

****Modification History****

****20220531 version changes****

****Abandoned mAPI.installApkSilently("/sdcard/API.apk"); function****

****Add mAPI.installApkSilently("/sdcard/API.apk","package name","class name"); automatically launch the application based on the package name and class name**

Interface used**

****20220621 Classification Interface Description****

****20220625 Added interfaces for setting serial number, setting/getting device model, setting/getting Ethernet MAC address, and setting/getting IMEI number ****

****20220630 Added Ethernet switch and status acquisition****

****20220701 Add APP guard interface****

**** 20220713 Added the ability to set startup pictures and animations Set the drop-down status bar state****

****20220830Get screen on/off status****

****20220922 Modify the timer power on/off interface and add several modes****

****20221107Add APP foreground guard interface****

****20230413 Added separate control screen display switch****

****20240321 Modify the description of the Wiegand asynchronous read interface function****

API Interface Description

- The API interface described in this document corresponds to our products. You need to check our hardware specifications to determine the supported hardware interface.
- For the related interface call effects, you can go to 'Settings' - 'User Settings', or 'Settings' - 'User Settings' - 'YNHCommonAPI' and select the corresponding function item test.
- If the document parameters are incomplete or missing, please refer to the YNHCommonAPI source code for details.

1: Get the interface instance

Example code:

```
// Get the unique instance of the process
YNHAPI mAPI = YNHAPI.getInstance();
```

Note: All sample codes that follow use this instance object.

2: System parameters

2.1: Get the motherboard model

Example code:

```
// Get the motherboard model , return the specific board model such as ynh960, ynh530,
etc. String model = mAPI.getBoardModel();
```

2.2: Serial Number

2.2.1: Get the serial number

Example code:

```
// Get the serial number.  
String sn = mAPI.getSerialNo(); //The returned result is consistent with Build.SERIAL
```

2.2.2: Set the serial number

Example code:

```
// Set the serial number. After setting, you need to call the restart function to make it effective after restarting  
String SerialNo = "12345678"; mAPI.setSerialNo(SerialNo);  
mAPI.reboot();
```

2.3: Ethernet MAC address

2.3.1: Get Ethernet MAC address

Example code:

```
// Get the Ethernet MAC address.  
String mac = mAPI.getEthernetMAC(); //The returned result is consistent with the Ethernet returned by the network service module
```

2.3.2: Set Ethernet MAC address

Example code:

```
// Set the Ethernet MAC address. After setting, you need to call the restart function for it to  
take effect after restarting. String mac = "00:E0:EC:37:2A:42";  
mAPI.setEthernetMAC(mac); mAPI.reboot();
```

2.4: 4G module IMEI

2.4.1: Get 4G module IMEI

Example code:

```
// Get the IMEI of the 4G module.  
String imei = mAPI.getIMEI(); //The returned result is consistent with the Ethernet returned by the network service module
```

2.4.2: Set 4G module IMEI

Example code:

```
// Set the IMEI of the 4G module. After setting, you need to call the restart
function, and it will take effect after restarting.
String imei = "864793030776509";
mAPI.setIMEI(imei); mAPI.reboot();
```

2.5: Device Model

2.5.1: Get device model

Example code:

```
// Get the device model.
String model = mAPI.getProductModel(); //The returned result is consistent with Build.MODEL
```

2.5.2: Set the device model

Example code:

```
// Set the device model. After setting, you need to call the restart
function, and it will take effect after
restarting. String model = "123456";
mAPI.setProductModel(model); mAPI.reboot();
```

2.6: Device storage information

Related data class description:

```
//Device storage information (path, space
size, etc.) public class StorageInfo{ public
    final static int TYPE_MEMORY = 1; public final static int
    TYPE_LOCAL_STORAGE = 2; public final static int TYPE_TFCARD = 3;
    public final static int TYPE_USB_STORAGE = 4; private int mType
    = 0; private long mTotalSize = 0; //Unit KB private long mFreeSize = 0; //
    Unit KB private String mPath = "";

}
```

Example code:

```
// Get the storage information of the device, including internal storage (EMMC), TF/SD card,
USB storage (U disk, mobile hard disk, etc.) List<StorageInfo> list
= mAPI.getStorageInfos(); for(StorageInfo
info : list){ }
```

2.7: Set the boot image

Example code:

```
mAPI.setBootLogo("/sdcard/boot.bmp"); //The boot image must be an 8-bit bmp bitmap
```

2.8: Set the boot animation

Example code:

```
mAPI.setBootAnimation("/sdcard/bootanimation.zip"); //The boot animation must be made with standard Android animation , and
the name of the zip archive can be modified at will
```

3: OTA upgrade

```
void otaUpdate(String otaFilePath)
```

parameter:

- otaFilePath : ota package path.

