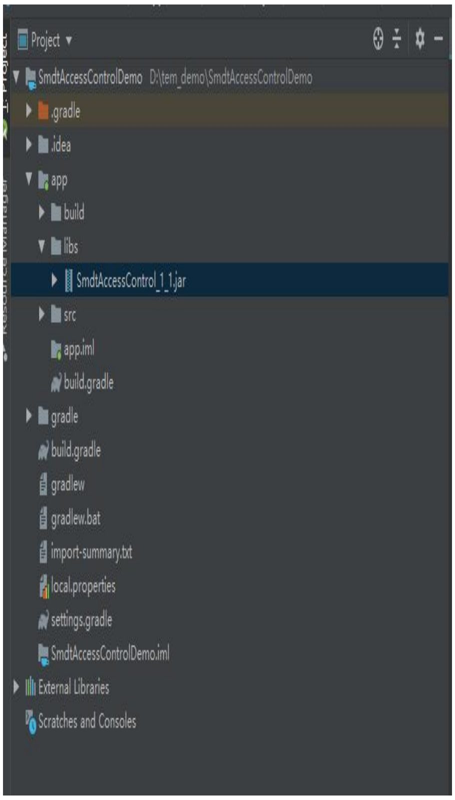
API document

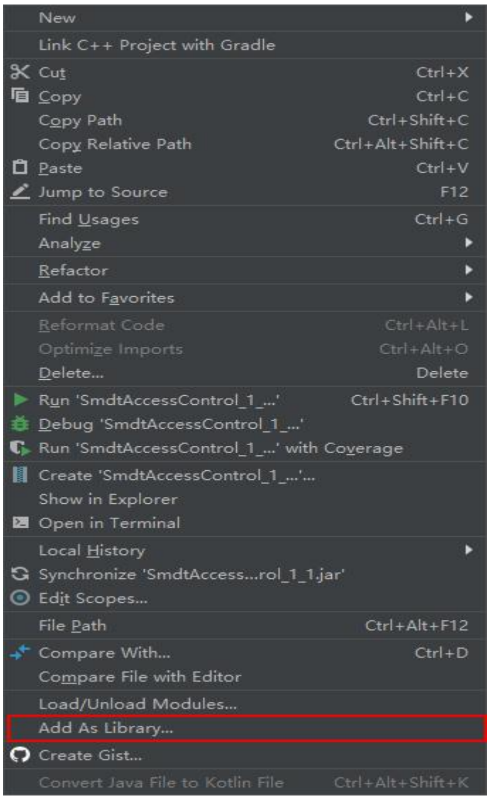
Profile

1. Version and modify record..错误！未定义书签。
2. Copyright declaration..错误！未定义书签。
3. Profile...错误！未定义书签。
4. how to use SMDT API in Android Studio..2
5. SDK interface function...3
5.1. public int smdtSendCard(String idCard, int transformat)......................................3
DESCRIPTION: Wiegand interface operation, parameter ID card number。错误！未定义书签。
5.2. public int smdtSendCard(String HID_value, String PID_value, int transformat).3
DESCRIPTION: Wiegand interface operation, the parameters for the Open Code Hidden Code。..错误！未定义书签。
5.3. public String smdtReadWiegandData()..4
DESCRIPTION: Get Wigan input, which is a blocking method, using the following parameters。...4
5.4. public int smdtReleaseWiegandRead()...5
Description: the active exit blocking method smdtRead Wieganddata () is not available until version 1.5。..5
5.5. public String[] smdtGetCameraVidPid(int cameraId)...5
Description: Read Camera Vid, PID interface..................错误！未定义书签。
5.6. public int setLedLighted(String ledColor, boolean lighted)......................................6
DESCRIPTION: FILL LAMP OPERATION NEW API -- White Light, red light, green light. can be used after Version 1.5 ....错误！未定义书签。
5.7. public int setControl(int type, int values)..6
DESCRIPTION: FILL LAMP OPERATION-WHITE LAMP。(2019-12-26 Later versions are recommended setLedLighted)......................错误！未定义书签。
5.8. public int smdtSetUsbPower(int type, int num, int values)......................................7
DESCRIPTION: FILL LAMP OPERATION-RED LAMP。(2019-12-26 Later versions are recommended setLedLighted)......................错误！未定义书签。
5.9. public int smdtSetGpioDirection(int gpioNumber, int direction, int value).............7
DESCRIPTION: FILL LAMP OPERATION-Green LAMP。(2019-12-26Later versions are recommended setLedLighted)....................错误！未定义书签。
5.10. public int setRelayIoMode(int mode, int delay);...8
Description: relay mode and delay settings。..................错误！未定义书签。
5.11. public int setRelayIoValue(int value);..9
Description: open or close relay。.....................................错误！未定义书签。
5.12. public int getRelayIoMode();..9
Description: in non-automatic mode, obtain the status of the relay错误！未定义书签。

