Card Dispenser F3 API Instruction

This file include F3-1300 reader's API instruction.

Files list:

F3API.h
F3API.lib
F3API.dll

API Reference:
Basic Operate:

F3_Connect
A6_Connect Build a connection between Program and reader
LONG WINAPI F3_Connect
(
IN DWORD dwPort,
IN DWORD dwSpeed,
IN BYTE bCRAddr,
OUT LPREADERHANDLE lphReader
);
Parameter:
dwPort COM Port number, Available value: 1 ~ 256.
DwSpeed Baud rate, available value:
9600
19200
38400
57600
BCRAddr Machine address, value at 00~15
phReader Return a mark with card reader connectoin's handle

Return Value:
Return 0 is success, other number are error.

F3_Disconnet Disconnect program and reader's communication
LONG
WINAPI
F3_Disconnect
(IN READERHANDLE hReader);

Parameter:
hReader Reference F3_Connect return handle value

Return Value:
Return 0 is success, other number are error.

F3_Initialize Reset card reader

```
LONG
WINAPI
F3_Initialize
(
  IN READERHANDLE hReader,
IN BYTE bMode,
IN BOOL fEnableCounter,
OUT PSTR pszRevBuff,
IN OUT PDWORD pcbRevLength
  );
```

Parameter:

hReader Quote F3_Connect return handle value.
bMode Reset model, available value:
INIT_RETURN_TO_FRONT Reset and move card to exit port
INIT_CAPTURE_TO_BOX Reset and reclaim card
INIT_WITHOUT_Movement Reset but card no action

fEnableCounter on/off the collect card counter function
pszRevBuff return firmware version info
pcbRevLength provide “pbVerBuff” parameter data length (byte number), and receive reader return data.

Return Value:
Return 0 is success, other number are error.

F3_GetCRStatus To get dispenser status

```
LONG WINAPI
F3_GetCRStatus
(
IN READERHANDLE hReader,
OUT PCRSTATUS lpStatus
);
```

Parameter:
hReader Quote F3_Connect return handle value.
lpStatus Point to return status.

Return Value:
Return 0 is success, other number are error.

F3_GetSenseDetail To get sensor status info

```
LONG
WINAPI
F3_GetSenserDetail
(
IN READERHANDLE hReader,
OUT BYTE (&bStatus)[NUM_SENSORS]
);
```

Parameter:

hReader Quote F3_Connect return handle value.
bStatus Return sensor status, it's from S1~S10、KS1、KS2, Value 0x31, mean detected card; 0x30, means no card.

Return Value:
Return 0 is success, other number are error.

F3_MoveCard Move card

```
LONG
WINAPI
F3_MoveCard
(
  IN READERHANDLE hReader,
  IN BYTE bMode
  );
```

Parameter:

hReader Quote F3_Connect return handle value.
bMode Move mode, valid value:
MM_RETURN_TO_FRONT Move card to front port and hold on
MM_RETURN_TO_IC_POS Move card to reader IC position
MM_RETURN_TO_RF_POS Move card to RFID reading position
MM_CAPTURE_TO_BOX Reclaim card
MM_EJECT_TO_FRONT Move card to front port and drop out

Return Value:
Return 0 is success, other number are error.

F3_PermitInsertion Enable card entry from front port

```
LONG
WINAPI
F3_PermitInsertion
(
  __in READERHANDLE hReader
);
```

Paramter

hReader Quote F3_Connect return handle value.

Return Value:
Return 0 is success, other number are error.

F3_DenieInsertion Disable card entry from front port

```
LONG
WINAPI
A6_DenieInsertion
(
_in READERHANDLE hReader
);
```

Parameter:
hReader Quote F3_Connect return handle value.

Return Value:
Return 0 is success, other number are error.

