

SAM4S Android ECR API Ver.1.1
Instructions Manual

SAM4S

CONTENTS

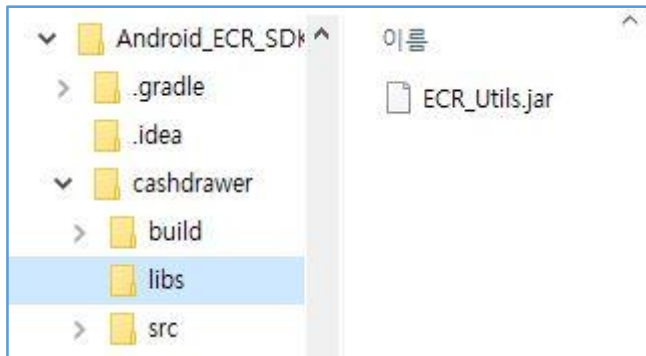
1. BUILD DEVELOPMENT ENVIRONMENT	2
1.1. Android Studio	2
2. API	3
2.1. Init API	3
2.2. Printer	4
2.3. Rear Display	9
2.4. Keyboard	10
2.5. Cashdrawer	11

1. Build development environment

1.1 Android studio

1.1.1 Load library (jar) file

Put ECR_Utils.jar into the libs, as shown in the figure



Add to library, use `compileOnly`

```
dependencies {  
    implementation 'androidx.appcompat:appcompat:1.7.0'  
    implementation 'com.google.android.material:material:1.12.0'  
    implementation 'androidx.constraintlayout:constraintlayout:2.1.4'  
    testImplementation 'junit:junit:4'  
    androidTestImplementation 'androidx.test.ext:junit:1.2.1'  
    androidTestImplementation 'androidx.test.espresso:espresso-core:3.6.1'  
    compileOnly files('libs\\ECR_Utils.jar')  
}
```

2. API

2.1. Init API

Use initial initialization,you must do it first

```
1. PtUtil.init();
```

2.2. Printer

2.2.1. Query Printer State

```
1. int status = PtUtil.quePrinterState();
2. //status value:
3. //3: hot and no paper
4. //2: hot
5. //1: no paper
6. //0: normal
7. //-1: query fail
8. //-2: time out
```

2.2.2. Set Printer Bold

Parameter 0: Normal
1: Bold

```
1. int ret = PtUtil.setPrinterBold(true);
2. if (ret == 0){
3.     //success
4. }
```

2.2.3. Set Printer Alignment Type

Parameter 0: Left
1: Center
2: Right

```
1. int ret = PtUtil.setPrinterAlignType(type); //type:0~2
2. if (ret == 0) {
3.     //success
4. }
```

2.2.4. Set Printer Concentration

```
1. int ret = PtUtil.setPrinterConcentration(con); //con:0~4
2. if (ret == 0) {
3.     //success
4. }
```

2.2.5. Printer UPCA

```
1. int ret = PtUtil.printerUPCA(content); //the range of content is 0~9,and the length of
content must be 11
2. if (ret == 0) {
3.     //success
4. }
```

2.2.6. Printer UPCE

```
1. int ret = PtUtil.printerUPCE(content); //the range of content is 0~9,and the length of
content must be 8,first word must be 0
2. if (ret == 0) {
3.     //success
4. }
```

2.2.7. Printer EAN8

```
1. int ret = PtUtil.printerEAN8(content); //the range of content is 0~9,and the length of
content must be 7
2. if (ret == 0) {
3.     //success
4. }
```

2.2.8. Printer EAN13

```
1. int ret = PtUtil.printerEAN13(content); //the range of content is 0~9, and the length of
content must be 12
2. if (ret == 0) {
3.     //success
4. }
```

2.2.9. Printer CODE39

```
1. int ret = PtUtil.printerCODE39(content); //the range of content is
'0'~'9', 'A'~'Z', '$', '+', '-', '.', '/', '|', '%'
2. if (ret == 0) {
3.     //success
4. }
```

2.2.10. Printer CODE93

```
1. int ret = PtUtil.printerCODE93(content); //the range of content is '0'~'9',
'A'~'Z', '$', '+', '-', '.', '/', '|', '%'
2. if (ret == 0) {
3.     //success
4. }
```

2.2.11. Printer ITF

```
1. int ret = PtUtil.printerITF(content); //the range of content is '0'~'9', and the length
of content must be even
2. if (ret == 0) {
3.     //success
4. }
```

2.2.12. Printer CODABAR

```
1. int ret = PtUtil.printerCODABAR(content); //the range of content is '0'~'9',  
   'A'~'Z','$','+','-','.', '/'  
2. if (ret == 0) {  
3.   //success  
4. }
```

2.2.13. Printer CODE128

```
1. int ret = PtUtil.printerCODE128(content); //the range of content is ASCII 0~127, and the  
   length of content Less than 14  
2. if (ret == 0) {  
3.   //success  
4. }
```

2.2.14. Printer QR Code

```
1. int ret = PtUtil.printQR(content, zoomIn); //zoomIn: set size of QR code , must be 2~7  
2. if (ret == 0) {  
3.   //success  
4. }
```

2.2.15. Printer Picture

```
1. int ret = PtUtil.printerPicture(bitmap, width, height); //Width: width of print, must be  
   less than or equal to 384(paper width) Height: height of print  
2. if (ret == 0) {  
3.   //success  
4. }
```


2.2.16. Printer Picture By Relative Path

```
1. int ret = PtUtil.printerPictureByRelativePath(path, width, height, class);
2. //path:path of picture such us "/res/drawable/test.png",width: width of print,must be
less than or equal to 384(paper width) height: height of print,Class:the class of use
this methods
3. if (ret == 0) {
4.     //success
5. }
```

2.2.17. Printer Picture By Absolute Path

```
1. int ret = PtUtil.printPictureByAbsolutePath(path, width, height);
2. //path:path of picture such us "/sdcard/test.png",width: width of print,must be less
than or equal to 384(paper width) height: height of print
3. if (ret == 0) {
4.     //success
5. }
```

2.2.18. Printer String

```
1. PtUtil.printerString(content, size, ttfAbsolutePath);
2. //size:font size,content:display content,ttfAbsolutePath:ttf path
```

2.3. Rear Display

2.3.1. Full display

```
1. PtUtil.digitalInteger(num);
```

2.3.2. Character display

```
1. PtUtil.digitalSinger(index, char);  
2. //index:Display position char:A single character to display
```

2.3.3. Partial display

```
1. PtUtil.digitalShow(start, end, string);  
2. //start:Start position end:End position string:Display string
```

2.3.4. Clean

```
1. PtUtil.digitalClean();
```

2.4. Keyboard

2.4.1. Set Scan Listener

```
1. PtUtil.setScanListener(new ScanListener() {  
2.     @Override  
3.     public void onResult(int res, String result) {  
4.         //result:Keyboard position  
5.     }  
6. });
```

2.4.2. Cancel Scan

```
1. PtUtil.setScanListener(null);
```

2.5. Cashdrawer

2.5.1. Status

```
1. int value = PtUtil.getGpioIn(109);
2. if (value == 1) {
3.     //opened
4. } else if (value == 0) {
5.     //closed
6. } else {
7.     //not found box
8. }
```

2.5.2. Open

```
1. PtUtil.setGpio(110, 1);
2. try {
3.     Thread.sleep(1000);
4. } catch (InterruptedException e) {
5.     e.printStackTrace();
6. }
7. PtUtil.setGpio(110, 0);
8. //Gpio 110 Can not be high for a long time
```