

Scanner Application Instructions

This scanner application applies to the following PaceBlade models: SDT-121, MDT-801 and LDT-101.

LDT-101



MDT-801



SDT-121



Applications and Scan key installation

1. Please upgrade the device to firmware version 1.5.2

2. Next step is to apply your own perso to the device
For this, you can make a configuration file on the PaceBlade remote server. Remote Server URL:
<https://fw.paceblade.hk/paceblade/web/#/login>

a) Start with creating a new configuration file.
In the remote server, you can find it by clicking on Management
->Setting ->Add new

Path	owner	Create Time	Action
update_config_md085_finet_scanner_509954463.zip	topicon	2023-11-14 09:34:26	Edit Copy Delete

b) You will find the file ScannerService.apk on your screen, please download and install this file.



ScannerService_20231110.apk

c) In the key_map user option, use "b:hk.paceblade.scan" as remap code.

key_map:

b:hk.topicon.scan

For instance, if you want to use the P3, P2 and P1 hardware buttons of the MDT-801 or LDT-101 as 'A', 'B' and 'Scanner'. Then you would have to install it in the key_map field like this: "A,B,b:hk.paceblade.scan".

key_map:

A,B,b:hk.topicon.scan

3. Next step is to install the ScannSetup.apk file through the perso in your custom configuration (MDM) or do this manually via a SD card. You can configure general settings of the scan engine, such as selecting the scan mode, entering fixed prefix or suffix values etc.

Instruction of the Scanner Setup application

General settings



6:31 ▲ 49%

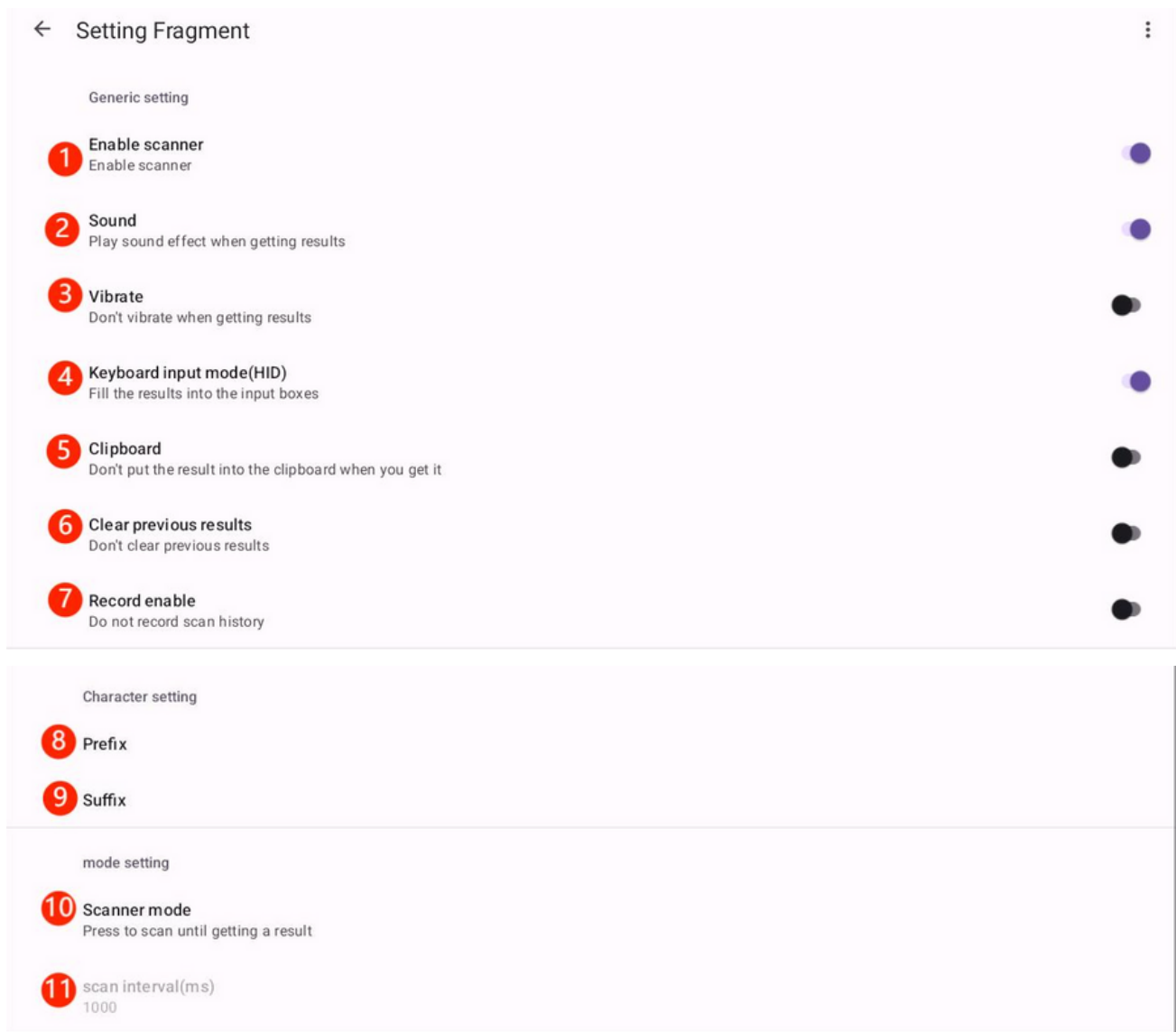
Scanner Fragment ⋮

2023-11-14 18:28:47	9311770599283
2023-11-14 18:28:53	9311770599283
2023-11-14 18:29:19	9311770599283
2023-11-14 18:29:59	6971242826557
2023-11-14 18:30:09	https://ucmp.sf-express.com/wxaccess/weixin/activity/wxapp_b2sf_order?p1=SF1513564940310

Scanner Fragment

General settings

Scanner result



① Enable and disable scanner

With this you can activate and deactivate the scanner module of the device.

② A function to activate and deactivate the scan sound of the device.

③ A function to activate and deactivate the vibration function of the device.


