PASSTI Library for Newland

User Guide

V1.0.0

Date	Version	Revision
2020-12-07	1.0.0	Initial Release

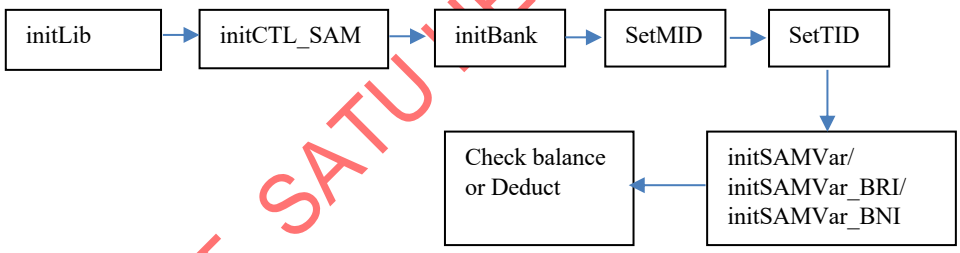
Contents

Introduction

This document is intended to serve developers to integrate STI PASSTI Lib to an Android apps.

1. Process Flow

1. Initialize library : Activate library using provided key. Transaction can only be executed if library has been successfully initialized. This step is done by executing initLib function. This step only need to be executed one time.
2. Power on SAM and activate CTL Reader. Power on targeted SAM slot and contactless card reader. This step is done by executing initCTL_SAM function. This step only need to be executed one time.
3. Initialize supported banks. This step is done by executing initBank function. This step only need to be executed one time.
4. Set Device MID. This step is done by executing setDeviceMID function. This is mandatory to execute transaction. Library will reject transaction if Device MID not set.
5. Set Device TID. This step is done by executing setDeviceTID function. This is mandatory to execute transaction. Library will reject transaction if Device TID not set.
6. Set Banks parameters. Setting corresponding banks parameter. Functions related to this step are : initSAMVar_Mandiri,initSAMVar_BRI,initSAMVar_BNI. Bank parameters must be set every time user want to do transaction. Library will reject transaction if bank's parameters not set.
7. Do transaction : Check Balance or Deduct. Reader will try to detect contactless card for 5 seconds before timeout.



2. Function

1. Package : id.co.softorb.lib.passti.STIUtility

a) initLib

Prototype	int initLib(String clientID)
Feature	Initialize library to activate bank cards transaction functionality. Application must call this function before able to use other functions.
Parameter	ClientID, 16 characters ID provided by PT STI
Return	Statuscode: 1. Success = OK 2. Fail = 2.1. ERR_INVALID_LENGTH: incorrect length of clientID 2.2. ERR_INVALID_KEY: incorrect value of clientID
Comment	

b) initCTL_SAM

Prototype	int initCTL_SAM(int banktype,int samslot);	
Feature	Initialize SAM in samslot	
Parameter	banktype [int]	type: Luminos(1), Mandiri(2),BRI(3),BNI(4), DKI(6).
	Samslot	Sam slot position for card type. Valid value : 1
Return	Statuscode: 1. Success = OK 2. Fail = 2.1. ERR_LIB_NOTINIT : initLib not has not been executed. 2.2. NOT_OK : Exception during bank initialization during bank initialization 2.3. ERR_INCORRECT_SAMSLOT : incorrect samslot valud 2.4. ERR_INCORRECT_CARDTYPE : incorrect card type value 2.5. ERR_SAMNOTREADY : no SAM detected in targeted slot	
Comment	Init SAM bank based on bank type code. Fail to execute this will cause checkbalance and deduct return ERR_CARDTYPEINACTIVE.	

c) initBank

Prototype	int initBank()
Feature	Initialize supported Bank
Parameter	None
Return	Statuscode: 1. Success = OK 2. Fail = 2.1. ERR_LIB_NOTINIT : initLib not has not been executed. 2.2. NOT_OK : Exception during bank initialization
Comment	This **must** be initialize before initCTL_SAM

d) getLibVersion

Prototype	String getLibVersion()
Feature	Get PASSTI library version
Parameter	None
Return	Library version (format xx.yy.zz)
Comment	

e) DeviceType

Prototype	int DeviceType()
Feature	Get library device type registered in PASSTI system
Parameter	None

