

# **MIPS Smart Pass LAN Interface Documentation**

## **V1.5**

## **Copyright statement**

All information contained in this user manual, including but not limited to it, is protected by copyright law. Without the permission of Shenzhen Smart Device Technology Co., Ltd. (hereinafter referred to as "SMDT"), there shall be no counterfeiting, copying, excerpting, translating, publishing, etc.

## **About the file**

This document is used as a guide. The photos, graphics, charts and illustrations provided in the document are for explanation and explanation purposes only, and may differ from the specific product, please refer to the actual product.

Version	Description	Date
V1.0	Create	2019-07-26
V1.1	Add temperature interface	2020-02-28
V1.2	Added the interface for obtaining setting parameters, setting card number, obtaining device parameters, setting time and setting callback.	2020-03-22
V1.3	The temperature parameter setting interface adds stranger record switch and fan switch to obtain temperature parameters, open the door and restart the interface.	2020-03-28
V1.4	Add personnel query interface	2020-04-10

## Content

Interface Operation Manual.....	5
Interface Operation Instructions .....	5
Interface list .....	6
1.Set device password.....	6
2.Parameter configuration .....	7
3.Get parameter configuration .....	9
4.Personnel registration .....	10
5.Personnel information editing .....	12
6.Delete person .....	12
7. Identification record acquisition .....	13
8. Modify LOGO.....	14
9.Get device mac address .....	15
10.Request method: POST request .....	15

11. Body temperature and mask parameter configuration .....	15
12. Get pass photos.....	17
13. Device parameter information acquisition .....	17
14. Device photo .....	18
15. Device initialization .....	18
16. System time setting.....	19
17. Identify the recording interface callback settings .....	19
18. Device open door .....	20
19. Device restart .....	20
20. Get body temperature and mask parameters.....	21
21. Personnel inquiry .....	21
22. Personnel page query.....	22

# Interface Operation Manual

## Interface Operation Instructions

### Interface specification

- Interface root address: http: // device ip address: 8080 /
- Interface form: provide external services through HTTP requests.
- Interface security: The device password (pass) needs to be set first when calling the interface for the first time, and the device password (pass) needs to be passed in as the interface security verification key for any subsequent calls to the interface.

### Interface returns

Interface general return instructions:

```
public class ResultInfo<T> {  
    private Int result;//Indicates whether the interface is tuned up. 1 is success, 0 is failure. As long as  
    the device server can respond, the value is always 1.  
    private Boolean success;// Refers to the success of this operation. Success is true and failure is  
    false.  
    private T data;// Represents the business data returned by the interface. The type can be numeric,  
    string, or collection.  
}
```

The interface returns mentioned in the documentation are examples. The returned data of individual interfaces will be adjusted slightly, and the actual returned results shall prevail.

# Interface list

## 1. Set device password

Request method: POST request

Request address: http:// device IP: 8080 / setPassWord

Request data:

Field	Types	Required	Description
oldPass	String	Y	The old password
newPass	String	Y	New password, the password cannot be empty or blank

Request description:

- New device or device after reset (initialization resumed). Before calling other interfaces, you need to set the initial password first, oldPass and newPass can be passed the same value.
- When changing the password, enter the new and old passwords separately.
- This interface call does not need to enter pass.
- The password cannot be empty or blank.

The return example

```
{
  "data": "passWord:q", // The device password, also called the interface calling password,
  please keep it properly. If you forget the password, you need to reset the device, and the device
  will clear all data.
  "result": 1, //Interface tuning
  "success": true //The device password is set successfully.
}
```

Example:

The screenshot shows a REST client interface with the following details:

- Method: POST
- URL: http://192.9.51.214:8080/setPassWord
- Body type: form-data
- Form fields:
  - oldPass: 123456
  - newPass: q
- Response body (JSON):

```
{
  "data": "passWord:q",
  "result": 1,
  "success": true
}
```

## 2. Parameter configuration

Request method: POST request

Request address: http://device IP: 8080 / setConfig

Request data:

Field	Types	Required	Description
pass	String	Y	Device password
config	Json	Y	Parameter configuration Json parameters for json.

