# Local Area Network Interface Document for MIPS Smart Pass Instrument

# Version 1.2

# copyright notice

The copyright of this manual belongs to Shenzhen Shimeitai Communication Technology Co., Ltd, and all rights are reserved.Without our consent (in writing),No unit or individual may extract part or all of this manual without permission.

About this document

This document is for guidance. Photos, graphics, diagrams and illustrations provided in the document are for explanation only,There may be differences with specific products, please refer to the actual product.

# Release Notes

|  |  |  |  |
| --- | --- | --- | --- |
| **Version** | **description** | **description** | **date** |
| V1.0 | create |  | 2019-07-26 |
| V1.1 | Increase Temperature Interface |  | 2020-02-28 |
| V1.2 | Added Get Setting Parameters, Set Card Number, Get Device Parameters, SetTime,set callback interface and other functions | 2020-03-22 |

content

Local Area Network Interface Document for MIPS Smart Pass Instrument 1

Version 1.2 1

copyright notice 2

Release Notes 3

Interface Operations Guide 5

Interface Description 5

Interface list 6

1．Set device password 6

2.Parameter Configuration 7

3.Get parameter configuration 9

4.Personnel registration 9

5.Edit personnel information 10

6.Delete personnel 11

7.Get identification record 12

8 .Modify LOGO 13

9 .Get device MAC address 13

10. Temperature and Mask Parameter Configuration 14

11. Get Passage Photos 15

12. Get device parameter information 16

13. Device photo 16

14. Device initialization 16

15. Set the system time 17

16. Set the identification record interface callback 18

# Interface Operations Guide

## Interface Description

### interface specification

* Interface Address:[http://IP of device](http://设备/)  address:8080/
* Interface form:Provide services externally through HTTP requests
* Interface security:The first time you call the interface, you need to set the device password,Subsequent calls to any interface require the device password to be passed in as the interface security verification key.

### Interface Return

Interface Common Return Description:

public class ResultInfo<T> {

private Int result;//Indicates whether the interface is tuned, ‘1'means success,'0' means fail,Generally, as long as the device server can respond, the value is 1.

private Boolean success;//Whether the operation was successful,success shows ‘true',fail shows ‘false'.

 private T data;//The business data returned by the interface. the type can be numeric value, string, collection, etc

}

For the interface return examples involved in the document, the return data of individual interfaces will be slightly adjusted, and the actual return results must prevail.

#

# Interface list

## 1．Set device password

request mode:Requests use POST

Request address: ip of device :8080/set Pass Word

Request data:

|  |  |  |  |
| --- | --- | --- | --- |
| **field** | **Data type** | **Must fill** | **description** |
| OldPass | String | Y | Old Pass |
| NewPass | String | Y | New Pass,Password may not be blank or have spaces |

Request description:

* New device or device after reset (recovery initialization), you need to set the initial password before calling other interfaces.Just fill in the old password field and the new password field with the same value.
* When changing the password, pass in the new and old password respectively
* This interface call does not need to pass in a password.
* Password may not be blank or have spaces.

Return example

{

"data": "passWord:q", //Device password, also known as interface calling password, please keep it properly. If you forget your password, you need to reset the device, and the device will empty all data

"result": 1, //Interface accessible

"success": true //Device password set successfully

}

‘Postman' example



## 2.Parameter Configuration

Request mode:Requests use POST

Request address: [http://](http://设备/) IP of device :8080/set Config

Request data:

|  |  |  |  |
| --- | --- | --- | --- |
| **Field** | **Type** | **Must fill** | **description** |
| pass | String | Y | Device password |
| config | Json | Y | Parameter configuration JSON, the following are the parameters of JSON |

Request description:

* POST- request parameters are placed in the body
* config- the request parameters are in JSON format, with the following examples (please adjust them to suit your needs)