④ Keyboard input mode(HID)

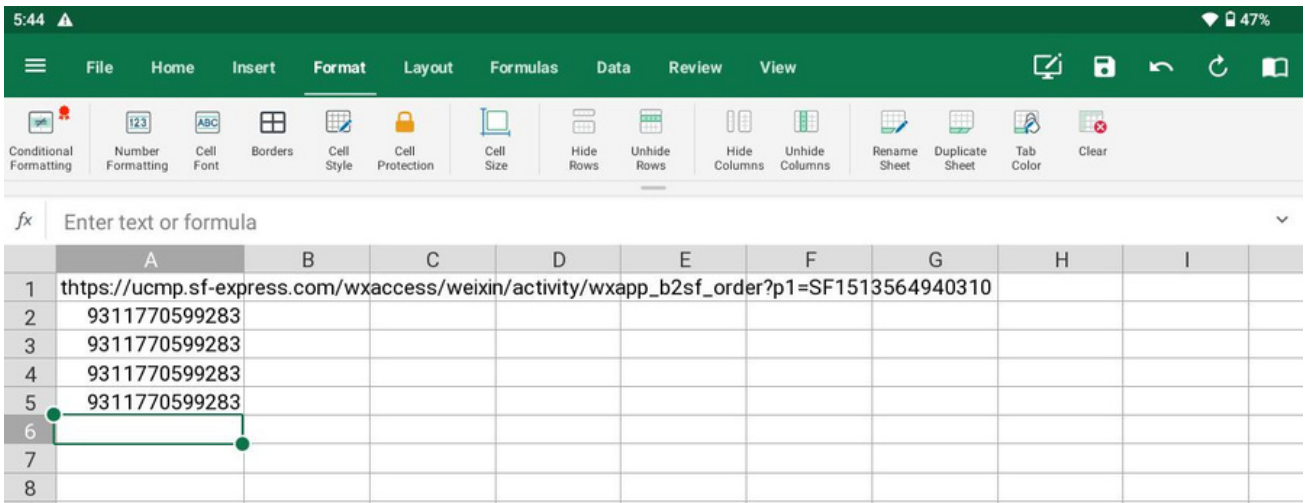
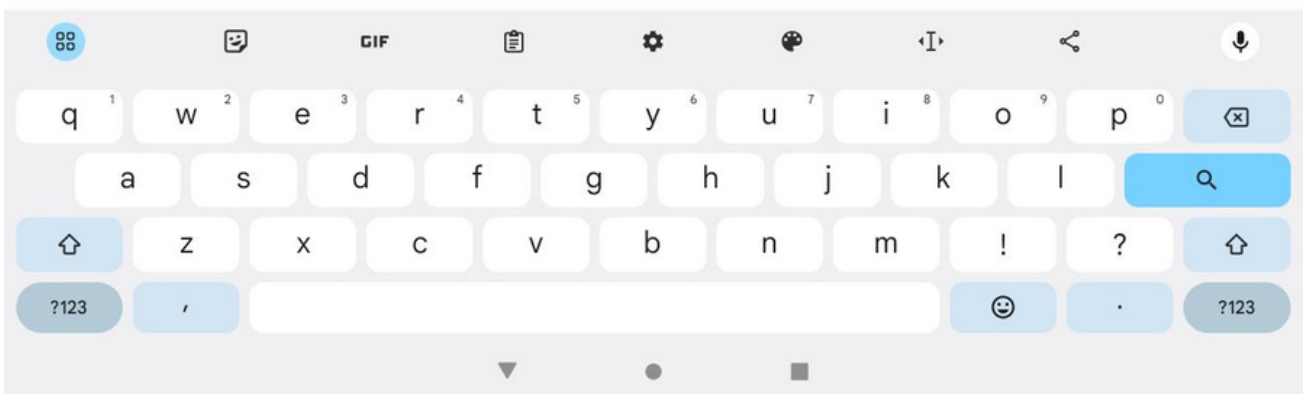
When this function is enabled, the information of the scandata will automatically be filled in to the empty input boxes.

For example, the scan information can be filled in to a search bar, document or spreadsheet.

5:46 ▲

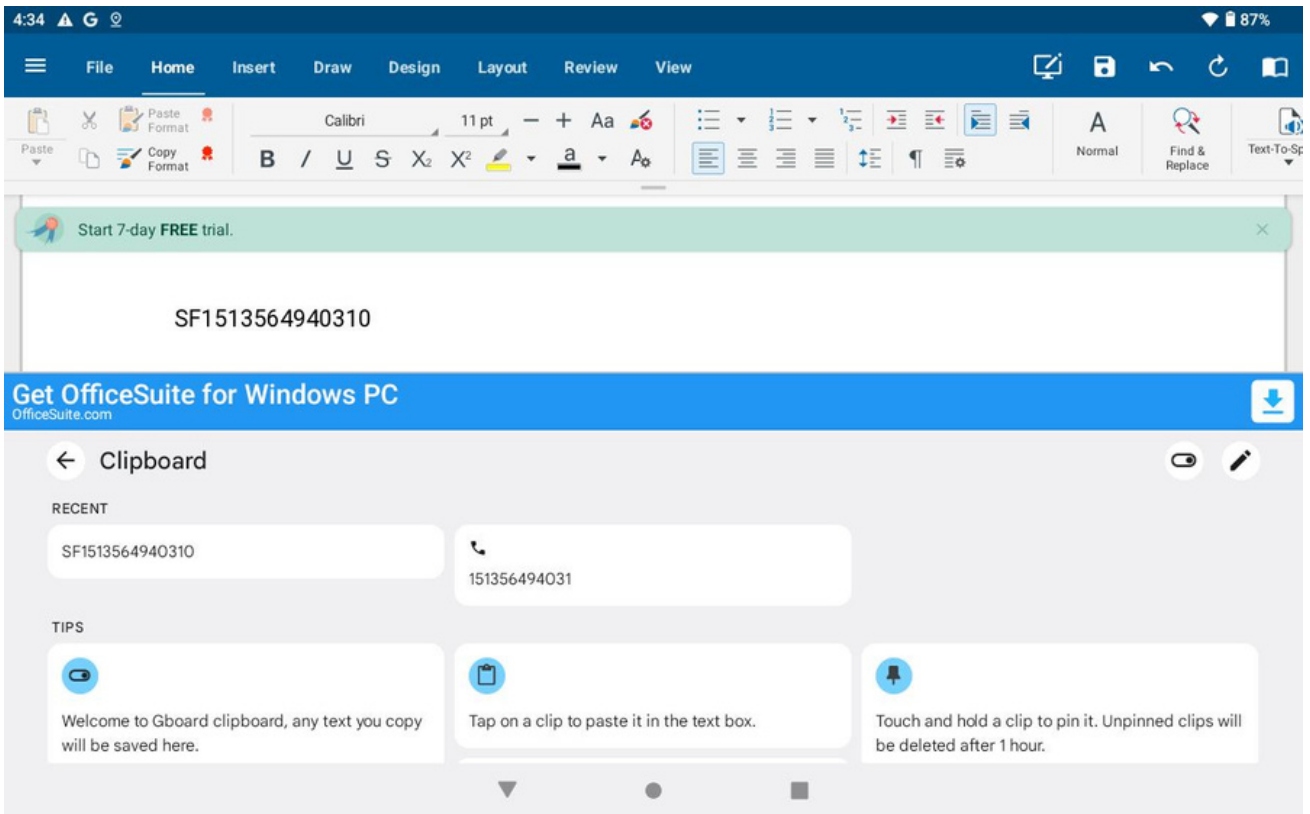
📶 47%

 https://ucmp.sf-express.com/wxaccess/weixin/activity/wxapp_b2sf_order? ✕



⑤ Clipboard

When this function is enabled, the scanned data will be pasted to the clipboard.