Return	3 (constant value)
Comment	

f) DeviceLibVersion

Prototype	String DeviceLibVersion()
Feature	Get Newland SDK version
Parameter	None
Return	Newland SDK version
Comment	

g) SetDeviceService

Prototype	void SetDeviceService(ModuleManage modulemanage)
Feature	Set Newland SDK service to library.
Parameter	ModuleManage instance from APOS SDK
Return	none
Comment	This function must be called for library to execute contactless dan SAM command

h) DisconnectService

Prototype	void DisconnectService
Feature	Close device SDK service
Parameter	none
Return	none
Comment	This function must be called when application closed.

i) checkbalance

Prototype	int checkbalance()
Feature	Checkbalance for supported bank card
Parameter	None
Return	Statuscode: 1. Success = OK 2. Fail (check corresponding code in ErrorCode table)= 2.1. ERR_LIB_NOTINIT : initLib not has not been executed. 2.2. ERR_CARDTYPEINACTIVE: card type has not initialized 2.3. CTL_ERR_NOTFOUND : no contactless card detected 2.4. CTL_ERR_READFAIL : contactless reading fail
Comment	Checkbalance for 4 bank card (BNI,BRI,Mandiri,Bank DKI), Note: you can use this function **after** successfully init device and init power. See flow process for reference of init device and init power

j) deduct

Prototype	int deduct(int amount)	
Feature	Do deduct balance from card	
Parameter	Amount[int]	Amount of balance we want to deduct

Return	Statuscode: 1. Success = OK 2. Fail (check corresponding code in ErrorCode table)= 2.1. ERR_LIB_NOTINIT : initLib not has not been executed. 2.2. ERR_CARDTYPEINACTIVE: card type has not initialized 2.3. CTL_ERR_NOTFOUND : no contactless card detected 2.4. CTL_ERR_READFAIL : contactless reading fail 2.5. ERR_PARAM_TID : device TID value is not 4 bytes 2.6. ERR_PARAM_MID: device MID value is not 8 bytes
Comment	Deduct with amount of balance we want to deduct, Note: you can use this function **after** successfully init device and init power. See flow process for reference of init device and init power

k) getBalance

Prototype	String getBalance()
Feature	Get balance value of card
Parameter	None
Return	Balance value
Comment	This function is to get current balance (as balance before in deduct) **after** successfully checkbalance or deduct. When this function called before initLib, return value will be"-13".

l) getLastBalance

Prototype	String getLastBalance()
Feature	Get balance value of card
Parameter	None
Return	String balance of card
Comment	This function is to get current balance **after** successfully deduct. When this function called before initLib, return value will be"-13".

m) getPASSTILog

Prototype	byte[] getPASSTILog()
Feature	Get PASSTI Log transaction
Parameter	None
Return	byte[] (PASSTI Log), -13 means empty log
Comment	This function is to get PASSTI log transaction **after** successfully deduct Composition: OPR Log (40B) + Length Encrypted STI Log + STI Log + Length RFU + RFU + CRC16 (2B) OPR Log: Card Type (1B) + MID (8B) + TID (4B) + Transaction Date (7B) + CardNo (8B) + Amount (4B) + Last Balance (4B) + Transaction Number (4B)

	CardType: 01 : Luminos 02 : Mandiri 03 : BRI 04 : BNI 06 : DKI When this function called before initLib, return value will be -13

n) getCardNo

Prototype	String getCardNo()
Feature	Get card number
Parameter	None
Return	String card number
Comment	This function is to get card number **after** successfully checkbalance or deduct. When this function called before initLib, return value will be"-13".

o) SetMID

Prototype	int SetMID(byte[] MID)	
Feature	Set Device MID	
Parameter	MID [byte]	Device MID. Get it from STI..valid length 8 bytes.
Return	Statuscode: 1. Success = OK 2. Fail (check corresponding code in ErrorCode table)= 2.1. ERR_PARAM_MID : InstitutionID value is not 8 bytes	

p) SetTID

Prototype	int SetTID(byte[] TID)	
Feature	Set Device TID	
Parameter	MID [byte]	Device TID. Get it from STI..valid length 4 bytes.
Return	Statuscode: 3. Success = OK 4. Fail (check corresponding code in ErrorCode table)= 4.1. ERR_PARAM_TID : TID value is not 4 bytes.	

q) initSAMVar_Mandiri

Prototype	String initSAMVar_Mandiri(String PIN, String InsID, String TID);	
Feature	Initialize bank variable	
Parameter	PIN [String]	Mandiri SAM PIN. Get it from Bank, initialize this after initBank.valid length 16 characters.