Note: The upgrade package must be in the sdcard directory and named update.zip

Example code:

```
mAPI.otaUpdate("/sdcard/update.zip");
```

4: APK

4.1: Application starts automatically at boot

```
void setBootLaunchApk(String packageName, boolean launch)
```

parameter:

- packageName : The name of the application package that needs to be started automatically at boot.
- launch : true, indicates automatic startup. false, indicates automatic startup is canceled.

Example code:

```
mAPI.setBootLaunchApk("com.innohi.helloworld", true);
```

#

4.2: Silently install APK and automatically start after installation

```
void installApkSilently(String apkPath,String packageName,String className)
```

parameter:

- apkPath : The path of the APK.
- packageName : The full package name of the APK to be launched. For example, the APK package name is com.example
- className : The full class name of the APK to be launched. For example, the APK class name is com.example.MainActivity

Example code:

```
// Silently install
```

```
APK mAPI.installApkSilently("/sdcard/API.apk","com.example","com.example.MainActivity ");
```

If the last two parameters are null , it means that there is no need to automatically pull up the silent installation application

```
mAPI.installApkSilently("/sdcard/API.apk",null,null);
```

4.3: Uninstall APK

```
void uninstallApkSilently(String packageName)
```

parameter:

- packageName : The package name of the APK.

Example code:

```
mAPI.uninstallApkSilently("com.innohi.api");
```

4.4: Guardian APK

A: Non-foreground guard

```
void setAppKeepLive(String packageName, int keepAliveTimeSec)
```

The reboot is not retained, that is, the APK needs to be called again after the reboot

parameter:

- packageName : The package name of the APK.
- keepAliveTimeSec: APK detection timeout, in seconds. If the APK is not detected alive after this time, the APK will be manually pulled up.

Example code:

```
mAPI.setAppKeepLive("com.innohi.api", 10); //Cancel
non-foreground
guard mAPI.setAppKeepLive("", 1);
```

B: Front desk guard

```
void setForegroundAppKeepLive(String packageName, int keepAliveTimeSec)
```

The reboot is not retained, that is, the APK needs to be called again after the reboot

parameter:

- packageName : The package name of the APK.
- keepAliveTimeSec: APK detection timeout, in seconds. If the APK is not detected alive after this time, the APK will be manually pulled up.

Example code:

```
mAPI.setForegroundAppKeepLive("com.innohi.api", 10); //Cancel
foreground
guard mAPI.setForegroundAppKeepLive("", 1);
```

5: Display related

5.1: Get the screen rotation angle

```
RotationDegree getScreenRotation(ScreenType screenType)
```

parameter:

- screenType : Indicates which screen's rotation angle is to be obtained. For example, ScreenType.MAIN indicates the main screen, and ScreenType.AUX indicates the secondary screen.

Return value:

- RotationDegree: Returns the rotation angle of the specified screen. For example, RotationDegree.DEGREE_90 means 90 degrees.

Example code:

```
YNHAPI.RotationDegree screenRotation
    = mAPI.getScreenRotation(YNHAPI.ScreenType.MAIN);
if (screenRotation == YNHAPI.RotationDegree.DEGREE_90) { Log.d(TAG,
    "Main screen 90 degrees");
}
```

5.2: Set the screen rotation angle

```
void setScreenRotation(ScreenType type, RotationDegree degree)
```

parameter:

- type : Sets the screen rotation angle. For example, ScreenType.MAIN indicates the main screen.
- degree : The angle to be set. For example, RotationDegree.DEGREE_90 means 90 degrees.

Example code:

**// Set the secondary
screen to rotate 180**

```
degreesmAPI.setScreenRotation(YNHAPI.ScreenType.AUX, YNHAPI.RotationDegree.DEGREE_180);
```

5.3: Get screen density

LcdDensity getLcdDensity()

Return value:

- LcdDensity : Screen density. For example, LcdDensity.DENSITY_160 means the screen density is 160 dpi.