Request description

- POST request parameters are placed in the body.
- The config request parameter is in json format, the example is as follows (please adjust according to the actual situation).

```
{
  "tricolorLamp": 2,
  "idCardNo": 2,
  "icNo": 2,
  "companyName": "smdt",
  "deviceId": 86,
  "diplayCustom": "{name}",
  "displayMode": 1,
  "id": 1,
  "liveIdentiLevel": 1,
  "passType": false,
  "password": "123456",
  "recoDistance": "1.5",
  "recoInterval": "2000",
  "relayDelay": 5,
  "relayMode": 0,
  "serialCustomize": "#{idcardNum}#",
  "serialMode": 2,
  "similarity": 80,
  "strangerVoiceCustom": "moshengren",
  "strangerVoiceMode": 2,
  "voiceCustom": "name",
  "voiceMode": 1,
  "wg": 0
}
```

Config Field description

Field	Types	Description
companyName	String	Company name

deviceId	int	Device ID
displayCustom	String	Display customization, the default is {name}.
displayMode	int	Display mode, 1: display name, 100: custom
id	int	
liveIdentiLevel	int	Live portrait recognition level, the default is fast mode.0 No live portraits; 1 Quick mode cannot refuse to recognize portrait photos; 2 Can refuse to recognize some portrait photos, 3 Can refuse to recognize portrait photos and videos.
passType	boolean	Access type: false is entry, true is exit.
password	String	Access password
recoDistance	String	Recognition distance (0.5m ~ 3m)
recoInterval	String	Identification interval (2000 ~ 10000)
relayDelay	Int	Relay automatically closes after a delay of x seconds (5 ~ 63)
relayMode	Int	Relay mode, 0: indicates the automatic closing mode, that is, it will automatically close after a delay of X seconds after opening the relay; 1: indicates that the mode is not automatically closed, that is, it will not automatically close after opening the relay
serialCustomize	String	Serial port definition, the default is "# {idcardNum} #", unlimited length.
serialMode	int	Serial port mode: 1 open the door; 2 do not output; 3 output personnel ID; 4 output ID / IC card number; 100 customized output.
similarity	Int	Similarity, the default is 80,30 ~ 100 adjustment.
strangerVoiceCustom	String	Stranger voice customization, default for stranger recognition, within 32 characters.
strangerVoiceMode	Int	Stranger voice mode-the default stranger alarm; 1 does not require voice announcement; 2 stranger alarm; 100 customization
voiceCustom	String	Voice customization, default is "name", within 32 characters
voiceMode	Int	Voice mode, 1 does not need to broadcast, 2 broadcast name, 100 customization
wg	Int	Wiegand 26 or 34 output mode, 0: output card number WG26, 1: output personnel ID WG26, 2: output card number WG34, 3: output personnel ID WG34
icNO	int	IC card, 1: swipe card; 2: human & card comparison
idCardNo	int	ID card, 1 Closed; 2 Person & ID comparison; 3 Person & ID comparison (visitor version)
tricolorLamp	int	Red light; 1. off, 2. on

Return data

```
{
  "data": { //Set parameters
    "tricolorLamp": 2,
    "idCardNo": 2,
    "icNo": 2,
    "companyName": "smdt",
    "deviceId": 86,
    "displayCustom": "{name}",
    "displayMode": 1,
    "id": 1,
    "liveIdentiLevel": 1,
    "passType": false,
    "password": "123456",
    "recoDistance": "1.5",
    "recoInterval": "2000",
    "relayDelay": 5,
    "relayMode": 0,
    "serialCustomize": "#{idcardNum}#",
    "serialMode": 2,
    "similarity": 80,
    "strangerVoiceCustom": "moshengren",
    "strangerVoiceMode": 2,
    "voiceCustom": "name",
    "voiceMode": 1,
    "wg": 0
  },
  "result": 1, //Interface tuning
  "success": true //Set up successfully
}
```

### 3. Get parameter configuration

Request method: POST request

Request address: http://device IP: 8080 / getConfig

Request data:

Field	Type	Required	Description
pass	String	Y	Device password

Return example

```
{
```

```

"data": { //the setting parameters
  "tricolorLamp": 2,
  "idCardNo": 2,
  "icNo": 2,
  "companyName": "smdt",
  "deviceId": 86,
  "displayCustom": "{name}",
  "displayMode": 1,
  "id": 1,
  "liveIdentiLevel": 1,
  "passType": false,
  "password": "123456",
  "recoDistance": "1.5",
  "recoInterval": "2000",
  "relayDelay": 5,
  "relayMode": 0,
  "serialCustomize": "#{idcardNum}#",
  "serialMode": 2,
  "similarity": 80,
  "strangerVoiceCustom": "moshengren",
  "strangerVoiceMode": 2,
  "voiceCustom": "name",
  "voiceMode": 1,
  "wg": 0
},
"result": 1, //Interface tuning
"success": true //Set up successfully
}

```

## 4. Personnel registration

Request method: POST request

Request address: [http:// device IP:8080/person/create](http://device IP:8080/person/create)

Request data:

Field	Types	Required	Description
pass	String	Y	Device password
person	Json	Y	Personnel information Json parameter in json

Request description:

- POST request parameters are placed in the body.
- The base64 string of the photo, without the header, such as: data: image / jpg; base64.

- Staff photos only support JPG and PNG.

#### Person field description

Field	Types	Description
age	int	Age
imgBase64	String	Photo base64 string
name	String	Name
prescription	String	Pass period (start time and end time separated by comma), yyyy-MM-dd HH: mm, yyyy-MM-dd HH: mm
sex	int	Sex
type	Int	The default type is 1. 1 is a visitor, 2 is a blacklist, and 3 is an employee.
vipID	Int	Create the id of the person, and find the person to edit based on the id.
welCome	String	Reserved, you can leave it blank.
icCard	String	IC card information
card	String	ID numbers
wn	String	Reserved, you can leave it blank.

#### Return example

```
{
  "data": "msg: Successful registration ",
  "result": 1,
  "success": true
}
```

If the registration fails, a failure description is returned. The corresponding error codes are as follows:

- <0: add failed
- 1: Failed to get image bitmap
- 2: SDK feature extractor abnormal
- 3: Set the face id parameter  
mipsFaceVipDB .idxInFaceDB) Less than 0 or id repeated
- 4: Image conversion failed
- 5: Image resolution exceeds requirements (4096\*2688)
- 6: SDK face detector abnormal
- 7: No face detected
- 8: Face detection failed
- 9: Failed to add face
- 10: The number of faces exceeds
- 11: Face size is smaller than the preset pixel threshold
- 12: Face image quality is poor
- 13: Face elevation angle exceeds threshold
- 14: Face tilt angle exceeds threshold

-15: Face elevation angle exceeds threshold

## 5. Personnel information editing

Request method: POST request

Request address: http://device ip: 8080 / person / update

Request data:

Field	Types	Required	Description
pass	String	Y	Device password
person	Json	Y	Personnel information Json parameter in json

Request description:

- POST request parameters are placed in the body
- The base64 string of the photo, without the header, such as: data: image / jpg; base64.
- Personal photos only support JPG and PNG

Person Parameter Description

Field	Types	Description
age	int	age
imgBase64	String	Photo base64 string
name	String	name
prescription	String	Pass period (start time and end time separated by comma), yyyy-MM-dd HH: mm, yyyy-MM-dd HH: mm
sex	int	gender
type	Int	Type: The default is 1, 1 is a visitor, 2 is a blacklist, and 3 is employee
vipID	Int	Create the id of the person, and find the person to edit based on the id.
welcome	String	Reserved, you can leave it blank.
icCard	String	ic card information
card	String	ID numbers
wn	String	Reserved, you can leave it blank.