	InsID [String]	Mandiri Institution ID. Get it from Bank, initialize this after initBank.valid length 4 characters.
	TID [String]	Mandiri Terminal ID. Get it from Bank, initialize this after initBank.valid length 8 characters.
Return	Statuscode: 5. Success = OK 6. Fail (check corresponding code in ErrorCode table)= 6.1. ERR_PARAM_PIN : SAM PIN value not 16 characters 6.2. ERR_PARAM_MID : InstitutionID value is not 4 characters 6.3. ERR_PARAM_TID : TID value is not 8 characters	
Comment	Initialize this after initBank and everytime you want to do deduct a Mandiri Card	

r) initSAMVar_BRI

Prototype	String initSAMVar_BRI(String Procode, String BatchNo, String MID, String TID);	
Feature	Initialize bank variable	
Parameter	Procode [String]	BRI Procode. Get it from Bank, initialize this after initBank.valid length 6 characters.
	BatchNo [String]	BRI Batch Number. Get it from Bank, initialize this after initBank.valid length 2 characters.
	MID [String]	BRI Merchant ID. Get it from Bank, initialize this after initBank.valid length 16 characters.
	TID [String]	BRI Terminal ID. Get it from Bank, initialize this after initBank.valid length 8 characters.
Return	Statuscode: 1. Success = OK 2. Fail (check corresponding code in ErrorCode table)= 2.1. ERR_PARAM_TID : TID value not 8 characters 2.2. ERR_PARAM_PROCODE : Procode value is not 6 characters. 2.3. ERR_PARAM_MID: MID value is not 16 characters 2.4. ERR_PARAM_BATCH : Batch no value not 2 characters.	
Comment	Initialize this after initBank and everytime you want to do deduct a BRI Card	

s) initSAMVar_BNI

Prototype	String initSAMVar_Mandiri(String MID, String TID, String MarriedCode);	
Feature	Initialize bank variable	
Parameter	MID [String]	BNI Merchant ID. Get it from Bank, initialize this after initBank.valid length 16 characters.

	TID [String]	BNI Terminal ID. Get it from Bank, initialize this after initBank.valid length 8 characters.
	MarriedCode [String]	BNI MarriedCode. Get it from Bank, initialize this after initBank.valid length 32 characters.
Return	Statuscode: 1. Success = OK 2. Fail (check corresponding code in ErrorCode table)= 2.1. ERR_PARAM_PIN : Marriage code value not 32 characters 2.2. ERR_PARAM_TID : TID value is not 8 characters 2.3. ERR_PARAM_MID: MID value is not 16 characters	
Comment	Initialize this after initBank and everytime you want to do deduct a BNI Card	

t) checkMarriedBNI

Prototype	String checkMarriedBNI()	
Feature	Check for married status of BNI SAM	
Parameter	Amount[int]	Amount of balance we want to deduct
Return	String SAM Already Married/Married Code: xx/ErrorCode	
Comment	You **have to** save it if SAM not married yet and returned Married Code: xx	

u) SetBRIRefNo

Prototype	void SetBRIRefNo (String refno)
Feature	Set BRI Reference Number used for composing transaction log
Parameter	Reference number
Return	none

Comment	Reference number format must be written in 6 digit of integer.
	Example
	1. 1 = 000001
	2. 10 = 000010
	Code example :
	int brirefno;
	sti = new STIUtility(MainActivity.this);
	String strBRIRef = String.format("%06d",brirefno);
	sti.SetBRIRefNo(strBRIRef);

v) SetTrxCounter

Prototype	void SetTrxCounter (int trxcounter)
Feature	Set Device transaction counter. This counter is used device wide. Transaction counter must be a running number.
Parameter	Transaction counter
Return	none
Comment	Code example : int trxcounter;

	sti = new STIUtility(MainActivity.this);
	trxcounter=1;
	sti.SetTrxCounter(trxcounter);

w) getActionCode

Prototype	int getActionCode ()
Feature	Get library state from last transaction
Parameter	Transaction counter
Return	Action code : 1. NORMAL , code 1: last transaction is normal. 2. REPURCHASE , code 2: last transaction is incomplete, user must complete previous transaction by tapping the same card