Example code:

```
YNHAPI.LcdDensity density = mAPI.getLcdDensity(); if (density ==  
YNHAPI.LcdDensity.DENSITY_160) { Log.d(TAG, "Screen  
density is 160dpi");  
}
```

5.4: Setting screen density

void setLcdDensity(LcdDensity density)

parameter:

- density: Indicates the screen density that needs to be set. For example, LcdDensity.DENSITY_160 means setting the screen density to 160 dpi.

Example code:

**// Set the screen density
to 240dpi** mAPI.setLcdDensity(YNHAPI.LcdDensity.DENSITY_240);

5.5: Get the screen on/off status

boolean isScreenOn()

parameter:

- none
- Cannot be used to determine whether the screen is on or off after sleep/wakeup

- Can be used by APP to call `setScreenOnOff` to set the screen on or off independently

Example code:

```
// Screen is on if true, off if  
false mAPI.isScreenOn();
```

5.6: Set screen switch

```
void setScreenOnOff(boolean on)
```

parameter:

- boolean on whether the screen is displayed
- For non-standard sleep/wakeup scenarios, the APP can control the screen switch separately
- Determine the screen switch status through `isScreenOn`

Example code:

```
// The screen is on if true, off  
if false mAPI.setScreenOnOff(false);
```

6: Watchdog

6.1: Turn on/off the watchdog

```
void enableWatchdog(boolean enable)
```

parameter:

- enable : true, means to enable the watchdog, false, means to disable the watchdog.

Example code:

```
// Open the  
watchdog  
  
mAPI.enabledWatchdog(true); // Close the watchdog mAPI.enableWatchdog(false);
```

6.2: Feed the dog

```
void feedWatchdog()
```

Example code:

```
// Feed the
watchdog // After turning on the watchdog, it will automatically reset if it is not fed within 15 seconds at most. It is recommended to call the feedWatchdog function once every 10
seconds to feed the watchdog mAPI.feedWatchdog();
```

7: Shutdown and restart

Example code:

```
//
ShutdownmAPI.shutdown
(); //
RestartmAPI.reboot ( ) ;
```

8: System sleep and wake-up

Example code:

```
//Use PowerManager to determine the system sleep and wake-up status
PowerManager pm = (PowerManager)
mSystemContext.getSystemService(Context.POWER_SERVICE); boolean
isInteractive = pm.isInteractive(); //Android 11 and subsequent Android version interface // Sleep

mAPI.sleep(); //
Wake
up mAPI.wakeup();
```

9: Navigation bar and drop-down status bar

9.1: Setting the Navigation Bar Visibility

```
void setNavigationBarVisibility(NavigationBarVisibility visibility)
```

parameter:

- visibility : Navigation bar visibility. There are three values, as follows
 - NavigationBarVisibility.INVISIBLE : The navigation bar is not visible, but you can slide it out from the bottom of the screen.
 - NavigationBarVisibility.VISIBLE : The navigation bar is visible.
 - NavigationBarVisibility.ALWAYS_INVISIBLE : The navigation bar is not visible and cannot be slid out from the bottom of the screen.

Example code:

```
mAPI.setNavigationBarVisibility(YNHAPI.NavigationBarVisibility.ALWAYS_INVISIBLE);
```

9.2: Get navigation bar visibility

```
NavigationBarVisibility getNavigationBarVisibility()
```

Return value:

- NavigationBarVisibility : Navigation bar visibility. There are three values, as follows
 - NavigationBarVisibility.INVISIBLE : The navigation bar is not visible, but you can slide it out from the bottom of the screen.
 - NavigationBarVisibility.VISIBLE : The navigation bar is visible.
 - NavigationBarVisibility.ALWAYS_INVISIBLE : The navigation bar is not visible and cannot be slid out from the bottom of the screen.

Example code:

```
// Get the navigation bar visibility.  
YNHAPI.NavigationBarVisibility visibility = mAPI.getNavigationBarVisibility(); // If the navigation bar is  
always invisible, then display the navigation  
bar. if (visibility == YNHAPI.NavigationBarVisibility.ALWAYS_INVISIBLE) {  
    mAPI.setNavigationBarVisibility(YNHAPI.NavigationBarVisibility.VISIBLE);  
}
```

9.3: Set the visibility of the drop-down status bar

```
void setExtendStatusBarVisibility(ExtendStatusBarVisibility visibility)
```

Example code:

```
// After the setting is completed, you need to restart the device. The  
setting will take effect after the next boot.mAPI.setExtendStatusBarVisibility (YNHAPI.ExtendStatusBarVisibility.VISIBLE); //You  
can pull down the top status bar
```