Example of returned data

```
{
  "data": "msg: Personnel editing success ",
  "result": 1,
  "success": true
}
```

## 6. Delete person

Request method: POST request

Request address: http://device ip: 8080/person/delete

Request data:

Field	Types	Required	Description
pass	String	Y	Device password
deleteld	String	Y	Deleted person ID

Return example

```
{
  "data": "msg: Person deleted successfully ",
  "result": 1,
  "success": true
}
```

## 7. Identification record acquisition

Request method: POST request

Request address: http://device ip: 8080/newFindRecords

Request data:

Field	Types	Required	Description
pass	String	Y	Device password
startTime	String	Y	Start time: 2019-07-11 12:00
endTime	String	Y	End time: 2019-07-11 17:00

Return example

```
{
  "data": [
    {
      "birthDate": "", //Reserve
      "currentTime": 1562819094334, // Access time, milliseconds
      "idCardNum": "", // Personnel ID
      "imageFlag": 0, // Reserve
      "imageName": "1_0_1562819094334.jpg", // The name of the photo taken by the public. If you
      want to get the photo, please call the getRecordImg interface.
      "name": "pdf", // Passer name
      "type": 1, // Type
      "tempratrue": "36.5", // "mask": 1 // Whether to wear a mask, 1: worn, 0: not worn, -1 when
      mask detection is closed
    },
    {
      "birthDate": "",
```

```

        "currentTime": 1562824109687,
        "idCardNum": "",
        "imageFlag": 0,
        "imageName": "1_86_1562824109687.jpg",
        "name": "pdf",
        "type": 1
        "temprature": "36.5",
        "mask": 1 // Whether to wear a mask, 1: worn, 0: not worn
    },
    {
        "birthDate": "",
        "currentTime": 1562825856546,
        "idCardNum": "null",
        "imageFlag": 1,
        "imageName": "100_86_1562825856546.jpg",
        "name": "tts",
        "type": 1,
        "temprature": "36.5",
        "mask": 1 // Whether to wear a mask, 1: worn, 0: not worn
    }
},
"result": 1, // Interface tuning
"success": true // Success
}

```

## 8. Modify LOGO

Request method: POST request

Request address: http://device ip: 8080 / changeLogo

Request data:

Field	Type	Required	Description
pass	String	Y	Device password
imgBase64	String	Y	Logo icon Base64, only supports png, jpg

Request description

- POST request parameters are placed in the body.
- The base64 string of the photo, without the header, such as: data: image / jpg; base64.
- Staff photos only support JPG and PNG.

Return example

```
{
```

```
"data": "msg: logo Successfully modified ",
"result": 1,
"success": true
}
```

## 9. Get device mac address

## 10. Request method: POST request

Request address: <http://device ip: 8080/getDeviceMac>

Request data:

Field	Type	Required	Description
pass	String	Y	Device password

Return example

```
{
  "data": "8CFCA0000023",
  "result": 1,
  "success": true
}
```

## 11. Body temperature and mask parameter configuration

Request method: POST request

Request address: <http://device ip: 8080/tempAndMaskSetting>

Request data:

Field	Type	Required	Description
pass	String	Y	Device password
config	Json	Y	Parameter configuration Json parameters for json

Request description

- POST request parameters are placed in the body
- The config request parameter is in json format, the example is as follows (please adjust according to the actual situation)

```
{
  "isBodyTempAlarm": 1,
  "isBodyTempStart": 1,
  "isHighFeverAdopt": 0,
}
```

```

    "isLowFeverAdopt": 0,
    "isLowTempAdopt": 0,
    "isStandardTempAdopt": 1,
    "isWearingMask": 0,
    "standardBodyTemp": "37.3",
    "isStrangerRecord":0,
    "tempCompensation":0.3,
    "isFan":0
}

