Dorpo TCP/IP Control Protocol V1.1

Prepared by	Date	Version	Applicable Models	Remarks: [Key changes and reasons]
Peter	2023-03-15	V1.0	X3	New Document
Peter	2024-08-27	V1.1	X3 UHP-3001 X300	Added Status/Event Notification Interface

2. IP Control Parameter Passing/Interface Example

(1) Device Connection Code Example (HTTP Port 9527):

API: http://192.168.*.*:9527/doopoo/connect ?uniqueId={uniqueId} & from={from} &ip={ip}
Request Method: Get
uniqueId : Client Unique Identifier
from:Source: From C4 end, transmit "C4" directly
The outgoing parameter has been authorized (200) : json={


```
    "code": 0,
    "data": {
        "bluetooth_mac_address": "02:ad:3e:01:4b:23",
        "device_model": "doopoo X3",
        "device_name": "doopoo X32222",
        "device_uniqueId": "911c7bda4c6525",
        "id": 0,
        "ip": "192.168.**",
        "is_allowed": 1
    },
    "msg": "success"
```

}

bluetooth_mac_address : Server address

device_mode : Device Mode

device_name : Device Name

device_uniqueId : Client Unique Identifier

Ip : IP Address

is_allowed : Authorization status 1Authorized Unauthorized 1 Unauthorized

Unauthorized : The HTTP response code is 401

Special Note: During connection, simultaneously listen to the TCP server (port number 9528). After authorization confirmation, the output parameter will be in JSON format, as shown in the following example:


```
{
    "clientAck": "911c7bda4c6525da",
    "deviceModel": "doopoo X3",
    "deviceName": "doopoo X32222",
    "from": "doopoo_service",
    "ipAddress": "192.168.**",
    "macAddress": "02:ad:3e:01:4b:25",
    "msgId": "msg_search_device",
    "playStatus": 0,
    "playType": 0,
"authCode": "1",
    "uniqueId": "1d5a901803ec3b37"
}
```


authCode : “1” Authorization successful “0” Authorization revoked

(2) Example of Sending Key Code Values (HTTP Port 9527):

API: http://192.168.*.*:9527/doopoo/sendKey ?action={action} & from={from}&keyValue={keyValue}

Request Method Specification: Get

action : Key Name (Mandatory field, refer to the key name reference at the end)

keyValue : Key Value (Required field, refer to the key value reference at the end)
from: Source (cannot be empty) For example: Transmit from port c4 to port c4.
Response Parameters : json={
    "code": 0,
    "msg": "success"
}

(3) WOL Network Wakeup Code Example

public static final int DEFAULT_PORT = 9; //Default port number
public static final int DEFAULT_TIMEOUT_MILLIS = 10000; // Default timeout period
public static final int DEFAULT_NO_PACKETS = 5; // default packets number

```
 /**
      * Send a Wake-On-Lan packet
      *
      * @param address          - InetAddress to send wol packet to
      * @param macStr           - MAC address to wake up
      * @param port             - port to send packet to
      * @param timeoutMillis - timeout (millis)
      * @param packets          - number of packets to send
      *
      * @throws IllegalArgumentException - invalid ip or mac
      * @throws IOException - error sending packet
      */
     public static void sendWakeOnLan(final InetAddress address, final String macStr, final int port, final int timeoutMillis, final int packets) throws IllegalArgumentException, IOException {
          if (address == null) throw new IllegalArgumentException("Address
```

```
cannot be null");
        if (macStr == null) throw new IllegalArgumentException("MAC
Address cannot be null");
        if (port <= 0 || port > 65535) throw new
IllegalArgumentException("Invalid port " + port);
        if (packets <= 0)
            throw new IllegalArgumentException("Invalid number of
packets to send " + packets);

        byte[] macBytes = MacTools.getMacBytes(macStr);
        byte[] bytes = new byte[6 + 16 * macBytes.length];
        for (int i = 0; i < 6; i++) {
            bytes[i] = (byte) 0xff;
        }
        for (int i = 6; i < bytes.length; i += macBytes.length) {
            System.arraycopy(macBytes, 0, bytes, i, macBytes.length);
        }

        DatagramPacket packet = new DatagramPacket(bytes, bytes.length,
address, port);

        for (int i = 0; i < packets; i++) {
            DatagramSocket socket = new DatagramSocket();

            socket.setSoTimeout(timeoutMillis);

            socket.send(packet);
            socket.close();
        }
    }
```