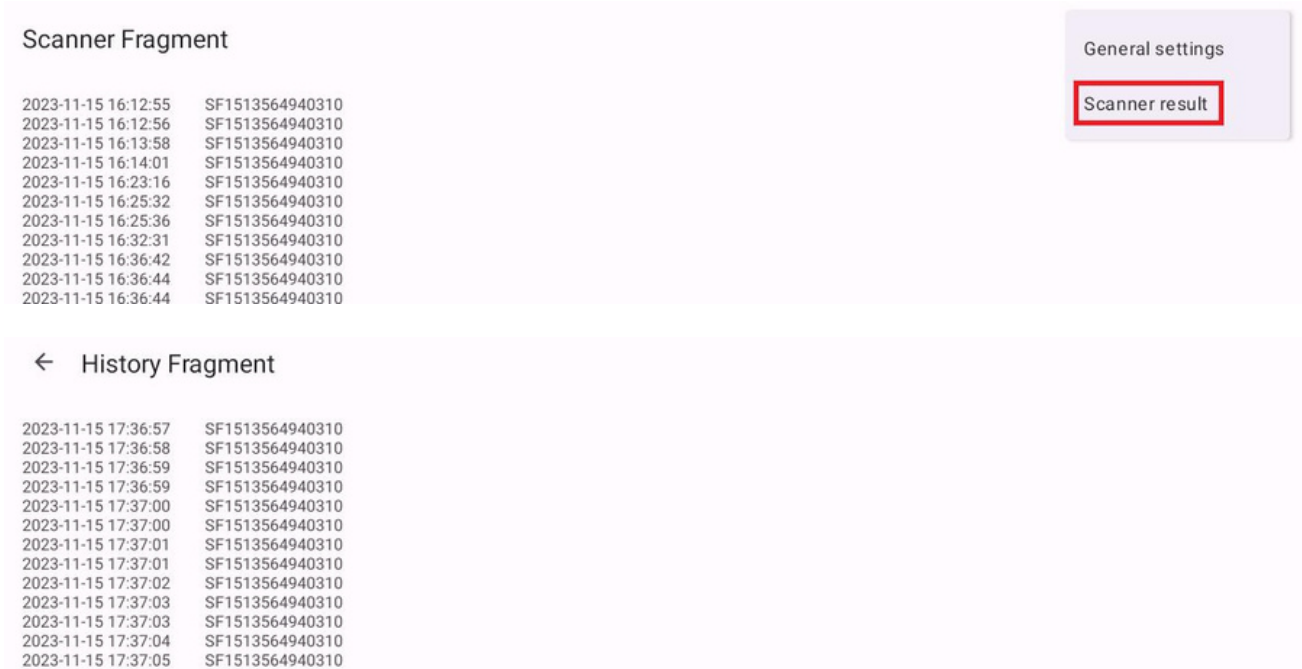


⑥ Clear previous results option

When this function is enabled, only the currently scanned data will be retained, while previously scanned data will be cleared.

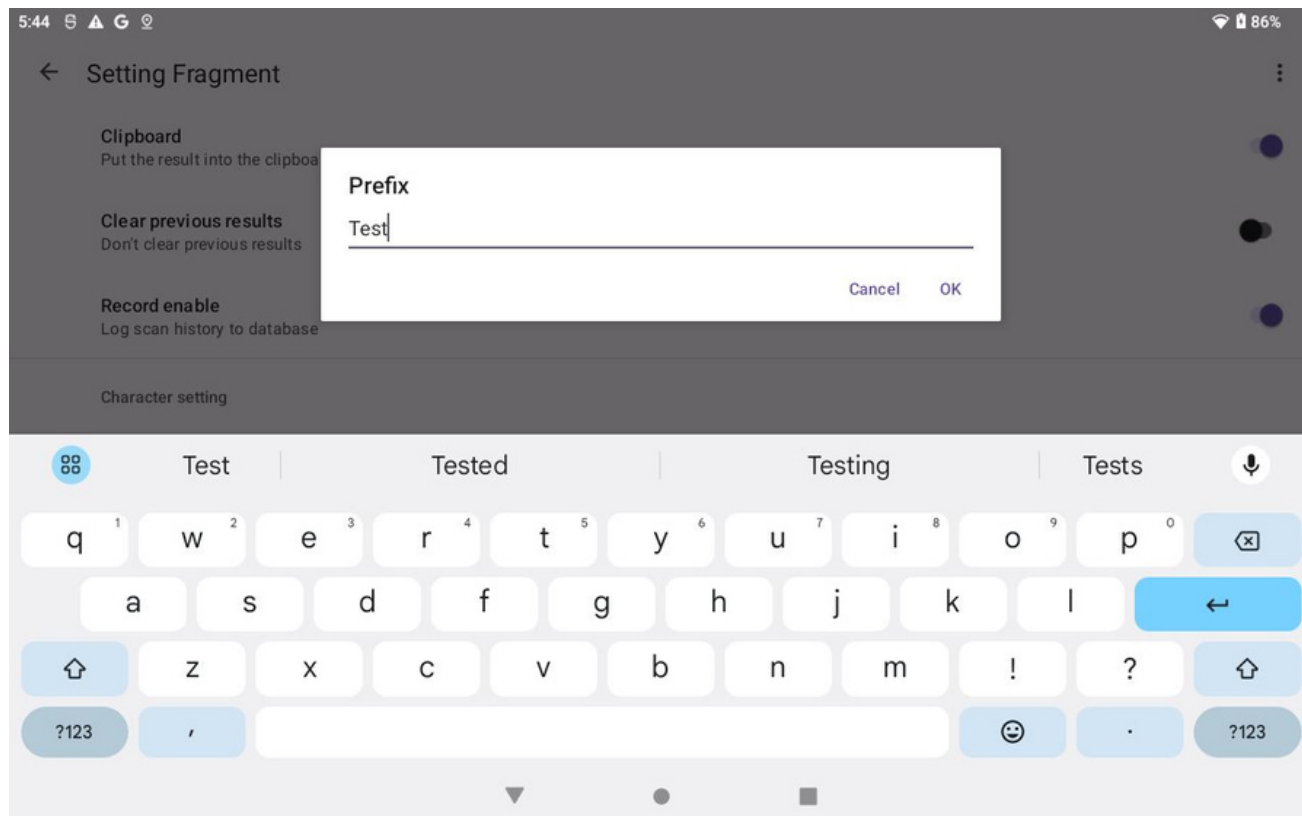
⑦ Record enable

When this function is enabled, the scan history will be saved in the History Fragment.