9.4: Get the visibility of the drop-down status bar

```
ExtendStatusBarVisibility getExtendStatusBarVisibility()
```

Return value:

- ExtendStatusBarVisibility: Status bar visibility. There are two values, as follows
- ExtendStatusBarVisibility.INVISIBLE: The status bar cannot be pulled down
- ExtendStatusBarVisibility.VISIBLE : The status bar can be pulled down

Example code:

```
// Get the visibility of the drop-down status bar.  
YNHAPI.ExtendStatusBarVisibility visibility =  
mAPI.getExtendStatusBarVisibility();
```

10: root privileges

10.1: Detecting root

`boolean isRoot()`

Return value:

- `boolean` : true, indicating that root privileges can be obtained through su. Otherwise, no.

10.2: Turn root on/off

`boolean enableRoot(boolean enable)`

parameter:

- `enable` : true, means to enable root. false, means to disable root.

Return value:

- `boolean` : true indicates success. false indicates failure.

Example code:

```
// If root permission is disabled, enable
root permission if (!mAPI.isRoot()) {
    if (mAPI.enableRoot(true)) { Log.d(TAG,
        "Successful root");
    }
}
```

11: Network related

11.1: Ethernet

11.1.1: Get IP information

`IpConfig getIpConfig()`

Return value:

- `IpConfig`: An entity class representing IP information.

Example code:

```
// Get IP
information IpConfig ipConfig = mAPI.getIpConfig(); //
IP address
String ip = ipConfig.ip; // Subnet
mask
String mask = ipConfig.mask; //
Gateway
String gateway = ipConfig.gateway; // dns list

ArrayList<String> dnsList = ipConfig.dnsList;
```

11.1.2: Using Ethernet Static IP

```
void setStaticIp(IpConfig config)
```

parameter:

- config: Indicates the information required to set a static IP.

Example code:

```
mAPI.setStaticIp(new IpConfig("192.168.1.2", "255.255.255.0", "192.168.8.1", Arrays.asList("192.168.8.1")));
```

11.1.3: Using Ethernet Dynamic IP

```
void setDhcpIp()
```

Example code:

```
// Set to use
dynamic IP mAPI.setDhcpIp();
```

11.1.4: Get Ethernet IP mode

```
IpMode getIpMode()
```

Return value:

- IpMode : Indicates the IP mode. For example, IpMode.STATIC indicates static IP mode, and IpMode.DHCP indicates dynamic IP mode.

Example code:

```
IpMode mode = mAPI.getIpMode(); if
(mode == IpMode.STATIC)
    { Log.d(TAG, "Static IP mode");
  } else if (mode == IpMode.DHCP) {
    Log.d(TAG, "Dynamic IP Mode");
  }
```

11.1.5: Is Ethernet enabled?

```
boolean isEthernetOpen()
```

Return value:

- boolean: true is on, false is off

Example code:

```
boolean isOpen = mAPI.isEthernetOpen();
```

11.1.6: Switch Ethernet

```
void setEthernetState(boolean open)
```

parameter:

- open: true to open Ethernet, false to close Ethernet

Example code:

```
mAPI.setEthernetState(true);
```

12: System time

12.1: Setting system time

```
void setSystemTime(long timeInMills)
```

parameter:

- timeInMills: Indicates the system time that needs to be set, in milliseconds.

Example code:

```
// Set the system to 2021-9-28 6:0:0 //
```

Note that the Calendar class starts with

month 0 and day 1 Calendar calendar = Calendar.getInstance(Locale.getDefault());

```
calendar.set(2021, 10, 28, 6, 0, 0);
```

```
mAPI.setSystemTime(calendar.getTimeInMillis());
```

12.2: Set timer power on/off

```
void setPowerOnOffAlarmCycle(int type, int[] timeOff, int[] timeOn)
```

parameter:

- type: You need to set the power on/off type.

- timeOff: The scheduled shutdown time that needs to be set.
- timeOn : The scheduled power-on time that needs to be set.