#### {

"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": 26

}

config field description

|  |  |  |
| --- | --- | --- |
| **field** | **Type** | **description** |
| CompanyName | String | Company Name |
| deviceId | int | ID of the associated device |
| diplayCustom | String | Display custom, default is {name} |
| displayMode | int | display mode,1-display name,100-display custom |
| id | int |  |

|  |  |  |
| --- | --- | --- |
| liveIdentiLevel | int | Life recognition level, default quick mode, '0' means no living body, ‘1' means fast mode cannot Reject photos, ‘2' means you can reject some photos, ‘3' means you can reject photos and fit |
| passType | boolean | Access type: ‘false' means in, ‘true' means out |
| password | String | Passcode |
| recoDistance | String | Recognition distance(0.5m~3m) |
| recoInterval | String | Recognition interval(2000 ~10000) |
| relayDelay | Int | Relay delay x seconds will automatically close(5~63) |
| relayModerelayMode | IntInt | Relay mode: ‘0' means auto-close mode, delaying X seconds after switching on the relay will automatically close; ‘1' means non-auto-close mode, does not automatically close after switching on the relay |
| serialCustomize | String | Serial port customization,The default is"#{idcardNum}#",Unlimited length |
| similarity | Int | similarity,default 80,30~100 adjustments |
| strangerVoiceCustom | String | Stranger voice customization, stranger recognition by default, within 32 characters |
| strangerVoiceMode | Int | Stranger voice mode, default stranger alarm, ‘1' means no voice broadcast, ‘2' means stranger alarm, ‘100' means custom |
| voiceCustom | String | Voice customization, default is "name", within 32 characters |
| voiceMode | Int | Voice mode, ‘1' means no need to broadcast, ‘2' means broadcast name, ‘100' means customization |
| wg | Int | wigan 26、34 |

Return data

#### {

"data": { //Set parameters "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": 26

},

"result": 1, //Interface accessible

 "success": true //Set up successfully

}

## 3.Get parameter configuration

Request mode:Requests use POST

Request address: http:// ip of device:8080/getConfig

Request data:

|  |  |  |  |
| --- | --- | --- | --- |
| **field** | **type** | **must fill** | **Description** |
| pass | String | Y | Device password |

return example

#### {

"data": { //Set parameters "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": 26

},

"result": 1, //Interface accessible "success": true //Set up successfully

}

## 4.Personnel registration

Request mode:Requests use POST

Request address:http:// ip of device:8080/person/create

Request data:

|  |  |  |  |
| --- | --- | --- | --- |
| **field** | **type** | **must fill** | **Description** |
| pass | String | Y | Device password |
| person | Json | Y | Parameter configuration JSON, the following are the parameters of JSON |

Request description:

* POST-request parameters are placed in the body
* The Base64 string of the photo, without header, e.g.data:image/jpg;base64.
* Personnel photos only support JPG and PNG

‘Person' field description

|  |  |  |
| --- | --- | --- |
| **field** | **type** | **Description** |
| age | int | age |
| imgBase64 | String | The Base64 string of the photo |
| name | String | name |
| prescription | String | Passage period (start time and end time separated by commas),yyyy-MM-ddHH:mm,yyyy-MM-dd HH:mm |
| sex | int | sex |
| type | Int | type defaults to ‘1', ‘1' means visitors,' 2' means blacklists, and ‘3' means employees |
| vipID | Int | Create a person's id, find that person by ID |
| welCome | String | Reserved, not required |
| icCard | String | IC card information |
| card | String | ID number |
| wn | String | Reserved, not required |

Return example

{

"data": "msg: Person registered successfully",

"result": 1,

"success": true

}

## 5.Edit personnel information

Request mode:Requests use POST

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

Request data:

|  |  |  |  |
| --- | --- | --- | --- |
| **field** | **type** | **must fill** | **Description** |
| pass | String | Y | Device password |
| person | Json | Y |  Personnel information JSON, the following are the parameters of JSON  |

request description:

* POST-request parameters are placed in the body
* The Base64 string of the photo, without header,e.g.data:image/jpg;base64.
* Personnel photos only support JPG and PNG

Person parameter description

|  |  |  |
| --- | --- | --- |
| **field** | **type** | **Description** |
| age | int | age |
| imgBase64 | String | The Base64 string of the photo |
| name | String | name |
| prescription | String | Passage period (start time and end time separated by commas),yyyy-MM-ddHH:mm,yyyy-MM-dd HH:mm |
| sex | int | sex |
| type | Int | type defaults to ‘1', ‘1' means visitors,' 2' means blacklists, and ‘3' means employees |
| vipID | Int | Create a person's id, find that person by ID |
| welCome | String | Reserved, not required |
| icCard | String | IC card information |
| card | String | ID number |
| wn | String | Reserved, not required |

Return data sample

{

 "data": "msg: People edited successfully",

 "result": 1,

 "success": true

}

## 6.Delete personnel

Request mode:Requests use POST

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

Request data:

|  |  |  |  |
| --- | --- | --- | --- |
| **field** | **type** | **must fill** | **Description** |
| pass | String | Y | Device password |
| deleteId | String | Y | Deleted person id |