```

#### Config Field description

Field	Type	Description
isBodyTempAlarm	int	1 Turn on body temperature alarm 0 Turn off body temperature alarm
isBodyTempStart	Int	1 Turn on body temperature detection 0 Turn off body temperature detection
standardBodyTemp	String	Body temperature threshold 37.3 (accurate to one decimal place). After turning on the body temperature detection and body temperature alarm, it detects that the body temperature exceeds the threshold and plays a sound alarm.
isHighFeverAdopt	int	Whether high fever passed (0: not passed 1: passed); used to open the door (range 38.5-43.0)
isLowFeverAdopt	int	Whether low fever passed (0: not passed 1: passed); used to open the door (range 37.3-38.5)
isLowTempAdopt	int	Whether low fever passed (0: not passed 1: passed); used to open the door (range 37.3-38.5)
isStandardTempAdopt	int	Whether the normal body temperature passes (0: not pass 1: pass); used to open the door (range 36.1-37.2)
isWearingMask	int	Wear mask detection switch (0: No 1: Yes)
isStrangerRecord	int	Stranger identification record storage switch (0: No, 1: Yes)
isFan	int	Fan switch (0: No, 1: Yes)
tempCompensation	float	Temperature compensation, range (-1 ~ 1), positive number means upward compensation, negative number downward compensation

#### Return example

```

{
  "data": {
    "deviceId": 88,
    "id": 10,
    "isBodyTempAlarm": 1,
    "isBodyTempStart": 1,
    "isHighFeverAdopt": 0,
    "isLowFeverAdopt": 0,
    "isLowTempAdopt": 0,

```

```

    "isStandardTempAdopt": 1,
    "isWearingMask": 0,
    "standardBodyTemp": "37.3",
    "isStrangerRecord":0,
    "tempCompensation":0.3,
    "isFan":0
  },
  "result": 1,
  "success": true
}

```

## 12. Get pass photos

Request method: POST request

Request address: http://device ip: 8080/getRecordImg

Request data:

Field	Type	Required	Description
pass	String	Y	Device password
imgName	String	Y	Photo name (obtained by the identification record interface)

Return example

```

{
  "data":"/9j/4AAQSkZJRgABAQAAQABAAQ/2wB", // Image base64 string
  "result": 1, //1: The interface is tuned up; 0: The interface is abnormal
  "success": true // true: successful acquisition; false: acquisition failure;
}

```

## 13. Device parameter information acquisition

Request method: POST request

Request address: http://device ip: 8080/getDeviceInfo

Request data:

Field	Type	Required	Description
pass	String	Y	Device password

Return example

```

{
  "data":{"freeSpace":"3.61GB", // freeSpace:Free space
    "ip":"192.9.51.45", //ip:ip address
  }
}

```

```

    "mac": "8CFCA0036225", // mac address
    "time": 1584518232928, // System time (ms) long type
    "version": "1.5.0.22.0.01" // App current version number
  },
  "result": 1,
  "success": true
}

```

## 14. Device photo

Request method: POST request

Request address: <http://equipment ip: 8080/photograph>

Request data:

Field	Type	Required	Description
pass	String	Y	Device password

Return example

```

{
  "data": "/9j/4AAQSkZJRgABAQAA ...", // Base64 string, need to be converted to suffix png
  pictures
  "result": 1,
  "success": true
}

```

## 15. Device initialization

Request method: POST request

Request address: <http://device ip: 8080/initialization>

Request data:

Field	Type	Required	Description
pass	String	Y	Device password

Request description:

- Delete all identification data, personnel data, characteristics and other data on the device, and clear all databases
- Delete the attributes set through the device configuration interface

Return example

```
{
  "data": "Initialization successful"
  "result": 1,
  "success": true
}
```

## 16. System time setting

Request method: POST request

Request address: <http://device ip: 8080/setDeviceTime>

Request data:

Field	Type	Required	Description
pass	String	Y	Device password
year	String	Y	year
month	String	Y	month
day	String	Y	day
hours	String	Y	hours (24-hour clock)
minute	String	Y	minute

Request description:

- After the configuration is successful, the device time is changed to the currently set time.
- If the device is connected to the public network, the device itself has a network time calibration mechanism, and the system adjusts the device time to the public network time.
- For the device to display the time set manually, the device must be in the local area network. If connected to the public network, the device uses the public network time by default when refreshing its time.