Example code:

```
// type = 1 sets the power on/off time, which only takes effect once . The customer needs to set it again through the APP each time the machine is
turned on. // type = 2 sets the power on/off time for each day. After the setting is completed, the system saves and takes effect. Resetting is required after restoring factory settings or
flashing the firmware. // type = 3 sets the power on/off time in a weekly cycle. After the setting is completed, the system saves and takes effect. Resetting is required after restoring
factory settings or flashing the firmware.

int type = 1; int[]
weekdays= null; //This value can be null or set randomly. When type is not 3, weekdays is invalid int[]
poweroff={2017,3,16,20,30}; //That is , shut down at 20:30 on 2017/3/16, the hour value is 0-23, and the minute value is 0-59
int[] poweron={2017,3,17,8,35}; //That is, turn on at 8:35 on 2017/3/17, the hour value is 0-23, and the minute
value is 0-59 mapi.setPowerOnOffAlarmCycle(type,weekdays,poweroff,poweron);

int type = 2; int[]
weekdays= null; //This value can be null or set randomly. When type is not 3, weekdays is invalid int[]
poweroff={20,30}; //That is, shut down at 20:30 every day, the hour value is 0-23, and the minute
value is 0-59 int[] poweron={8,35}; //That is, turn on at 8:35 every day, the hour value is 0-23,
and the minute value is 0-59 mAPI.setPowerOnOffAlarmCycle(type,weekdays,poweroff,poweron);

int type = 3; int[]
weekdays={1,3,4,5};//Set the scheduled power on and off time for Monday, Wednesday,
Thursday and Friday, required option int[] poweroff={20,30};//Set the power off time to 20:30,
the hour value is 0-23, the minute value is 0-59 int[] poweron={8,35};//Set the power on time
to 8:35, the hour value is 0-23, the minute value is 0-59 mAPI.setPowerOnOffAlarmCycle(type,weekdays,poweroff,poweron);
```

12.3: Cancel the scheduled power on/off

```
void cancelPowerOnOffAlarm()
```

Example code:

```
mAPI.cancelPowerOnOffAlarm();
```

12.4: Network Synchronization Time

12.4.1: Whether to use network synchronization time

```
void isEnableNetworkProvidedTime()
```

Example code:

```
boolean isEnable = mAPI.isEnableNetworkProvidedTime();
```

12.4.2: Setting network synchronization time

```
void setEnableNetworkProvidedTime(boolean enable)
```

Example code:

```
// Set whether to allow network synchronization time, true means allowed, false means not allowed
mAPI.setEnableNetworkProvidedTime(true);
```

13: Hardware Module

13.1: GPIO

The API in this section is applicable to all devices controlled by GPIO. For example, IO1, relay, LED light, USB power supply (GPIO port control

USB port power supply) etc.

Pay attention to the peripherals supported by the board, such as relays, LEDs, GPIO ports, etc. You need to check the specific board specifications. The parameters passed during use can be

Note the internal member variables in the YNHAPI.jar provided by our company. For detailed implementation, please refer to the

YNHCommonApiAPK source code

The operable peripheral IO ports are as follows (some boards do not support it, so the call may not work). For details, see YNHAPI.jar

```
const int GPIO_1           = 0;
const int GPIO_2           = 1;
const int GPIO_3           = 2;
const int GPIO_4           = 3;
const int GPIO_5           = 4;
const int GPIO_6           = 5;
const int GPIO_7           = 6;
const int GPIO_8           = 7;
const int GPIO_9           = 8;
const int GPIO_10          = 9;
const int GPIO_11          = 10;
const int GPIO_12          = 11;
const int GPIO_13          = 12;
const int GPIO_14          = 13;
const int GPIO_15          = 14;
const int LIGHT_RED        = 15; Red fill light
const int LIGHT_GREEN      = 16; Green fill light
const int LIGHT_BLUE       = 17; blue fill light
const int LIGHT_WHITE      = 18; White fill light
const int LIGHT_INFRARED   = 19; USB camera infrared fill light
const int RELAY            = 20; Relay
const int BELL             = 21; Doorbell
const int CASHBOX_0        = 22; the first cash box
const int CASHBOX_1        = 23; Second cash box
const int LIGHT_BOARD_BLUE = 24; onboard blue light
const int USB_1 = 25; the first USB power supply
const int USB_2            = 26; Second USB power supply
const int USB_3            = 27; The third USB power supply
const int LIGHT_BOARD_RED  = 28; onboard red light
const int LIGHT_CAMERA_EXTERNAL_INFRARED = 29; MIPI camera infrared fill light
```

```
const int LIGHT_BOARD_YELLOW = 30;
```

13.1.1: Get GPIO status

```
GpioState getGpioState(int index)
```

parameter:

- index: indicates the GPIO whose status needs to be obtained. For example, GPIO_1 indicates IO1, RELAY indicates relay, LIGHT_RED indicates red light, etc. For details, please refer to YNHAPI.jar provided by our company.