Comment	Code example :
	try{
	i = sti.deduct(nominal.getText().toString());
	}
	catch(Exception e)
	{
	Log.e(TAG,"sti.deduct "+e.getMessage());
	i= ErrorCode.NOT_OK;
	}
	int code = sti.getActionCode();
	Log.e(TAG,"sti.getAction,code "+code);
	if (i != OK && code==1) {
	String text="";
	final String txt2display = text+"Deduct fail, Need Correction";
	runOnUiThread(new Runnable() {
	@Override
	public void run() {
	tvTeks.setText(txt2display);
	progressBar.dismiss();
	Toast.makeText(TestActivity.this, txt2display, Toast.LENGTH_SHORT).show();
	}
	});
	}
	}

x) CancelRepurchase

Prototype	void CancelRepurchase ()
Feature	To reset library to normal state if previous transaction cause reader in REPURCHASE mode

Parameter	none
Return	none

Comment	Code example :
	try{
	
	i = sti.deduct(nominal.getText().toString());
	}
	catch(Exception e)
	{
	Log.e(TAG,"sti.deduct "+e.getMessage());
	i= ErrorCode.NOT_OK;
	}
	
	int code = sti.getActionCode();
	
	Log.e(TAG,"sti.getAction,code "+code);
	if (i != OK && code==1) {
	String text="";
	sti.CancelRepurchase();
	
	}

2. Response Code

Code	Desc
0	OK
24	ERR_WRONGCARDNO
-1203	CTL_ERR_NOTFOUND
-1204	CTL_ERR_READFAIL
-109	ERR_INVALID_LENGTH
-108	ERR_INVALID_KEY
-33	ERR_PARAM_PROCODE
-32	ERR_PARAM_BATCH
-31	ERR_PARAM_TID
-30	ERR_PARAM_MID
-29	ERR_PARAM_PIN
-13	ERR_LIB_NOTINIT
-12	ERR_INCORRECT_CARDTYPE
-11	ERR_INCORRECT_SAMSLOT
-2	ERR_SW1SW2
-1	ERR_TIMEOUT
2	NO_TAG_ERROR
3	ERR_NO_RESP
77	ERR_CARDTYPEINACTIVE
-109	CTL_ERR_TIMEWINDOW
-608	DKI_CTL_ERR_SELECTAPP

-609	DKI_SAM_ERR_SELECTAID
-610	DKI_CTL_ERR_INITTOPUP
-611	DKI_SAM_ERR_DEBITLSAM
-612	DKI_CTL_ERR_TOPUP
-613	DKI_SAM_ERR_CONFIRMLSAM
-620	DKI_CTL_ERR_INITPURCHASE
-621	DKI_SAM_ERR_INITPSAM
-622	DKI_CTL_ERR_DEBIT
-623	DKI_SAM_ERR_CREDITPSAM
-630	DKI_CTL_ERR_INITREPURCHASE
-631	DKI_SAM_ERR_INITREPURCHASE
-632	DKI_CTL_ERR_REPURCHASE
-633	DKI_SAM_ERR_REPURCHASE
-516	CTL_ERR_INSUFFICIENTBALANCE
-517	CTL_ERR_TOPUPEXCEEDMAXBALANCE
-104	LUMINOS_CTL_ERR_GETTRXLOG
-110	LUMINOS_CTL_ERR_INITTOPUP
-111	LUMINOS_SAM_ERR_DEBITLSAM
-112	LUMINOS_CTL_ERR_TOPUP
-113	LUMINOS_SAM_ERR_CONFIRMLSAM
-120	LUMINOS_CTL_ERR_INITPURCHASE
-121	LUMINOS_SAM_ERR_INITPSAM
-122	LUMINOS_CTL_ERR_DEBIT
-123	LUMINOS_SAM_ERR_CREDITPSAM
-124	LUMINOS_SAM_ERR_SELECTAID
-130	LUMINOS_CTL_ERR_INITREPURCHASE
-131	LUMINOS_SAM_ERR_INITREPURCHASE
-131	LUMINOS_CTL_ERR_REPURCHASE
-133	LUMINOS_SAM_ERR_REPURCHASE
-211	MDR_CTL_ERR_READDATA
-212	MDR_CTL_ERR_READCARDINFO
-213	MDR_CTL_ERR_GETBALANCE
-214	MDR_SAM_ERR_INITPURCHASE
-215	MDR_CTL_ERR_DEBITPURSE
-216	MDR_SAM_ERR_GETLOG
-217	MDR_SAM_ERR_SELECT
-218	MDR_SAM_ERR_SELECTAID1
-219	MDR_SAM_ERR_LOGIN1
-220	MDR_SAM_ERR_UPDATETERMINALINFO
-221	MDR_SAM_ERR_SELECTAID2
-222	MDR_SAM_ERR_GETUID
-223	MDR_SAM_ERR_GETINFO
-224	MDR_SAM_ERR_GETJCOPBM
-225	MDR_SAM_ERR_STARTPAYMENT
-226	MDR_SAM_ERR_GETTRXREPORT
-256	MDR_CTL_GRACEPERIOD
-310	BRI_SAM_ERR_SELECTAID