(4) Specify to play the video (port number 9527 on the HTTP side):

API: http://192.168.*.*:9527/doopoo/play ?videoPath={videoPath} & from={from}

way to request: Get

videoPath : Absolute path of the video (needs to be URL-encoded; otherwise, the Chinese path will result in garbled characters)

The following are examples of absolute video paths for different storage devices:

NFS Device Path Demonstration :

/data/data/doopooexplorer/nfs/ 192.168.31.5 #Movie file / ISO / ISO - Death Squad 4K Dolby Vision/SHOOTER_4K_ULTRA_HD.iso

SMB device path demonstration:

/data/data/doopooexplorer/samba/ 192.168.31.5 #CloudNAS/CloudDrive2/115/movie file/SGNB329/The.Great.Wall.2016.ULTRAHD.BluRay.2160p.HEVC.Atmos.TrueHD.7.1-sGnB@CHDBits.iso

System built-in storage path example:

/storage/emulated/0/Movie2/4K+ATMOSThe Greatest Showman : The Greatest Show.mkv

Example of an absolute path for a local USB drive/hard disk:

/storage/048B-24DB/PT/Spider-Man1280X688(23.976fps)(AC-3 6ch 48kHz 16bit).mkv

Note:Red text indicates fixed/static characters.Blue text represents dynamic content.

from:Source Transmit from C4 to C4 directly.

Response Parameters : json={

```
    "code": 0,
    "msg": "success"
}
```

(5) Intelligent Reminder Service (HTTP Port: 9527)

API: http://192.168.*.*:9527/doopoo/remind ?content={content} & from={from}

Request Method: Get

content : Reminder Content (URL Encoded)

from:Source: Transmit from C4 to C4 directly.

Response Parameters : json={

```
"code": 0,
"msg": "success"
```

}

(6) Player Media Status Interface (TCP Port: 9528)

Connecting to TCP Service via Device IP and Port

Request Parameter Input Method(No Parameters Required (Optional)): json

```
{
    "clientAck": "bda4c6525da",
    "deviceModel": "PGCM10",
    "deviceName": "OPPO K9x 5G",
    "from": "c4",
    "ipAddress": "192.168.**",
    "msgId": "handle_shake",
    "macAddress": "02:08:06:04"
}
```

clientAck : Client Unique Identifier

deviceModel : Device Mode

deviceName : Device Name

from : source

ipAddress : Client IP address

msgId : Message ID (sent during connection: handle_shake)

macAddress : Client MAC Address

Response Parameters :

```
{
    "clientAck": "911c7bda4c6525da",
    "deviceModel": "doopoo X3",
    "deviceName": "doopoo X32222",
```

```
    "from": "doopoo_service",
    "ipAddress": "192.168.**",
    "macAddress": "02:ad:3e:01:4b:25",
    "msgId": "msg_search_device",
    "playStatus": 0,
    "playType": 0,
    "uniqueId": "1d5a901803ec3b37"
}
```

clientAck : Client Unique Identifier

deviceModel : Device Mode

deviceName : Device Name

from : source

ipAddress : Server IP Address

msgId : Message ID (Service Returns play_status)

macAddress : Server MAC Address

uniqueId : Server Unique Identifier

playType : Playback Device Type 1Video Player 2 Music Player

playStatus : Playback Status -1Stopped 1Play 2 Pause

(7) Player online status heartbeat (TCP port number 9528):

Connect to the TCP service using the box's IP address and the specified port number.

The request parameter format is as follows: json

```
{
    "clientAck": "bda4c6525da",
    "deviceModel": "PGCM10",
    "deviceName": "OPPO K9x 5G",
    "from": "c4",
    "ipAddress": "192.168.**",
    "msgId": "heart_beat",
    "macAddress": "02:08:06:04"
}
```

clientAck : Client Unique Identifier

deviceModel : Device Mode

deviceName : Device Name

from : source

ipAddress : Client IP Address Parameter

msgId : Message ID(heart_beat)

macAddress : Client MAC Address Parameter

Response Parameters :


```
{
    "clientAck": "911c7bda4c6525da",
    "deviceModel": "doopoo X3",
    "deviceName": "doopoo X32222",
    "from": "doopoo_service",
    "ipAddress": "192.168.**",
    "macAddress": "02:ad:3e:01:4b:25",
    "msgId": "msg_search_device",
    "playStatus": 0,
    "playType": 0,
    "uniqueId": "1d5a901803ec3b37"
}
```