Return example

{

"data": "msg: People deleted successfully",

 "result": 1,

"success": true

}

## 7.Get identification record

Request mode:Requests use POST

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

Request data:

|  |  |  |  |
| --- | --- | --- | --- |
| **field** | **type** | **must fill** | **Description** |
| pass | String | Y | Device password |
| startTime | String | Y | Start time:2019-07-11 12:00 |
| endTime | String | Y | End :2019-07-11 17:00 |

Return example

{

"data": [

{

"birthDate": "",//Reserved

"currentTime": 1562819094334, //current Time,Millisecond

"idCardNum": "",// personnel ID

"imageFlag": 0,//reserved

"imageName": "1\_0\_1562819094334.jpg", //The name of the pass photo.to get the pass photo.please call getRecordlmg interface

"name": "pdf", //Name of passer

 "type": 1, //type

"tempratrue": “36.5”, // body temperature while passing, pass ‘1' when body temperature detection is off

"mask": 1 //Whether to wear a mask,'1'means wearing , ‘0' means not wearing,'-1' means Mask detection off.

},

{

"birthDate": "",

"currentTime": 1562824109687, "idCardNum": "",

"imageFlag": 0,

"imageName": "1\_86\_1562824109687.jpg", "name": "pdf",

"type": 1

"tempratrue": “36.5”,

"mask": 1 //Whether to wear a mask,'1'means wearing ,,'1'means wearing , ‘0' means not wearing.

},

{

"birthDate": "",

"currentTime": 1562825856546, "idCardNum": "null", "imageFlag": 1,

"imageName": "100\_86\_1562825856546.jpg", "name": "tts",

"type": 1,

"tempratrue": “36.5” ,

"mask": 1 //Whether to wear a mask,'1'means wearing ,,'1'means wearing , ‘0' means not wearing.

}

],

"result": 1, //Interface accessible

"success": true //success

}

## .Modify LOGO

Request mode:Requests use POST

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

Request data:

|  |  |  |  |
| --- | --- | --- | --- |
| **field** | **type** | **must fill** | **Description** |
| pass | String | Y | Device password |
| imgBase64 | String | Y | Logo Icon Base64 only supports PNG and jpg |

Request description:

* POST-request parameters are placed in the body
* The Base64 string of the photo, without header,e.g.data:image/jpg;base64.
* Personnel photos only support JPG and PNG

Return example

{

"data": "msg: logo Modified success",

 "result": 1,

"success": true

}

## .Get device MAC address

Request mode:Requests use POST

Request address:http:// ip of device:8080/getDeviceMac

Request data:

|  |  |  |  |
| --- | --- | --- | --- |
| **field** | **type** | **must fill** | **Description** |
| pass | String | Y | Device password |

Return example

{

"data": "8CFCA0000023",

"result": 1, "success": true

}

## 10. Temperature and Mask Parameter Configuration

Request mode:Requests use POST

Request address:http:// ip of device:8080/tempAndMaskSetting

Request data:

|  |  |  |  |
| --- | --- | --- | --- |
| **field** | **type** | **must fill** | **Description** |
| pass | String | Y | Device password |
| config | Json | Y | Parameter configuration JSON, following are JSON's parameters |

Request description:

* POST-request parameters are placed in the body
* config- the request parameters are in JSON format, with the following examples (please adjust them to suit your needs)

{

"deviceId": 88,

"id": 10,

"isBodyTempAlarm": 1,

"isBodyTempStart": 1,

"isHighFeverAdopt": 0,

"isLowFeverAdopt": 0,

"isLowTempAdopt": 0,

"isStandardTempAdopt": 1,

"isWearingMask": 0,

"standardBodyTemp": "37.3"