-311	BRI_CTL_ERR_DESFIRESELECTAID1
-312	BRI_CTL_ERR_DESFIREGETCARDNO
-313	BRI_CTL_ERR_DESFIREGETCARDSTATUS
-314	BRI_CTL_ERR_DESFIRESELECTAID3
-315	BRI_CTL_ERR_DESFIREGETKEYCARD
-316	BRI_SAM_ERR_DESFIREAUTHKEY
-317	BRI_CTL_ERR_DESFIREAUTHCARD
-318	BRI_CTL_ERR_DESFIREGETLASTTRXDATE
-319	BRI_CTL_ERR_DESFIREGETBALANCE
-320	BRI_CTL_ERR_DESFIREDEBIT
-321	BRI_SAM_ERR_DESFIRECREATEHASH
-322	BRI_CTL_ERR_DESFIREWRITELOG
-323	BRI_CTL_ERR_DESFIREWRITELASTTRX
-324	BRI_CTL_ERR_DESFIRECOMMITTRX
-325	BRI_CTL_ERR_DESFIREABORTTRX
-326	BRI_CTL_ERR_GETPARTNERDATA
-327	BRI_CTL_ERR_WRITEPARTNERDATA
-328	BRI_CTL_ERR_AUTHPARTNERDATA
-329	BRI_SAM_ERR_GETRANDOM
-330	BRI_CTL_ERR_SELECTAID6
-331	BRI_CTL_ERR_GETKEYCARD8
-332	BRI_CTL_ERR_GETKEYCARD2
-333	BRI_CTL_ERR_SELECTAID5
-334	BRI_CTL_ERR_APPNOTACTIVE
-335	BRI_CTL_ERR_SELECTAID
-336	BRI_SAM_ERR_REINIT
-410	BNI_CTL_ERR_SELECTAPP
-411	BNI_CTL_ERR_GETCHALLENGE
-412	BNI_CTL_ERR_GETPURSEDATA
-413	BNI_SAM_ERR_VERIFYSECUREREADPURSE
-414	BNI_SAM_ERR_GENERATEDEBIT
-415	BNI_CTL_ERR_DEBITPURSE
-416	BNI_SAM_ERR_VERIFYDEBITRECEIPT
-417	BNI_SAM_ERR_SELECTAID
-481	BNI_SAM_ERR_GETINFO
-419	BNI_SAM_ERR_NOTMARRIED
-420	BNI_SAM_ERR_INIT
-421	BNI_CTL_ERR_PURSENOTENABLED