clientAck : Client Unique Identifier

deviceModel : Device Mode

deviceName : Device Name

from : source

ipAddress : Server IP Address

msgId : Message ID (Service Returns play_status)

macAddress : Server MAC Address

uniqueId : Server Unique Identifier

playType : Playback Device Type 1Video Player 2 Music Player

playStatus : Playback Status -1Stopped 1Play 2 Pause

Note: The heartbeat polling is implemented on the client side, sending messages to the server at a defined frequency. A response from the server indicates that the status remains online.

3. IP Remote Control Command Table (Port Number: 9527)

The following commands directly map to the corresponding functions of the remote control. When the player receives these commands, it is the same as receiving commands from the remote control.

NO.	Key Name (KeyEvent.java Key Name)	Sent Code (Decimal)	Function Description
1	Power (KEYCODE_POWER)	26 (Invalid)	Turn the player on/off (Use WOL wake-up code for power-on)
2	Wake (KEYCODE_WAKEUP)	224 (Invalid)	Power on (Use WOL wake-up code for power-on)
3	Sleep (KEYCODE_SLEEP)	223	Power off
4	Mute (KEYCODE_VOLUME_MUTE)	164	Mute or unmute
5	Up (KEYCODE_DPAD_UP)	19	Navigation: Up
6	Down (KEYCODE_DPAD_DOWN)	20	Navigation: Down
7	Right (KEYCODE_DPAD_RIGHT)	22	Navigation: Left
8	Left (KEYCODE_DPAD_LEFT)	21	Navigation: Right
9	Confirm (KEYCODE_DPAD_CENTER)	23	Navigation: Confirm
10	Back (KEYCODE_BACK)	4	Return to previous level
11	Info (KEYCODE_INFO)	165	Information
12	Home (KEYCODE_HOME)	3	Return to home screen
14	Volume Up (KEYCODE_VOLUME_UP)	24	Increase volume

15	Volume Down (KEYCODE_VOLUME_DOWN)	25	Decrease volume
16	Play/Pause (KEYCODE_MEDIA_PLAY_PAUSE)	85	Play or pause playback
17	Play (KEYCODE_MEDIA_PLAY)	126	Start playback
17	Pause (KEYCODE_MEDIA_PAUSE)	127	Pause playback
19	Menu (KEYCODE_MENU)	82	Options/Menu
20	Settings (KEYCODE_SETTINGS)	176	System settings
21	1 (KEYCODE_1)	8	1
22	2 (KEYCODE_2)	9	2
23	3 (KEYCODE_3)	10	3
24	4 (KEYCODE_4)	11	4
25	5 (KEYCODE_5)	12	5
26	6 (KEYCODE_6)	13	6
27	7 (KEYCODE_7)	14	7
28	8 (KEYCODE_8)	15	8
29	9 (KEYCODE_9)	16	9
30	0 (KEYCODE_0)	7	0
32	Resolution (KEYCODE_UI)	289	Resolution
33	Subtitle (KEYCODE_CAPTIONS)	175	Subtitle
34	Audio Track (KEYCODE_MEDIA_AUDIO_TRACK)	222	Audio track
35	Page Up/Previous (KEYCODE_MEDIA_PREVIOUS)	88	Page up / Previous track
36	Page Down/Next	87	Page down / Next track

	(KEYCODE_MEDIA_NEXT)		
37	Red (KEYCODE_PROG_RED)	183	Red
38	Green (KEYCODE_PROG_GREEN)	184	Green
39	Yellow (KEYCODE_PROG_YELLOW)	185	Yellow
40	Blue (KEYCODE_PROG_BLUE)	186	Blue
41	Video (KEYCODE_MOVIE)	290	Video
42	Music (KEYCODE_MUSIC)	209	Music
43	Photo (KEYCODE_PHOTO)	291	Photo
44	File (KEYCODE_FILE)	292	File
45	Delete (KEYCODE_DEL)	67	Delete
46	Stop (KEYCODE_MEDIA_STOP)	86	Stop
47	Fast Forward (KEYCODE_MEDIA_FAST_FORWARD)	90	Fast forward
48	Rewind (KEYCODE_MEDIA_REWIND)	89	Rewind