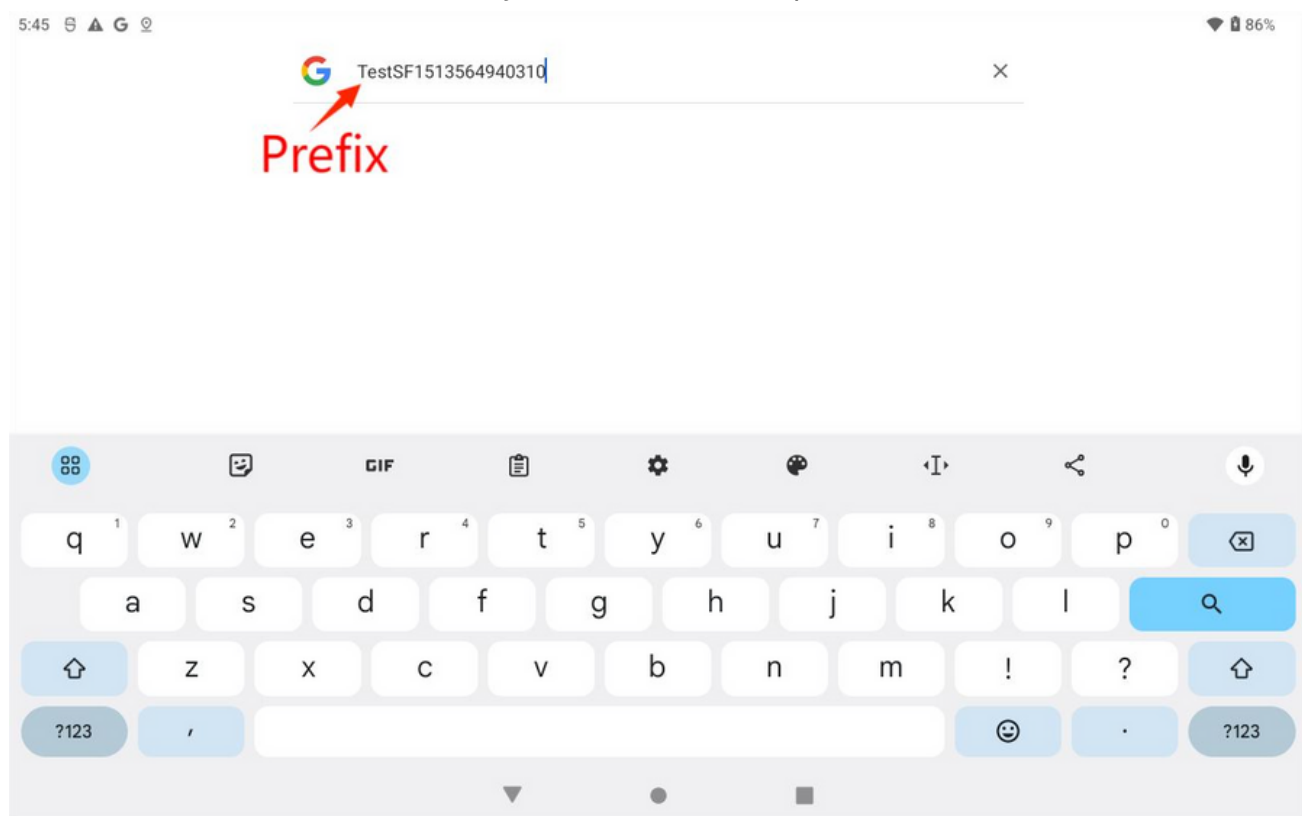


⑧ Prefix

With this function you can set a prefix for the data you will be scanning.

