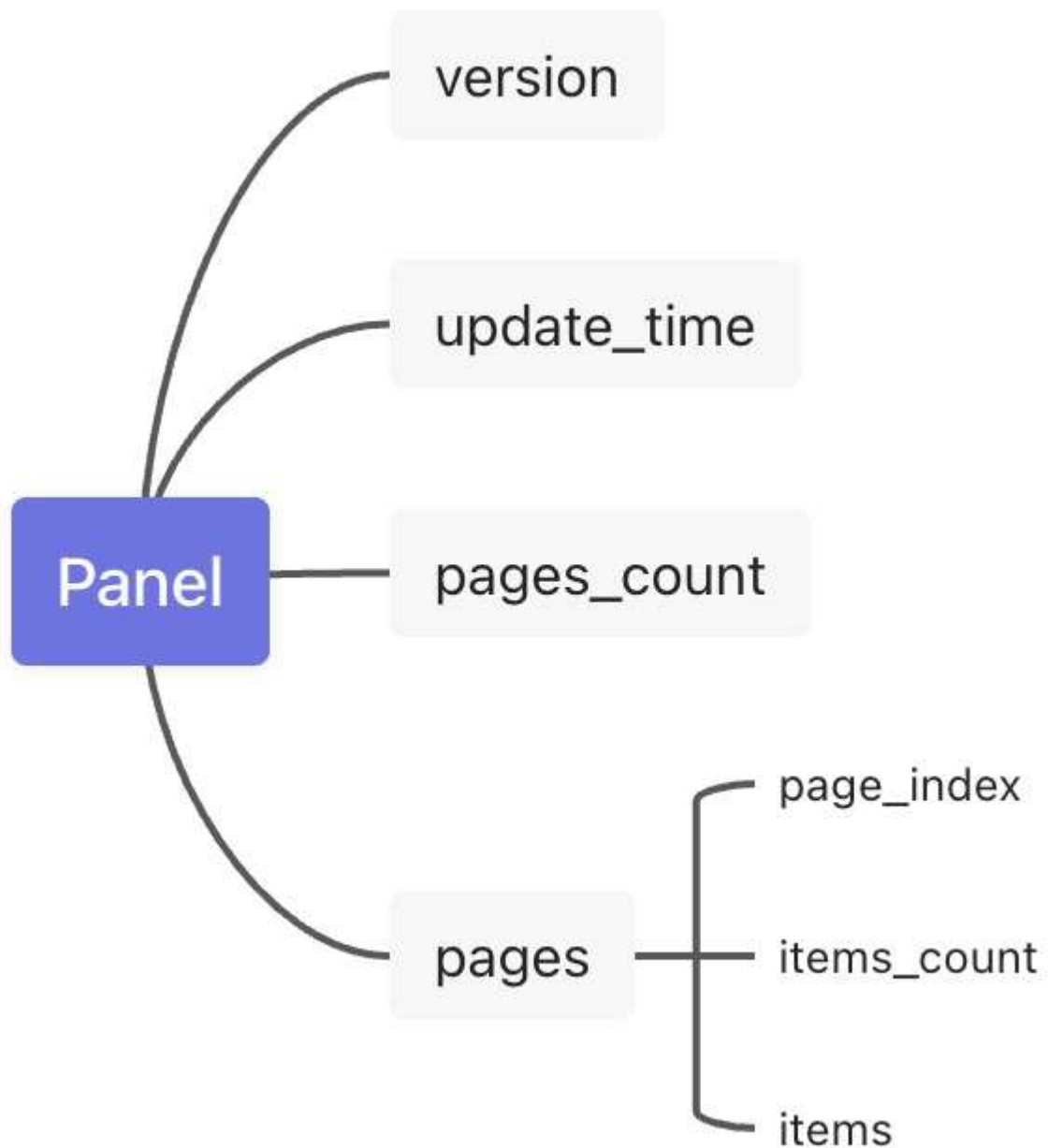
5.8 User Information Files

user_profile.json

Object	Type	Default Value	Range	Description
MAXHR	Uint	180	0~255	Maximum heart rate

sample:

```
{
  "version": "2.0.0",
  "device_model": "XL",
  "sn": "xxxx.xxxx",
  "updated_at": 1233676623,
  "user_profile": {
    "MAXHR": 180
  }
}
```

5.9 Watch face settings file

Page definition page

Object	Type	Default Value	Description
index	Uint		Page indexing
pid	Uint	1	Page mode
items	[item]		An array of watch face elements
count	Uint		The number of dial elements in an array

pid	item_count	Description
0	1	Only one piece of data is displayed in the watch face
1	2	Two pieces of data are displayed in the watch face

The item in the watch face is defined as follows: Default: 5 pages, 1-2 data per page. Users can customize the data displayed in the app. Factory default 5 pages: clock, real-time speed, exercise

time, sports mileage, navigation information Real-time speed, exercise time, exercise mileage, real-time heart rate, real-time cadence, navigation information, altitude, real-time slope, cumulative climb, current clock, up to 5 configurations

item id	type	title	Chinese name	default value	unit	description	size limit
100	1	Moving Time	Exercise time	0	s	second	
103	1	Current Time	The current clock	0	s	clock	
200	2	Distance	distance	0.0	m	m/km/mi	
300	3	Speed	Real-time speed	0.0	m/s		
400	4	Elevation	elevation	0	m	m/ft	
403	4	Grade	Real-time slope	0.0	%		
404	4	Elevation Gain	Cumulative climb	0.0	m		
500	5	Heartrate	Real-time heart rate	0	BPM		
600	6	Cadence	Real-time cadence	0	RPM		
801	8	Navi Instruction	Navigation guides	""	String		

sample:

```
{
  "version": "2.0.0",
  "device_model": "XL",
  "sn": "xxx.xxx",
  "updated_at": 1233676623,
  "pages_count": 3,
  "pages": [
    {
      "index": 0,
      "pid": 0,
      "count": 1,
      "items": [
        300
      ]
    },
    {
      "index": 1,
      "pid": 1,
      "count": 2,
      "items": [
        200,
        400
      ]
    },
```

```json
        {
            "index": 2,
            "pid": 0,
            "count": 1,
            "items": [
                500
            ]
        }
    ]
}
```

Factory documentation

```json
{
    "version": "2.0.0",
    "device_model": "XL",
    "sn": "xxx.xxx",
    "updated_at": 1718782574,
    "pages_count": 5,
    "pages": [
        {
            "index": 0,
            "pid": 0,
            "count": 1,
            "items": [
                103
            ]
        },
        {
            "index": 1,
            "pid": 0,
            "count": 1,
            "items": [
                300
            ]
        },
        {
            "index": 2,
            "pid": 0,
            "count": 1,
```

```json
      "items": [
        100
      ]
    },
    {
      "index": 3,
      "pid": 0,
      "count": 1,
      "items": [
        200
      ]
    },
    {
      "index": 4,
      "pid": 0,
      "count": 1,
      "items": [
        801
      ]
    }
  ]
}
```