}

config field description

|  |  |  |
| --- | --- | --- |
| **field** | **type** | **Description** |
| isBodyTempAlarm | int | ‘1'-Turn on temperature alarm.‘0'- Turn off temperature alarm |
| isBodyTempStart | Int | ‘1'-Turn on temperature detection. ‘0 ‘Turn off temperature detection |
| standardBodyTemp | String | Temperature threshold 37.3 (accurate to one decimal point). When temperature detection and alarm are turned on, temperature exceed the threshold  |

|  |  |  |
| --- | --- | --- |
|  |  | and sound alarm is played |
| isHighFeverAdopt | int | Pass with high fever or not('0'-not pass '1'-pass)For opening doors(range38.5-43.'0') |
| isLowFeverAdopt | int | Pass with low fever or not('0'-not pass '1'-pass)For opening doors(range37.3-38.5) |
| isLowTempAdopt | int | Pass with low temperature or not('0'-not pass '1'-pass)For opening doors(range below3'0'.'0'-) |
| isStandardTempAdopt | int | Pass with normal temperature or not('0'-not pass '1'-pass)For opening doors(range 36.'1'-37.2) |
| isWearingMask | int | Wearing mask detection switch('0'-NO '1'-YES)) |

Return example

{

"data": { "deviceId": 88,

"id": 10,

"isBodyTempAlarm": 1,

"isBodyTempStart": 1,

"isHighFeverAdopt": 0,

"isLowFeverAdopt": 0,

"isLowTempAdopt": 0,

"isStandardTempAdopt": 1,

"isWearingMask": 0,

"standardBodyTemp": "37.3"

},

"result": 1, "success": true

}

## 11. Get Passage Photos

Request mode:Requests use POST

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

Request data:

|  |  |  |  |
| --- | --- | --- | --- |
| **field** | **type** | **must fill** | **Description** |
| pass | String | Y | Device password |
| imgName | String | Y | Photo name (obtained by identification record interface) |

Return example

{

"data":“/9j/4AAQSkZJRgABAQAAAQABAAD/2wB", //Image's base64 string

"result": 1, //'1'- Interface accessible ‘0'- Interface exception

"success": true // ‘true' -success ‘false' -fail；

}

## 12. Get device parameter information

Request mode:Requests use POST

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

Request data:

|  |  |  |  |
| --- | --- | --- | --- |
| **field** | **type** | **must fill** | **Description** |
| pass | String | Y | Device password |

Return example

{

"data":"{"freeSpace":"3.61GB", // freeSpace: "ip":"192.9.51.45", //address of ip "mac":"8CFCA0036225", // mac address "time":1584518232928, //system time long type "version":"1.5.0.22.0.01" // App current version number

}",

"result": 1, "success": true

}

## 13. Device photo

Request mode:Requests use POST

Request address:http:// ip of device:8080/ photograph

Request data:

|  |  |  |  |
| --- | --- | --- | --- |
| **field** | **type** | **must fill** | **Description** |
| pass | String | Y | Device password |

Return example

{

"data": "/9j/4AAQSkZJRgABAQAA …", //base64 string, need to be converted to png suffix image

"result": 1, "success": true

}

## 14. Device initialization

Request mode:Requests use POST

Request address:http:// ip of device:8080/ initialization

Request data:

|  |  |  |  |
| --- | --- | --- | --- |
| **field** | **type** | **must fill** | **Description** |
| pass | String | Y | Device password |

request description:

 Delete all identification data, personnel data, characteristics and other data on the device, and empty all databases.

 Delete attributes set through the device configuration interface

Return example

{

"data": "initialized successfully", "result": 1,

"success": true

}

## 15. Set the system time

Request mode:Requests use POST

Request address:http:// ip of device:8080/ setDeviceTime

Request data:

|  |  |  |  |
| --- | --- | --- | --- |
| **field** | **type** | **must fill** | **Description** |
| pass | String | Y | Device password |
| year | String | Y | year |
| month | String | Y | month |
| day | String | Y | day |
| hours | String | Y | hour(24-hour clock) |
| minute | String | Y | minute |

request description:

* After successful configuration, the device time will be changed to the currently set time
* If the device is connected to the public network, the device has a network time calibration mechanism, and the system adjusts the device time to the public network time.
* For the device to display the manually set time, the device must be on a 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", //Current device time

"result": 1, "success": true

}

## 16. Set the identification record interface callback

Request mode:Requests use POST

Request address:http:// ip of device:8080/ setIdentifyCallback

Request data:

|  |  |  |  |
| --- | --- | --- | --- |
| **field** | **type** | **must fill** | **Description** |
| pass | String | Y | Device password |
| callbackUrl | String | Y | Identify the url interface for record uploads |

request description:

* set callback in other settings of APP
* 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 up successfully, ‘-1’- no callbackUrl field ,

‘-2'-callbackUrl field is empty

"success": true //'true' - Set up successfully,‘false' - set up failed；

}