Example code:

```
if (mAPI.getGpioState(YNHAPI.GPIO_1) == YNHAPI.GpioState.HIGH) {  
    Log.d(TAG, "GPIO1 is high level");  
}
```

13.1.2: Setting GPIO Status

```
boolean setGpioState(int index, GpioState state)
```

parameter:

- index: indicates the GPIO whose status needs to be set. For example, GPIO_1 indicates IO1, RELAY indicates relay, LIGHT_RED indicates red light, etc. For details, please refer to YNHAPI.jar provided by our company.
- state : The state to be set. For example, GpioState.HIGH means setting it to high level.

Return value:

- boolean : true indicates success, false indicates failure.

Example code:

```
if (mAPI.setGpioState(YNHAPI.GPIO_1, YNHAPI.GpioState.HIGH)) { Log.d(TAG,  
    "Successfully set high level");  
}
```

13.1.3: Get GPIO mode

```
GpioMode getGpioMode(int index)
```

parameter:

- index: indicates the GPIO whose mode needs to be obtained. For example, GPIO_1 indicates IO1, RELAY indicates relay, and LIGHT_RED indicates red light. For details, please refer to YNHAPI.jar provided by our company.

Return value:

- GpioMode: Indicates the GPIO mode obtained. For example, GpioMode.INPUT indicates input mode.

Example code:

```
YNHAPI.GpioMode gpioMode = mAPI.getGpioMode(YNHAPI.GPIO_1); if
(gpioMode == GpioMode.INPUT) { Log.d(TAG,
    "GPIO1 is input mode");
}
```

13.1.4: Setting GPIO Mode

```
boolean setGpioMode(int index, GpioMode mode)
```

parameter:

- index: indicates the GPIO index for which the mode needs to be set. For example, GPIO_1 indicates IO1, RELAY indicates relay, and LIGHT_RED indicates red light. For details, please refer to YNHAPI.jar provided by our company.
- mode: Indicates the mode to be set. For example, GpioMode.INPUT sets it to input mode.

Return value:

- boolean : true indicates success, false indicates failure.

Example code:

```
if (mAPI.setGpioMode(YNHAPI.GPIO_1, YNHAPI.GpioMode.INPUT)) {
    Log.d(TAG, "Successfully set input mode");
}
```

13.1.4: Monitoring GPIO Status Changes

```
synchronized void listenGpio(int index, GpioListenerCallback callback)
```

After this function is called, the GPIO status will be checked every 1 second. If there is a change, the application will be notified. Therefore, this function is more suitable for scenarios where the GPIO changes can be maintained for a period of time. It is not suitable for detecting short button presses or scenarios where the change time is less than 1 second. If there is such a scenario, getGpioState can be used directly, and the customer can control the cycle by himself.

parameter:

- index: indicates the GPIO index that needs to monitor the state change. For example, GPIO_1 indicates IO1, RELAY indicates relay, and LIGHT_RED indicates red light. For details, please refer to YNHAPI.jar provided by our company.
- callback : The callback of the listener.

Example code:

```
YNHAPI.GpioListenerCallback gpioListenerCallback = new
YNHAPI.GpioListenerCallback() {
    public void onChanged(int index, int oldValue, int newValue) {
        // When the state changes, call back this
        function // Parameter index is the GPIO index value, indicating which GPIO level
        state has changed // Parameter oldValue indicates
        the old level state // Parameter newValue indicates the new level state
    }
};
mAPI.listenGpio(YNHAPI.GPIO_1, gpioListenerCallback);
```

13.1.5: Cancel monitoring GPIO status changes

```
synchronized void unlistenGpio(int index, GpioListenerCallback callback)
```

parameter:

- index: indicates the GPIO index for which the monitoring status change needs to be canceled. For example, GPIO_1 indicates IO1, RELAY indicates relay, and LIGHT_RED indicates red light. For details, please refer to YNHAPI.jar provided by our company.
- callback : The callback of the listener that has been set.