F3_DetectIccType Analyse IC card type

```
LONG
WINAPI
F3_DetectIccType
(
IN READERHANDLE hReader,
OUT PBYTE pbCardType
);
```

Parameter:

hReader Quote F3_Connect return handle value.
pbCardType Return IC card type, value may as :

ICCTYPE_UNKNOWN
ICCTYPE_TO_CPU
ICCTYPE_T1_CPU
ICCTYPE_SLE4442
ICCTYPE_SLE4428
ICCTYPE_AT24C01
ICCTYPE_AT24C02
ICCTYPE_AT24C04
ICCTYPE_AT24C08
ICCTYPE_AT24C16
ICCTYPE_AT24C32
ICCTYPE_AT24C64
ICCTYPE_AT24C128
ICCTYPE_AT24C256

Return Value:
Return 0 is success, other number are error.

F3_DetectRfcType Detect RFID card type

```
LONG
WINAPI
F3_DetectRfcType
(
IN READERHANDLE hReader,
OUT PBYTE pbCardType
  );
```

Parameter:

hReader Quote F3_Connect return handle value.
pbCardType Return RFID card type may as:

RFCTYPE_UNKNOWN
RFCTYPE_MIFARE_S50
RFCTYPE_MIFARE_S70
RFCTYPE_MIFARE_UL
RFCTYPE_TYPEA_CPU
RFCTYPE_TYPEB_CPU

Return Value:
Return 0 is success, other number are error.

Contact CPU Card Operate

F3_CpuActivate CPU card activate (cold reset)

```
LONG
WINAPI
F3_CpuActivate
(
IN READERHANDLE hReader,
OUT PBYTE pbProtocol,
OUT PBYTE pbATRBuff,
IN OUT PDWORD pcbATRLength,
IN OPTIONAL BYTE bVCC = VCC_5V_EMV
);
```

Parameter:

hReader Quote F3_Connect return handle value.

pbProtocol Return CPU card type, value may as:

ICC_PROTOCOL_T0

ICC_PROTOCOL_T1

pbATRBuff Point to return reset info. If not NULL, pcbATRLength can not be NULL.

pcbATRLength Provide pbATRBuff parameter length (byte numbers) and receive reader return's length.

bVCC Activate IC card voltage, valid value:

VCC_5V_EMV Use 5V and EMV standard

VCC_5V_ISO7816 Use 5V and ISO7816 standard

VCC_3V_ISO7816 Use 3V and ISO7816 standard

Return Value:

Return 0 is success, other number are error.

F3_CpuDeactivate CPU card dis activate

```
LONG
WINAPI
F3_CpuDeactivate
(
  IN READERHANDLE hReader
);
```

Parameter:

hReader Quote F3_Connect return handle value.

Return Value:

Return 0 is success, other number are error.

F3_CpuGetStatus Get CUP card status

```
LONG
```

```
WINAPI
F3_CpuGetStatus
(
IN READERHANDLE hReader,
OUT PBYTE pbStatus
);
```

Parameter:
hReader Quote F3_Connect return handle value.
pbStatus Return CPU card status, value may as:
STATUS_DEACTIVATION CPU card haven't activated
STATUS_CLKFREQ_3_57 CPU card have activated, working frequency at 3.57 MHz
STATUS_CLKFREQ_7_16 CPU have activated, working frequency at 7.16 MHz

Return Value:
Return 0 is success, other number are error.

F3_CpuWarmReset CPU card warm reset

```
LONG
WINAPI
F3_CpuWarmReset
(
IN READERHANDLE hReader,
OUT PBYTE pbProtocol,
OUT PBYTE pbATRBuff,
IN OUT PDWORD pcbATRLength
);
```

Parameter:
hReader Quote F3_Connect return handle value.
pbProtocol Return CPU card protocol type, value may as:
ICC_PROTOCOL_T0
ICC_PROTOCOL_T1
pbATRBuff Point to return reset info. If not NULL, pcbATRLength can not be NULL.
pcbATRLength Provide pbATRBuff parameter length (byte numbers) and receive reader return's length.

Return Value:
Return 0 is success, other number are error.