Return example

```
{
  "data": "2020-03-15 22:10:50", // Device current time
  "result": 1,
  "success": true
}
```

## 17. Identify the recording interface callback settings

Request method: POST request

Request address: <http://device ip: 8080/setIdentifyCallback>

Request data:

Field	Type	Required	Description
-------	------	----------	-------------

pass	String	Y	Device password
callbackUrl	String	Y	Identify the url interface for record upload

Request description:

The callback can be set in other settings of the APP terminal settings

Recognition record return field, please refer to MIPS Face Recognition Callback Interface Document V1.0

Return example

```
{
  "data": "http://192.9.51.45:8080/setIdentifyCallback",
  "result": 1, //1: set successfully; -1: no callbackUrl field; -2: callbackUrl field is empty
  "success": true //true: set successfully, false: set failed
}
```

## 18. Device open door

Request method: POST request

Request address: http://device ip: 8080/device/openDoorControl

Request data:

Field	Type	Required	Description
pass	String	Y	Device password

Return example

```
{
  "data": " Open the door successfully ",
  "result": 1,
  "success": true
}
```

## 19. Device restart

Request method: POST request

Request address: http://device ip: 8080/restartDevice

Request data:

Field	Type	Required	Description
pass	String	Y	Device password

Return example

```
{
  "data": " Successful restart ",
  "result": 1,
  "success": true
}
```

## 20. Get body temperature and mask parameters

Request method: POST request

Request address: <http://device ip: 8080/getTempAndMaskSetting>

Request data

Field	Type	Required	Description
pass	String	Y	Device password
config	Json	Y	Parameter configuration Json parameters for json

Return data (for example, for details, please refer to temperature and mask parameter configuration instructions)

```
{
  "data": {
    "isBodyTempAlarm ":1,"
    isBodyTempStart ":1,"
    isHighFeverAdopt ":0,"
    isLowFeverAdopt ":0,"
    isLowTempAdopt ":0,"
    isStandardTempAdopt ":1,"
    isStrangerRecord ":0,"
    isWearingMask ":0,"
    standardBodyTemp ":37.3 ","
    tempCompensation ":0.3,"
    tempCompensationParam ":1
  },
  "result": 1,
  "success": true
}
```

## 21. Personnel inquiry

Request method: POST request

Request address: <http://device ip: 8080/person/find>

Request data:

Field	Type	Required	Description
pass	String	Y	Device password
vipId	String	Y	Personnel ID specifies to query multiple persons separated by commas, such as "100,101,102", the maximum number is 50;

Return data:

In addition to the faceID field, please refer to Interface 4 Personnel Registration for other detailed field descriptions

```
{
  "data": "[
    {
      "prescription": "2020-03-21 00:00,2030-12-13 00:00",
      "age": 0,
      "card": "",
      "faceID": 57358,
      "icCard": "",
      "name": "XieWen",
      "sex": 1,
      "type": 3,
      "vipID": 57358 // Less than 0 means that the picture verification failed. Greater
than or equal to 0 means success. For the reason of failure, please refer to the registration result
of interface 4 personnel. Return to description.
    }
  ]",
  "result": 1,
  "success": true
}
```

## 22. Personnel page query

Request method: POST request

Request address: <http://device ip: 8080/person/findByPage>

Request data:

Field	Type	Required	Description
pass	String	Y	Device password
index	int	Y	Page number, starting from 0

length	int	Y	The maximum number of each page is less than or equal to 50. If it is greater than 50, only 50 will be searched.
--------	-----	---	--

**Return data:**

In addition to the faceID field, please refer to Interface 4 for detailed field descriptions. Personnel registration

```

{
  "data": "{
    "pageInfo": {
      "index": 0,
      "length": 1,
      "size": 1,
      "total": 2
    },
    "records": [
      {
        "age": 0,
        "card": "",
        "faceID": 66614,
        "icCard": "",
        "name": "fff",
        "sex": 0,
        "type": 3,
        "prescription": "2020-03-21 00:00,2030-12-13 00:00",
        "vipID": 66614 // Less than 0 means that the picture verification fails, and
greater than or equal to 0 means success. For the reason of failure, please refer to the registration
result of interface 4 personnel. Back to description.
      }
    ]
  }",
  "result": 1,
  "success": true
}

```