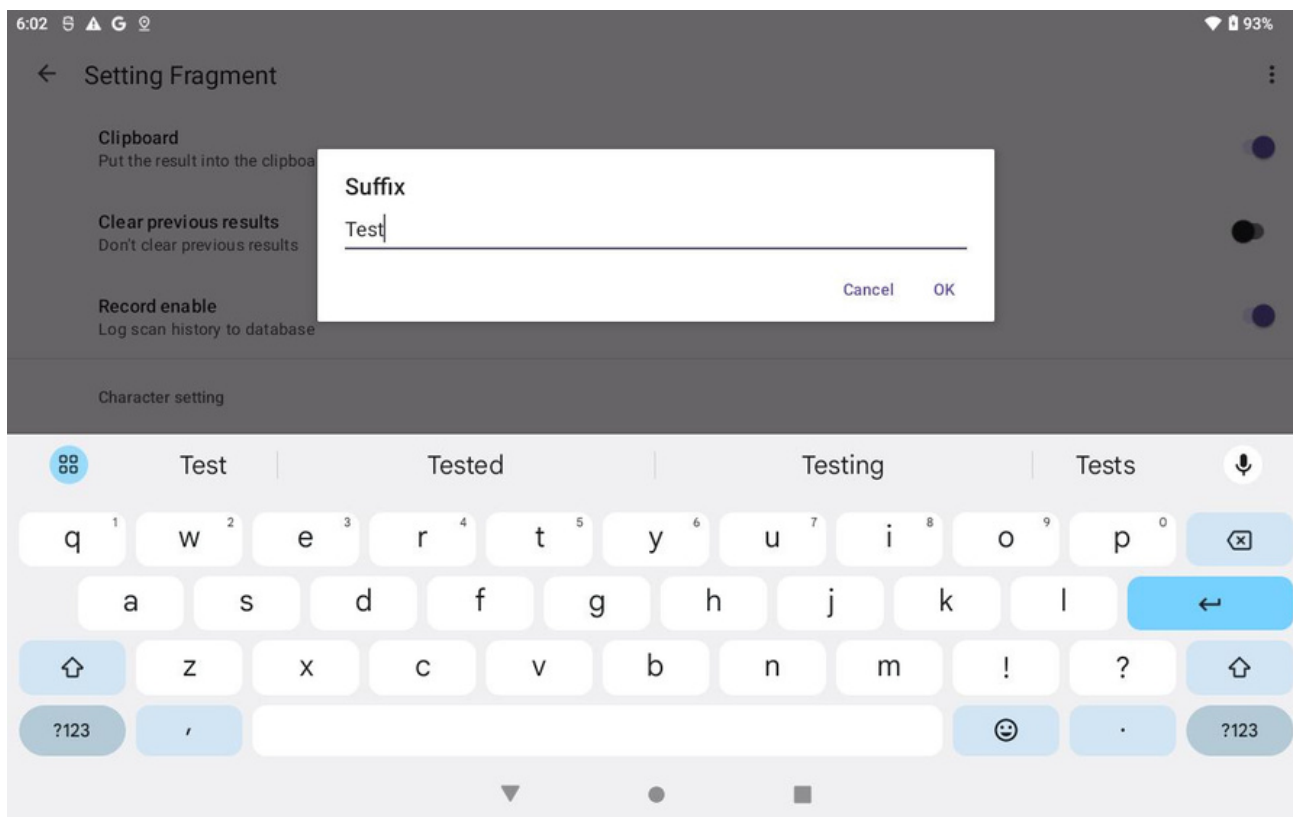


The scanned data will automatically be added after the prefix.

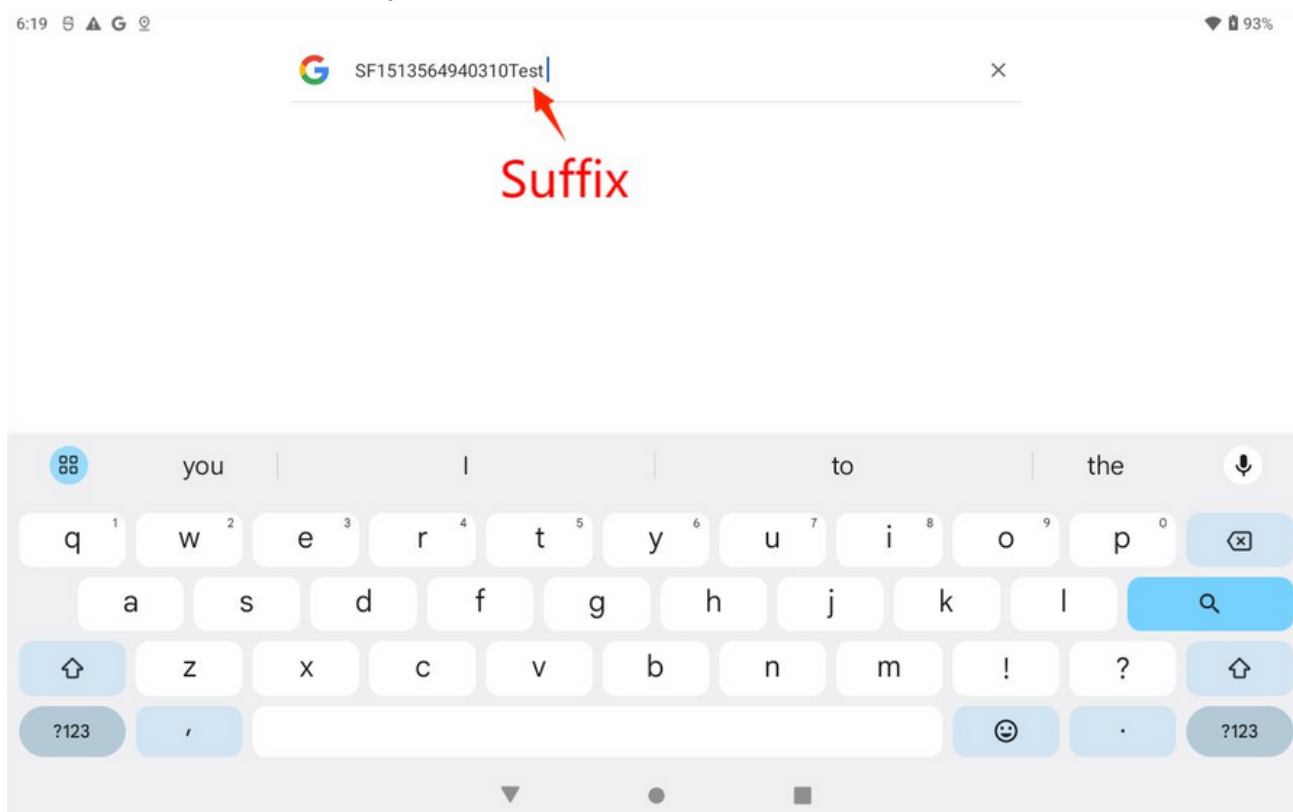


⑨ Suffix

With this function you can set a suffix for the data you will be scanning.