F3_CpuTransmit CPU Card data transmit

```
LONG
```

```
WINAPI
F3_CpuTransmit
(
IN READERHANDLE hReader,
IN BYTE bProtocol,
IN PBYTE pbSendBuff,
IN USHORT cbSendLength,
OUT PBYTE pbRecvBuff,
IN OUT PDWORD pcbRecvLength
);
```

Parameter:
hReader Quote F3_Connect return handle value.
bProtocol CPU card communicate protocol type, value as:
ICC_PROTOCOL_T0 T = 0 Protocol
ICC_PROTOCOL_T1 T = 1 Protocol
ICC_PROTOCOL_AUTO Automatic select T=0 Or T=1 Protocol
pbSendBuff Point to data which was written. Can not be "NULL"
cbSendLength Provide pbSendBuff parameter length. (bytes)
pbRecvBuff Point to return data. Can not be "NULL"
pcbRecvLength Provide pbRecvBuff parameter length. (bytes), and receive the real return data length, Can not be "NULL"

Return value:
Return 0 is success, other number are error.

SAM Card Operate Functions:

F3_SamActivate SAM Activate(cold reset)

```
LONG
WINAPI
F3_SamActivate
(
IN READERHANDLE hReader,
OUT PBYTE pbProtocol,
OUT PBYTE pbATRBuff,
IN OUT PDWORD pcbATRLength,
IN OPTIONAL BYTE bVCC = VCC_5V_EMV
);
```

Parameter:
hReader Quote F3_Connect return handle value.
pbProtocol Return SAM card protocol type, value as:
ICC_PROTOCOL_T0
ICC_PROTOCOL_T1
pbATRBuff Point to return reset info. If not NULL, pcbATRLength also can not be NULL.
pcbATRLength Provide pbATRBuff parameter length, and receive the real return data length

bVCC Aticate card voltage, value as:

VCC_5V_EMV	5V with EMV standard
VCC_5V_ISO7816	5V with ISO7816 standard
VCC_3V_ISO7816	3V with ISO7816 stadnard

Return value:
Return 0 is success, other number are error.

F3_SamDeactivate SAM card release(Deactivate)

```
LONG
WINAPI
F3_SamDeactivate
(
IN READERHANDLE hReader
  );
```

Parameter:
hReader Quote F3_Connect return handle value.
Return value:
Return 0 is success, other number are error.

F3_SamGetStatus , Check SAM card status

```
LONG
WINAPI
F3_SamGetStatus
(
IN READERHANDLE hReader,
OUT PBYTE pbStatus,
OUT PBYTE pbSAMNumber
);
```

Parameter:
hReader Quote F3_Connect return handle value.
pbStatus Return SAM card status, value as:
STATUS_DEACTIVATION CPU Card haven't activated
STATUS_CLKFREQ_3_57 CPU card have activated, working frequency at 3.57 MHz
STATUS_CLKFREQ_7_16 CPU card have activated, working frequency at 7.16 MHz
pbSAMNumber Return present SAM card number, value may be: 1, 2, 3, …
Return value:
Return 0 is success, other number are error code.

F3_SamWarmReset SAM card warm reset.

```
LONG
WINAPI
F3_SamWarmReset
(
IN READERHANDLE hReader,
OUT PBYTE pbProtocol,
OUT PBYTE pbATRBuff,
```

IN OUT PDWORD pcbATRLength
);
Parameter:
hReader Quote F3_Connect return handle value.
bSAMNumber SAM card number, available value: 1，2，3，…

Return value:
Return 0 is success, other number are error code.

SLE4442 Card Operation function:

F3_Sle4442 Activate Activate SLE4442 card
LONG
WINAPI
F3_Sle4442Activate
(
IN READERHANDLE hReader,
OUT PBYTE pbATRBuff,
IN OUT PDWORD pcbATRLength
);
Parameter:

hReader Quote F3_Connect return handle value
pbATRBuff Point to the returned reset info. If it is not NULL, bpbATRLength can also not NULL.
pcbATRLength Provide pbATRBuff parameter length (byte numbers) and receive the real return length from reader.