Example code:

```
// Monitor GPIO1 status changes
mAPI.listenGpio(YNHAPI.GPIO_1, gpioListenerCallback);

// When you no longer need to monitor GPIO1 level changes, call
unlistenGpio() mAPI.unlistenGpio(YNHAPI.GPIO_1, gpioListenerCallback);
```

13.2: Setting LED brightness

```
void setLightBrightness(int index, int brightness)
```

parameter:

- index : GPIO index corresponding to the LED. For example, LIGHT_RED, LIGHT_GREEN, LIGHT_WHITE. For details, please refer to YNHAPI.jar provided by our company.
- brightness : brightness value, ranging from [0, 255].

Example code:

```
// Set the brightness of the
red light to 80% mAPI.setLightBrightness(YNHAPI.LIGHT_RED, 204);
```

13.3: Wiegand

13.3.1: Get Wiegand mode

```
WiegandMode readWiegandMode()
```

Return value:

- WiegandMode : Get the Wiegand mode. For example, WiegandMode.INPUT indicates input mode.

Example code:

```
YNHAPI.WiegandMode wiegandMode = mAPI.readWiegandMode(); if
(wiegandMode == YNHAPI.WiegandMode.INPUT) { Log.d(TAG,
    "Wiegand mode is input mode");
}
```

13.3.2: Setting Wiegand Mode

```
void writeWiegandMode(WiegandMode mode)
```

parameter:

- mode : The Wiegand mode to be set. For example, WiegandMode.INPUT means setting it to input mode.

Example code:

```
mAPI.writeWiegandMode(YNHAPI.WiegandMode.INPUT);
```

13.3.3: Read Wiegand

Wiegand's read is divided into synchronous blocking read and asynchronous callback read.

13.3.4: Synchronous Blocking Read

```
long readWiegand()
```

Return value:

- long : The Wiegand number read.

Example code:

```
CompletableFuture.supplyAsync() -> mAPI.readWiegand() // Asynchronous thread  
    reading.whenComplete ((code, throwable) ->  
        { // Main thread  
            update runOnUiThread() -> mTextView.setText(String.valueOf(code));  
        });
```

13.3.5: Asynchronous Read

```
void readWiegandAsync(WiegandCallback callback)
```

parameter:

- callback : callback for asynchronously reading Wiegand.

Example code:

Note that the call should be made as follows: If you need to read Wiegand card numbers asynchronously multiple times, you must call readWiegandAsync in the onSuccess or onFailure callback function.

```
private YNHAPI.WiegandCallback mCallback = new YNHAPI.WiegandCallback() {  
    @Override  
    public void onSuccess(long code) {  
        // Successfully read the Wiegand number, the parameter code indicates the Wiegand number.  
    }  
}
```

```

        //If you need to read Wiegand in a loop, you need to call readWiegandAsync
        repeatedly mAPI.readWiegandAsync(mCallback);
    }

    @Override
    public void onFailure() { // Read
        failed

        //If you need to read Wiegand in a loop, you need to call readWiegandAsync
        repeatedly mAPI.readWiegandAsync(mCallback);
    }
};
mAPI.readWiegandAsyn(mCallback);

```

Note that the callback happens on the worker thread.

13.3.6: Write Wiegand

```
void writeWiegand(WiegandFormat format, long code)
```

parameter:

- format : Wiegand format. For example, WiegandFormat.FORMAT_26 means 26-bit Wiegand format.
- code : Wiegand code.

Example code:

```
mAPI.writeWiegand(YNHAPI.WiegandFormat.FORMAT_26, 12345678);
```

14: Function reference code

14.1: Cash drawer function

The cash box function is to operate GPIO, first pull the GPIO state low, and then pull it high after 500ms.

Example code:

```

CompletableFuture.runAsync(() ->
    { mYNHAPI.setGpioState(YNHAPI.CASHBOX_0, YNHAPI.GpioState.LOW); try

    { TimeUnit.MILLISECONDS.sleep(500);
    } catch (InterruptedException e)
    { e.printStackTrace();

    } }).thenAccept(u -> {
    mYNHAPI.setGpioState(YNHAPI.CASHBOX_0, YNHAPI.GpioState.HIGH);
});

```

14.2: Execute root privilege commands

Example code:

```
boolean executeCmd(String content) throws IOException, InterruptedException { boolean ret = false;
    Process p =
    Runtime.getRuntime().exec("su"); if (p != null) { PrintWriter
    pw = new
        PrintWriter(p.getOutputStream(), true); pw.println(content); pw.println("exit"); ret =
        p.waitFor() == 0;

    }
    return ret;
}
```