3. Sample code

id.co.softorb.sunmi.STIUtility

```
protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.activity_test);
        getWindow().addFlags(WindowManager.LayoutParams.FLAG_FULLSCREEN);
        // requestWindowFeature(Window.FEATURE_NO_TITLE);
        getWindow().addFlags(WindowManager.LayoutParams.FLAG_KEEP_SCREEN_ON);
```

```
        initpower = findViewById(R.id.bInitPower);

        checkbalance = findViewById(R.id.bCheckBalance);
        deduct = findViewById(R.id.bDeduct);
        nominal = findViewById(R.id.etPaymentDeduct);
        tvTeks = findViewById(R.id.tvTitle);
        initdevice = findViewById(R.id.bInitDevice);
        samtype = findViewById(R.id.etSAM);
        passtiversion = findViewById(R.id.valuePasstiVersion);
        aposversion = findViewById(R.id.valueDeviceLibVersion);
        samslot = findViewById(R.id.etSAMSlot);
        initLib = findViewById(R.id.bInitLib);
        devicetype=findViewById(R.id.valueDeviceType);
        login = findViewById(R.id.bLogin);
        post = findViewById(R.id.bPost);
        clear = findViewById(R.id.bClearScreen);
        trxcounter=0;
        brirefno=0;

/**********************Important**********************/
        clear.setOnClickListener(clickclear);

        deduct.setOnClickListener(clickdeduct);

        checkbalance.setOnClickListener(clickcheckbalance);

        initdevice.setOnClickListener(clickinitdevice);

        initLib.setOnClickListener(clickinitlib);

        initpower.setOnClickListener(clickinitpower);

        post.setOnClickListener(clickpostdata);

        login.setOnClickListener(clicklogin);

    }
    View.OnClickListener clicklogin=new View.OnClickListener() {
        @Override
        public void onClick(View v) {
            if(passti == null){
                showToast("Init Lib dahulu");
                return;
            }
            byte[] username = "Admin".getBytes();
            byte[] password = "Admin".getBytes();
            byte[] composeForLogin = passti.pcd_cmd_login(username,password);
            Log.d(TAG,"ComposeforLogin: " +
BytesUtil.bytes2HexString(composeForLogin));
            tvTeks.setText("ComposedLogin: " +
BytesUtil.bytes2HexString(composeForLogin));
        }
    };
    View.OnClickListener clickpostdata=new View.OnClickListener() {
        @Override
        public void onClick(View v) {
            if(passti == null || sti == null){
                showToast("Init Lib dahulu");
                return;
```

```
            }
            if(sti.getPASSTILog() == null){
                showToast("Deduct dahulu");
                return;
            }
            byte[] composeupload =
passti.pcd_cmd_dataupload(BytesUtil.hexString2Bytes("6361616531323737316537353439336161653966626663386563396162656265"),passti.MID,passti.TID,sti.getPASSTILog());
            Log.d(TAG,"ComposeUpload: " +
BytesUtil.bytes2HexString(composeupload));
            tvTeks.setText("ComposedUpload: " +
BytesUtil.bytes2HexString(composeupload));

        }
    };
    View.OnClickListener clickclear=new View.OnClickListener() {
        @Override
        public void onClick(View v) {
            if(tvTeks!=null)
            {
                tvTeks.setText("");

            }}};
    View.OnClickListener clickdeduct=new View.OnClickListener() {
        @Override
        public void onClick(View v) {
            int res;
            if (nominal.getText().toString() == null ||
nominal.getText().toString().matches("")) {
                tvTeks.setText("Fill amount coloumn first!");
                nominal.setError("Can't be blank/empty");
                return;
            }


            initialize_bankparam();
            ShowProgressBar();
            new Thread(new Runnable() {
                @Override
                public void run() {
                    if(sti == null){
                        progressBar.dismiss();
                        showToast("Init Lib dahulu");
                        return;
                    }
                    if(!sti.checkInitLib()){
                        progressBar.dismiss();
                        showToast("Init Lib dahulu");
                        return;
                    }
                    int i = 0;

                    try{
                        trxcounter++;
                        brirefno++;
                        sti.SetTrxCounter(trxcounter);
                        String strBRIRef = String.format("%06d",brirefno);
                        Log.d(TAG,"strBRIRef "+strBRIRef);
                        sti.SetBRIRefNo(strBRIRef);
```

```
                        i = sti.deduct(nominal.getText().toString());
                    }
                    catch(Exception e)
                    {
                        Log.e(TAG,"sti.deduct "+e.getMessage());
                        i= NOT_OK;
                    }
                    //if (i != OK && sti.getMandiriCorrection() ||
sti.getBRICorrection() || sti.getBNICorrection()) {
                    int code = sti.getActionCode();
                    //sti.CancelRepurchase();