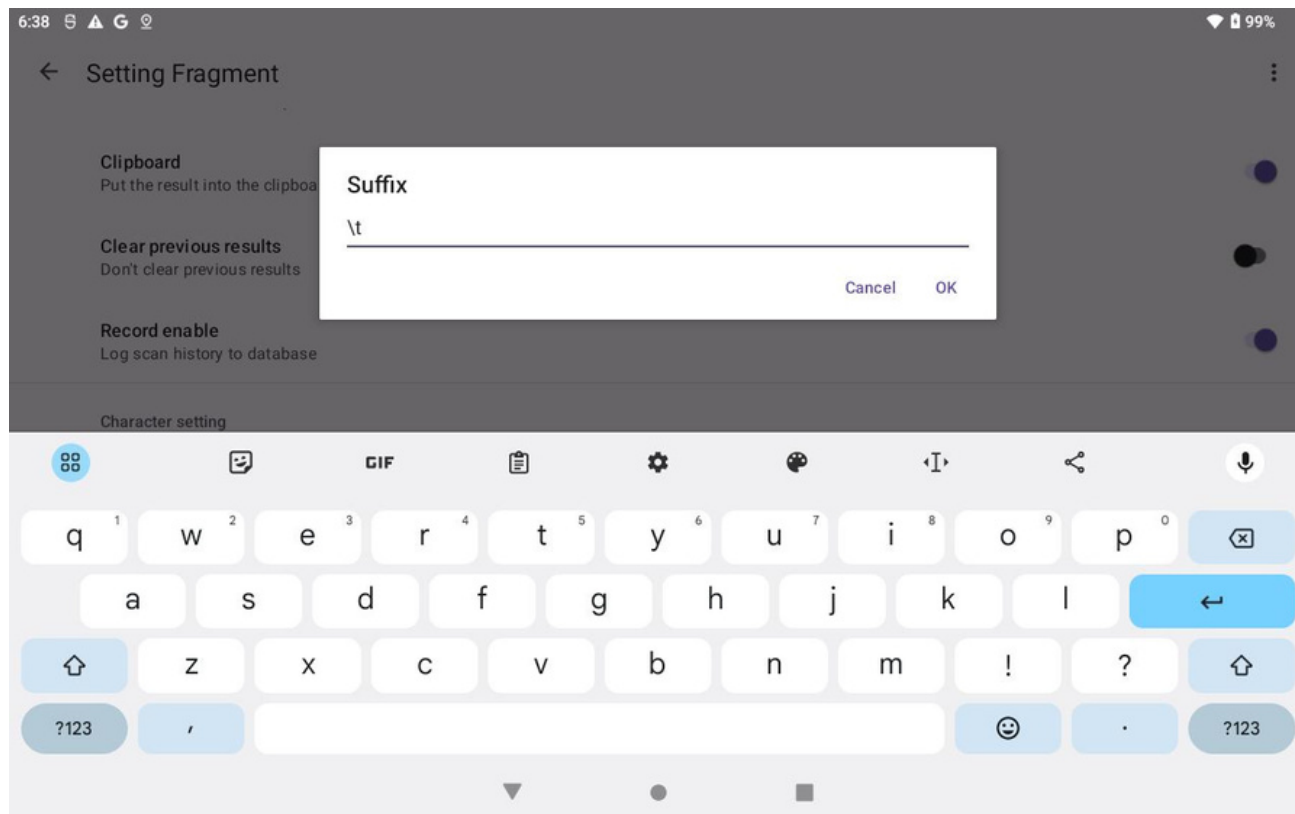


The suffix will automatically be added after the scanned data.