1. How to use SMDT In Android Studio

1. copy SmdtAccessControl_1_1.jar to【item demo\app\libs\】；
2. Press right button click “jar” in libs file,chose file “add as Library”...





Note：all api Invoke

SmdtManager smdt = SmdtManager.create(this);

```
Start to use SMDT API
  The SmdtManager object is first declared, and then you can start using the API.
Here's an example:
/declare SmdtManager object
private SmdtManager smdt;
smdt = SmdtManager.create(this);

//use API
smdt.setRelayIoValue(0);
```

2. SDK Interface function

2. 1. **public int smdtSendCard(String idCard, int transformat)**

DESCRIPTION: Wiegand interface operation, parameter ID card number.

API version implementing this interface: V1.1

Parameter name / return value	type	instruction	For example
idCard	String	Card number	2147584137
transformat	int	Format：Wiegand 26=1，Wiegand 34=2；	1
return value	int	success: 0, Fail:-1	

Example：

```
SmdtManager smdt = SmdtManager.create(this);
smdt.smdtSendCard("2147584137", 1);
```

2. 2. **public int smdtSendCard(String HID_value, String PID_value, int transformat)**

DESCRIPTION: Wiegand interface operation, the parameters for the Open Code Hidden Code.

API version implementing this interface: V1.0

Parameter name / return value	type	instruction	For example
HID_value	String	Hide core	001
PID_value	String	Public code	34953
transformat	int	Format: Wiegand 26 = 1, Wiegand 34 = 2;；	1
Return value	int	Success: 0, Fail:-1	

For example：

```
SmdtManager smdt = SmdtManager.create(this);
smdt.smdtSendCard("001", "34953", 1);
```

2. 3. public String smdtReadWiegandData()

DESCRIPTION: Get Wiegand input, which is a blocking method, using the following parameters。

Implement the API version of this interface：V1.0

Parameter name / return value	Type	Instruction	For example
Return value	String	Back to Wiegand card number	4017669612

Note: blocking method, Swipe a card will trigger a time, it is recommended that the Service inside the open thread has been listening

EXAMPLE:

```
private class ReadThread extends Thread {
    @Override
    public void run() {
        while (!isInterrupted()) {
            String result = smdt.smdtReadWiegandData();
            if (result == null || result.equals("")||                    result.equals("0")) {
                Log.i(TAG, "result ==== null...........");
                continue;
            }
            Log.i(TAG, "result =======" + result);
        }
    }
}
```

Using：

```
mReadThread = new ReadThread();
mReadThread.start();
```

Ending：

```
@Override
protected void onStop() {
    super.onPause();
```

```
   if (mReadThread != null) {
       mReadThread.interrupt();
//1.5 And later versions can use the downlink code to exit the blocking function in timesmdtReadWiegandData()
       smdt.smdtReleaseWiegandRead();
   }
}
```

2. 4. public int smdtReleaseWiegandRead()

Description: active unblocking method smdtReadWiegandData(),Only later 1.5 versions are available。

Implement the API version of this interface：V1.5

Parameter name / return value	type	instruction	For example
return value	int	Success：0 Fail：-1	4017669612

2. 5. public String[] smdtGetCameraVidPid(int cameraId)

Description: Read Camera Vid, PID interface

API version implementing this interface: v1.0

Parameter name / return value	type	Instruction	For example
cameraId	int	Camera ID value	0
Return value	String[]	string[0] means vid,string[1] means pid	

Note point:

For example：

```
SmdtManager smdt = SmdtManager.create(this);
String[] results = smdt.smdtGetCameraVidPid(0);
```

```
Result：results[0]=34b3 ,results[1]=53a9
```