                    Log.e(TAG,"sti.getAction,code "+code);
                    if (i != OK && code==1) {
                        String text="";


                        final String txt2display = text+"Deduct fail, Need
Correction";

                        runOnUiThread(new Runnable() {
                            @Override
                            public void run() {
                                tvTeks.setText(txt2display);
                                progressBar.dismiss();
                                Toast.makeText(TestActivity.this, txt2display,
Toast.LENGTH_SHORT).show();
                            }
                        });

                    }else if (i != OK && i == -102) {
                        runOnUiThread(new Runnable() {
                            @Override
                            public void run() {
                                tvTeks.setText("Gagal Deduct - Saldo
Kurang\nSaldo: " + sti.getBalance() + "\nPotongan Deduct: " +
nominal.getText().toString());
                                progressBar.dismiss();
                                Toast.makeText(TestActivity.this, "Gagal Deduct",
Toast.LENGTH_SHORT).show();
                            }
                        });
                    }
                    else if (i != OK) {
                        int finalI = i;
                        runOnUiThread(new Runnable() {
                            @Override
                            public void run() {
                                tvTeks.setText("Gagal Deduct\nResponseCode: " +
finalI);
                                progressBar.dismiss();
                                Toast.makeText(TestActivity.this, "Gagal Deduct",
Toast.LENGTH_SHORT).show();
                            }
                        });

                    } else {

                        String pesan = "Saldo Awal: " + sti.getBalance() +
```

```
"\nSaldo Akhir: " + sti.getLastBalance() + "\nCardNum: " + sti.getCardNo() +
"\nPASSTILog: " + BytesUtil.bytes2HexString(sti.getPASSTILog());
                        runOnUiThread(new Runnable() {
                            @Override
                            public void run() {
                                Log.d(TAG,"Messages: " + pesan);

//printing(sti.getBalance(),sti.getLastBalance(),sti.getCardNo());
                                tvTeks.setText(pesan);
                                progressBar.dismiss();
                                Toast.makeText(TestActivity.this, "Sukses Deduct",
Toast.LENGTH_SHORT).show();
                            }
                        });
                    }
                }
            }).start();
        }
    };
    View.OnClickListener clickcheckbalance=new View.OnClickListener() {
        @Override
        public void onClick(View v) {
            initialize_bankparam();
            ShowProgressBar();

            new Thread(new Runnable() {
                @Override
                public void run() {
                    try {
                        if(sti == null){
                            progressBar.dismiss();
                            showToast("Init Lib dahulu");
                            return;
                        }
                        if(!sti.checkInitLib()){
                            progressBar.dismiss();
                            showToast("Init Lib dahulu");
                            return;
                        }
                        int i = sti.checkbalance();

                        if (i != OK) {
                            runOnUiThread(new Runnable() {
                                @Override
                                public void run() {
                                    tvTeks.setText("Gagal Check Balance\nResponse
Code: " + i);

                                    if (progressBar != null)
                                        progressBar.dismiss();

                                    Toast.makeText(TestActivity.this, "Gagal Check
Balance", Toast.LENGTH_SHORT).show();
                                }
                            });

                        } else {

                            runOnUiThread(new Runnable() {
                                @Override
```

```
                                public void run() {
                                    if (progressBar != null)
progressBar.dismiss();
                                    tvTeks.setText("Saldo: " + sti.getBalance() +
"\nCardNo: " + sti.getCardNo());

                                    Toast.makeText(TestActivity.this, "Sukses
Check Balance", Toast.LENGTH_SHORT).show();
                                }
                            });

                        }


                    } catch (Exception e) {
                        e.printStackTrace();
                        runOnUiThread(new Runnable() {
                            @Override
                            public void run() {
                                if (progressBar != null) progressBar.dismiss();
                                Toast.makeText(TestActivity.this, "Exception :