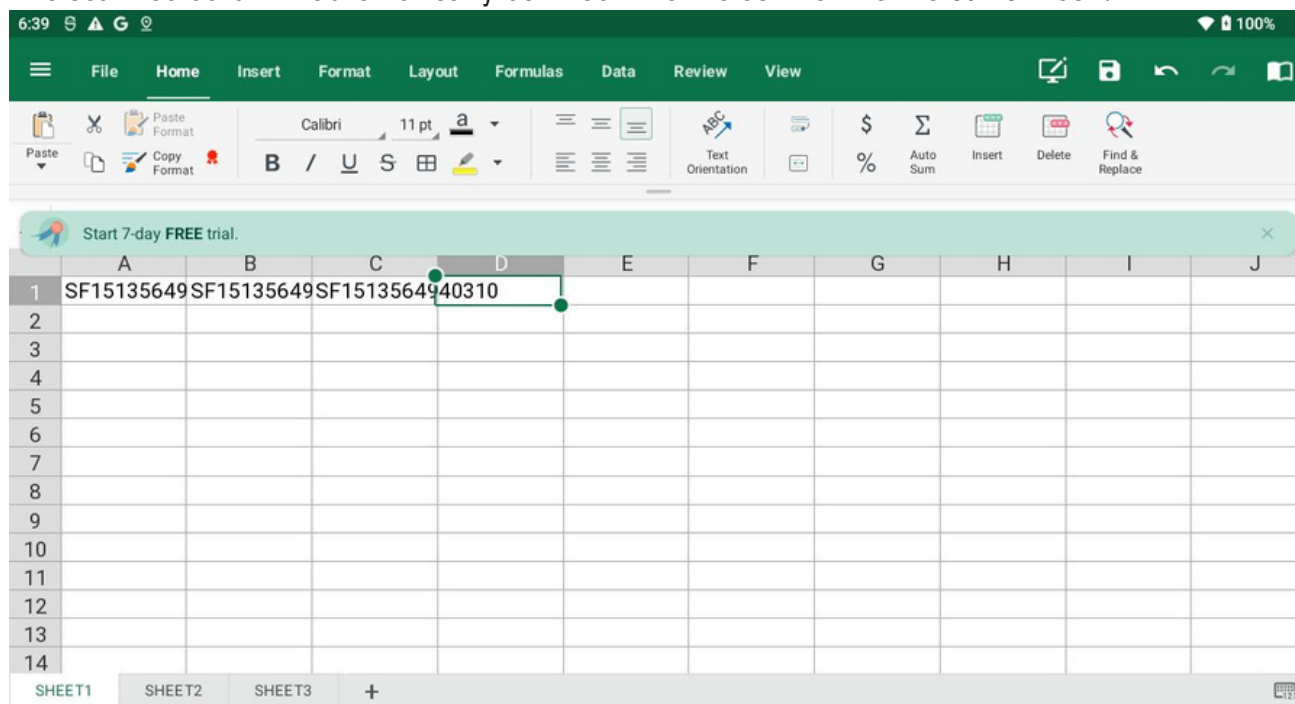


Set a special key in the suffix

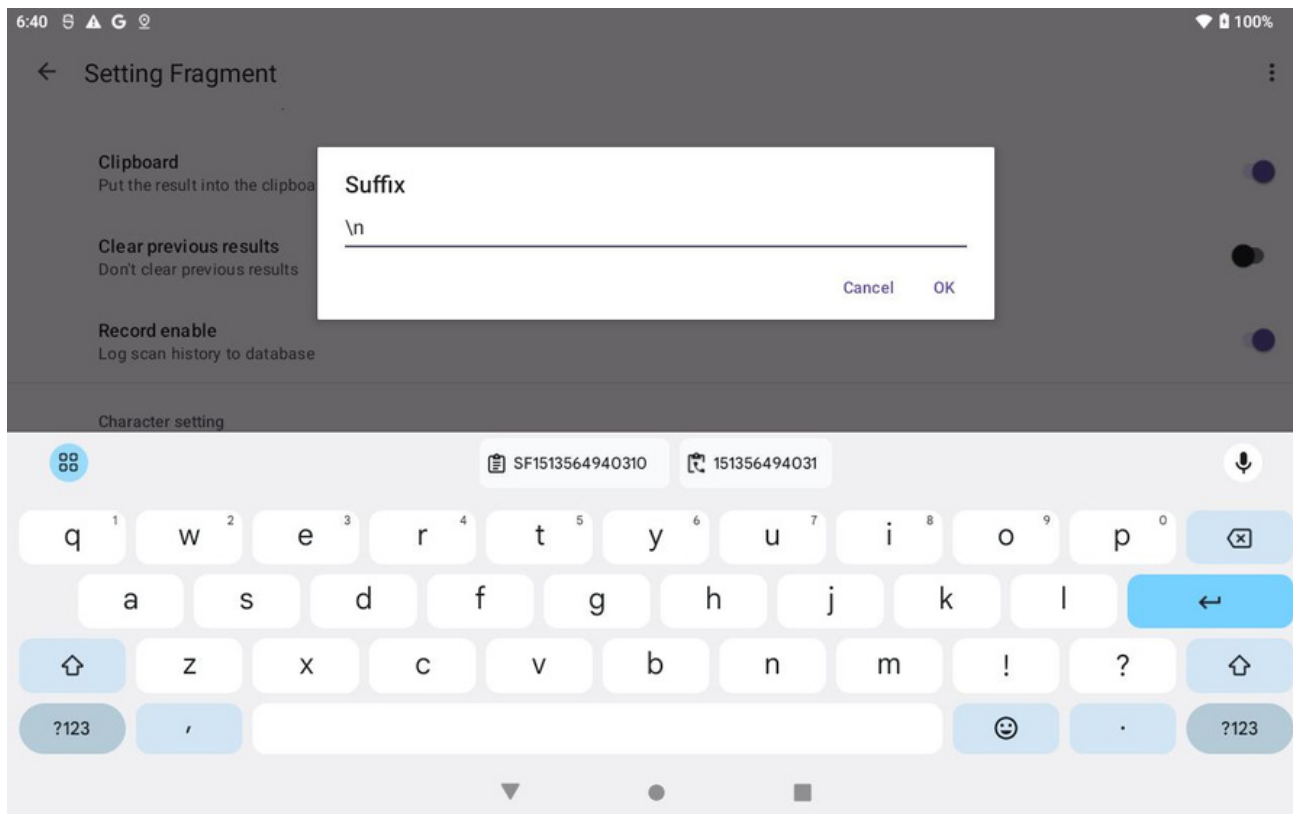
a) The possibility to add “\t” to the suffix (results displayed horizontally)



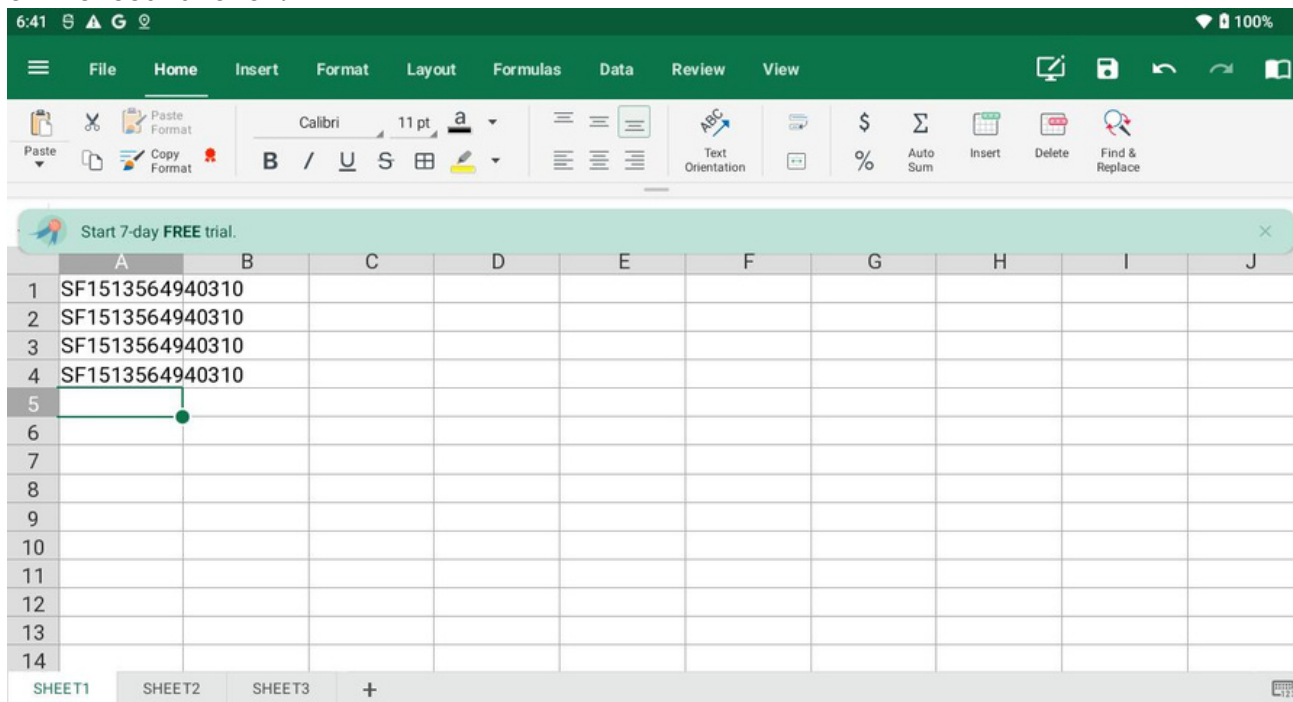
The scanned data will automatically be filled in to the cell next to the current cell.