2.6. public int setLedLighted(String ledColor, boolean lighted)

Description: fill light operation (new API) - white light, red light, green light. Available after 1.5 version

API version to implement this interface: v1.5

Parameter name / return value	type	instruction	For example
ledColor	String	SmdtManager.LED_WHITE white lamp SmdtManager.LED_RED red lamp SmdtManager.LED_GREEN Green lamp	
lighted	boolean	true: light on false:light off	1
Return value	int	Success: 0, Fail:-1	

Example：

```
SmdtManager smdt = SmdtManager.create(this);
smdt .setLedLighted(SmdtManager.LED_WHITE, true);
```

2.7. public int setControl(int type, int values)

Description: fill light operation - white light. (after December 26, 2019, setledlighted is recommended)

API version implementing this interface: v1.0

Parameter name / return value	type	instruction	Example

type	int	Fixed 3	
values	int	0: off, 1: on	1
Return value	int	Success: 0, Fail:-1	

Example：

```
SmdtManager smdt = SmdtManager.create(this);
smdt.setControl(3,  0);
```

2. 8. public int smdtSetUsbPower(int type, int num, int values)

Description: fill light operation - red light. (after December 26, 2019, setledlighted is recommended)

API version implementing this interface: v1.0.

Parameter name / return value	type	instruction	Example
type	int	Fixed 1	
num	int	Fixed 3	
values	int	0: off, 1: on	1
Return value	int	Success: 0, Fail:-1	

Example：

```
SmdtManager smdt = SmdtManager.create(this);
smdt.smdtSetUsbPower(1, 3, 1);
```

2. 9. public int smdtSetGpioDirection(int gpioNumber, int direction, int value)

Description: fill light operation - green light. (after December 26, 2019, setledlighted is recommended)

API version implementing this interface: v1.0

Parameter name / return	Type	Instruction	Example

value			
gpioNumber	int	Fixed 4	
direction	int	Fixed 1	
values	int	0: off, 1: on	1
Return value	int	Success: 0, Fail:-1	

Example：

```
SmdtManager smdt = SmdtManager.create(this);
smdt.smdtSetGpioDirection(4, 1, 1 );
```

2.10. public int setRelayIoMode(int mode, int delay);

Description: relay mode and delay setting.

API version to implement this interface: v1.1

Parameter name / return value	Type	Instruction	Example
mode	int	mode： 0: it means that it will not close automatically, that is, it will not close automatically after opening the relay 1: Indicates auto close mode (high effective - default low level, then high level x seconds, last low level), that is, delay x seconds after opening the relay will automatically close 2: Indicates auto close mode (low effective - default low level, then high level x seconds, then low level), that is, delay x seconds after opening the relay will automatically close	
delay	int	Delay in auto close mode,unit as “s”, Max 63s	5
Return	int	Success: 0, Fail:-1	

Example：

```
SmdtManager smdt = SmdtManager.create(this);
smdt.setRelayIoMode(1,5); //Auto mode, five seconds auto off
smdt.setRelayIoValue(1); //Both 0 and 1 can turn on the relay in automatic mode
```

2. 11. public int setRelayIoValue(int value);

Description: turns the relay on or off.

API version to implement this interface: v1.1

Parameter name / return value	Type	Instruction	Example
value	int	Non electric close mode, 0 for close relay, 1 for open relay. In auto close mode, calling this method will open the relay.	
delay	int	Delay in auto close mode,unit as “s”, Max 63s	5
Return value	int	Success: 0, Fail:-1	

Example:

```
SmdtManager smdt = SmdtManager.create(this);
smdt.setRelayIoMode(0,5); //Non automatic mode, invalid delay parameter
smdt.setRelayIoValue(1); //Open relay
smdt.setRelayIoValue(0); //close relay
```

2. 12. public int getRelayIoMode();

Description: in non automatic mode, obtain the status of the relay

API version implementing this interface: v1.0

Parameter name / return value	type	Instruction	example
return value	int	0 stands for close，1 stands for open	

Example：

```
SmdtManager smdt = SmdtManager.create(this);
int status = smdt.getRelayIoMode(); //Get the status of the relay
```