"+e.getMessage(), Toast.LENGTH_SHORT).show();
                            }
                        });
                    }
                }
            }).start();
        }
    };
    View.OnClickListener clickinitpower= new View.OnClickListener() {
        @Override
        public void onClick(View v) {
            //
            //type*
            //Luminos: 1
            //Mandiri: 2
            //BRI: 3
            //BNI: 4
            //for APOS with only 1 slot, fill with: 1
            //
            if(sti == null){
                showToast("Init Device dahulu");
                return;
            }
            if(!sti.checkInitLib()){
                showToast("Init Library dahulu");
                return;
            }
            if (samtype.getText().toString() == null ||
samtype.getText().toString().matches("")) {
                samtype.setError("Can't be blank/empty");
                return;
            }
            if (samslot.getText().toString() == null ||
samslot.getText().toString().matches("")) {
                samslot.setError("Can't be blank/empty");
                return;
            }
            String message = "";
```

```
            int samBankType = Integer.valueOf(samtype.getText().toString());
            int samSlot = Integer.valueOf(samslot.getText().toString());
            int iRet;

            iRet = sti.initCTL_SAM(samBankType,samSlot);
            switch(samBankType)
            {
                case 1 : message="Luminos : ";break;
                case 2 : message="Mandiri : ";break;
                case 3 : message="BRI : ";break;
                case 4 : message="BNI : ";break;
                default : message="Unknown Issuer : ";break;
            }
            message+= GetCodeDesc(iRet);

            tvTeks.setText(message);
        }
    };
    View.OnClickListener clickinitlib= new View.OnClickListener() {
        @Override
        public void onClick(View v) {
            try{

                if(sti == null){
                    showToast("Init Device dahulu");
                    return;
                }
                int res = sti.initLib("758F40D46D95D1641448AA19B9282C05");
                //int res = sti.initLib("4B2AA2F02AEF40C7A4F8EE425BFF4555");

                if(res==0) {
                    sti.initBank();

                    code=sti.SetMID(deviceMID);
                    if(code!=OK)
                    {
                        tvTeks.setText("SetMID fail, code "+code);
                        return;
                    }
                    code=sti.SetTID(deviceTID);
                    if(code!=OK)
                    {
                        tvTeks.setText("SetTID fail, code "+code);
                        return;
                    }

                }
                tvTeks.setText("Init " + String.valueOf("STI") + "\nResponseCode:
" + res);
            }catch (Exception e){
                e.printStackTrace();
            }
        }
    };

 View.OnClickListener clickinitdevice=new View.OnClickListener() {
        @Override
        public void onClick(View v) {
```

```
            String teks;
            try {
                if(sti==null) {
                    sti = new STIUtility(getApplicationContext());
                    passtiversion.setText(sti.getLibVersion());
                    aposversion.setText(sti.DeviceLibVersion());
                    devicetype.setText(Integer.toString(sti.DeviceType()));
                }
                moduleManage = ModuleManage.getInstance();


                sti.SetDeviceService(moduleManage);
                teks = "Serial #: " +
moduleManage.getSettingsModule().getInfo(InfoItem.SERIAL_NUMBER) + "\n"+
"OSVersion: " + moduleManage.getSettingsModule().getInfo(InfoItem.FIRMWARE) + "\n"
+ "IMEI: " + moduleManage.getSettingsModule().getInfo(InfoItem.IMEI) + "\n";
                Log.d(TAG, teks);
                tvTeks.setText(teks);
            } catch (Exception e) {
                e.printStackTrace();
            }
        }
    };


@Override
protected void onDestroy() {
    Log.d(TAG,".onDestroy");
    super.onDestroy();

    try {

        sti.DisconnectService();
        System.exit(0);
    } catch (Exception e) {
        e.printStackTrace();
    }
}
int initialize_bankparam()
{
    int code;
    int res;


res=sti.initSAMVar_Mandiri(BytesUtil.bytes2HexString(PIN),BytesUtil.bytes2HexStrin
g(ins_id),BytesUtil.bytes2HexString(tid));//PIN,insID,TID
    if(res!=OK)
    {
        tvTeks.setText("initSAMVar_Mandiri, code "+res);
        return -2;
    }



res=sti.initSAMVar_BNI(dummy_BNI_MID,dummy_BNI_TID,bnisam7050000000000210_mrgcode)
;//MID,TID,MarriedCode

    if(res!=OK)
    {
```

```
        tvTeks.setText("initSAMVar_BNI, code "+res);
        return -4;
    }


res=sti.initSAMVar_BRI(dummy_BRI_Procode,dummy_BRI_Batch,dummy_BRI_MID,dummy_BRI_TID);//procode,batchno,merchantID,TID
    if(res!=OK)
    {
        tvTeks.setText("initSAMVar_BRI, code "+res);
        return -3;
    }
    tvTeks.setText("Bank params init successful");
    return OK;
}
```