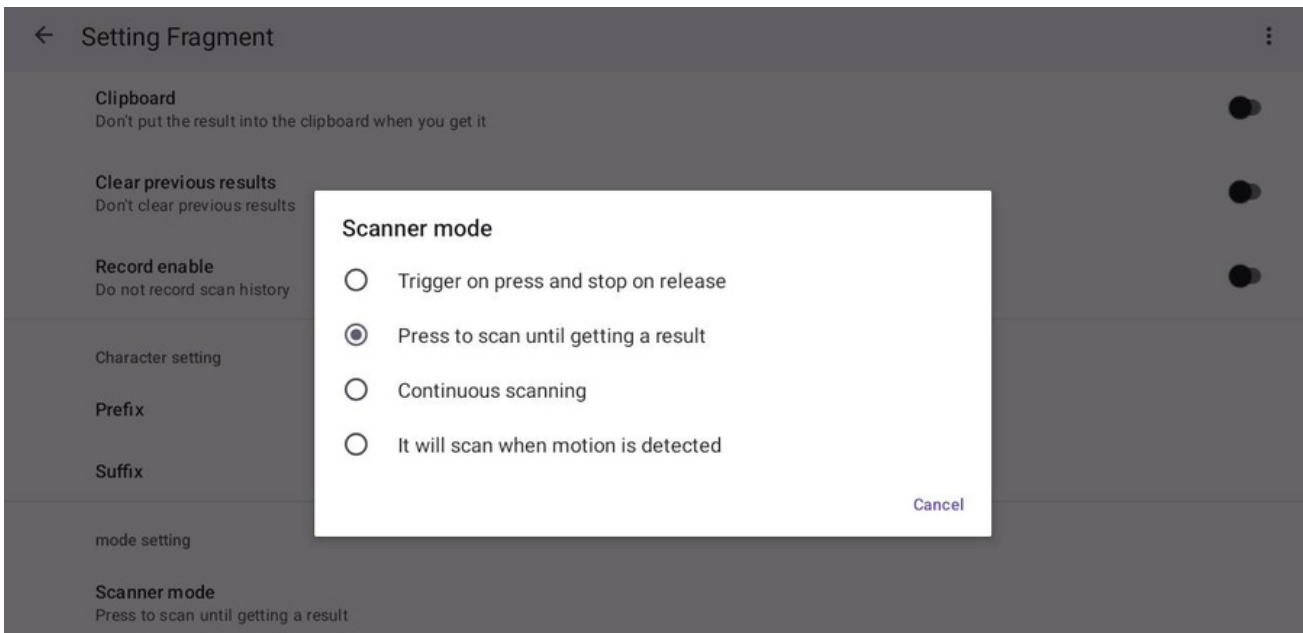
b) The possibility to add “\n” to the Suffix (results displayed vertically)



The scanned data will be filled in to the cell under the current cell through the carriage return or line feed function.



⑩ Scanner mode



a) Trigger on press and stop on release

Press the scan button to start scanning and release it to stop scanning.

b) Press to scan until getting a result

Press the scan button to start scanning, once a Barcode or QR code is scanned, the scanner will be turned off.

c) Continuous scanning

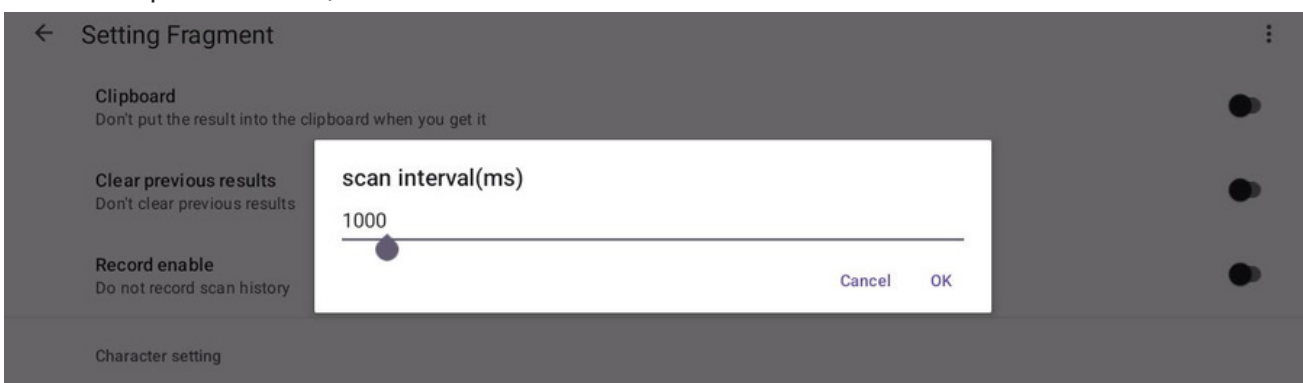
When press the scan button, it will continuously be scanning. If no Barcode or QR code is scanned for more than 10 seconds, the scanner module will be turned off.

d) It will scan when motion is detected

When the scanner module detects a moving object, it will start scanning automatically.

⑪ Scan Interval (ms)

This function can set a scan interval when you've selected the continuous scanning mode (C from the options above).



Demo Application SDK support

We can provide the SDK of the demo app to you, and you can integrate it into your own application.

You can refer to the instructions below to modify your application.

1. Call the following code to bind to ScannerService

```
val serviceIntent = Intent()
serviceIntent.component = ComponentName("hk.paceblade.scannerservice",
"hk.paceblade.scannerservice.ScannerService")
startService(serviceIntent)
bindService(serviceIntent, conn, Activity.BIND_AUTO_CREATE)
```

When connected to the ScannerService, it will enter the onServiceConnected function of the conn callback. The following is the binder variable for obtaining service communication:

```
private val conn: ServiceConnection = object : ServiceConnection {
override fun onServiceConnected(name: ComponentName, service: IBinder) {
//Get the Messenger object associated with the server

Log.d(ScannerFragment.TAG, "onServiceConnected")
binder = IScannerServiceBinder.Stub.asInterface(service)

scannerState = get(R.string.key_enable_state,
getString(R.string.def_can_used) == "true")
                this@MainActivity.toast(if (scannerState) SCANNER_ENABLED_MESSAGE
else SCANNER_DISABLED_MESSAGE)
}

override fun onServiceDisconnected(name: ComponentName) {
Log.d(ScannerFragment.TAG, "onServiceDisconnected")
binder = null
}

}
```

Below is the unbinding method.

```
unbindService(conn)
```

2. Call the following code to communicate with the ScannerService.

```
binder?.putString(String, String)
binder?.putInt(String, Int)
binder?.putBoolean(String, Boolean)
```

```
binder?.putLong(String, Long)
binder?.getString(String, String)
binder?.getInt(String, Int)
binder?.getBoolean(String, Boolean)
binder?.getLong(String, Long)
```

Supported view the parameters:

```
app/src/main/java/hk/paceblade/scannersetup/SettingFragment.kt
```

3.Receive current scan results

```
IntentFilter().apply {
    addAction("hk.paceblade.scannerservice.Barcode") //Scan result
    addAction("hk.paceblade.scannerservice.EnableState") //Scanner if enable
    registerReceiver(barcodeReceiver,this)
}
```

When the broadcast is received, the following callback will be called. The following example get the barcode value and timestamp value, and the scanner enable state.

```
private val barcodeReceiver = object: BroadcastReceiver() {
    override fun onReceive(context: Context, intent: Intent) {
        if (intent.action == "hk.paceblade.scannerservice.Barcode") {
            intent.extras?.apply {
                val barcode = getCharArray("barcode")
                val timestamp = getLong("timestamp").toDateString()
            }
        }else if (intent.action == "hk.paceblade.scannerservice.EnableState"){
            intent.extras?.apply {
                connected.scannerState = getBoolean("state", false)
            }
        }
    }
}
```