Return value:
Return 0 is success, other number are error code.

F3_Sle4442Deactivate

Release SLE4442 card:

LONG
WINAPI
F3_Sle4442Deactivate
(
IN READERHANDLE hReader
);

Parameter:
hReader Quote F3_Connect return handle value
Return value:
Return 0 is success, other number are error code.

F3_Sle4442GetStatus

Get SLE4442 card status:
LONG
WINAPI
F3_Sle4442GetStatus
(
  IN READERHANDLE hReader,
OUT PBOOL pfActivated
);

Parameter:
hReader Quote F3_Connect return handle value
pfActivated retrun value is TRUE, means card have been activated, return value is FALSE, means card didn't activated.

Return value:
Return 0 is success, other number are error code.

F3_Sle4442ReadMainMemory
Read the main storage memory:
LONG
WINAPI
F3_Sle4442ReadMainMemory
(
IN READERHANDLE hReader,
IN BYTE bStartAddress,
IN BYTE bBytesToRead,
OUT PBYTE pbBuffer,
IN OUT PDWORD pcbLength
);

Parameter:
HReader Quote F3_Connect return handle value
BstartAddress Address want to operate
bBytesToRead Byte numbers want to read
pcbLength Provide pbBuffer parameter length (byte numbers) and receive the real return length from reader.

Return value:
Return 0 is success, other number are error code.

F3_Sle4442UpdateMainMemory

Update main memory:
LONG
WINAPI
F3_Sle4442UpdateMainMemory
(
IN READERHANDLE hReader,
IN BYTE bStartAddress,
IN BYTE nBytesToWrite,
IN PBYTE pbBuffer
);

Parameter:

hReader Quote F3_Connect return handle value
bStartAddress Address want to operate
bBytesToWrite Byte numbers want to write
pbBuffer Point to the data which want to write to a card. Can not be NULL.

Return value:
Return 0 is success, other number are error code.

F3_Sle4442ReadProtectionMemory

Read protected memory:

LONG
WINAPI
F3_Sle4442ReadProtectionMemory
(
IN READERHANDLE hReader,
IN BYTE bStartAddress,
IN BYTE bBytesToread,
OUT PBYTE pbBuffer
IN OUT PDWORD pcbLength
);

Parameter:
hReader Quote F3_Connect return handle value
bStartAddress Address want to operate
bBytesToRead Byte numbers want to read
pbBuffer Point to return value, can not be NULL
pcbLength Provide pbBuffer parameter length (byte numbers) and receive the real return length from reader.

Return value:
Return 0 is success, other number are error code.

F3_Sle4442WriteProtectionMemory

Write protect memory
LONG
WINAPI
F3_Sle4442WriteProtectionMemory
(
IN READERHANDLE hReader,
IN BYTE bStartAddress,
IN BYTE bBytesToread,
OUT PBYTE pbBuffer,
IN OUT PDWORD pcbLength
);

Parameter:

hReader Quote F3_Connect return handle value
bStartAddress Address want to operate
bBytesToWrite Byte numbers want to write
pbBuffer Point to the data which want to write to a card. Can not be NULL.

Return value:
Return 0 is success, other number are error code.

F3_Sle4442ReadSecurityMemory

Read security memory:
LONG
WINAPI
F3_Sle4442ReadSecurityMemory
(
IN READERHANDLE hReader,
IN BYTE bStartAddress,
IN BYTE bBytesToRead,
OUT PBYTE pbBuffer,
IN OUT PDWORD pcbLength
);

Parameter:

hReader Quote F3_Connect return handle value
bStartAddress Address want to operate
bBytesToRead Byte numbers want to read
pbBuffer Point to return value, can not be NULL
pcbLength Provide pbBuffer parameter length (byte numbers) and receive the real return

length from reader.

Return value:
Return 0 is success, other number are error code.

F3_Sle4442VerifyPSC

Verify security code:

```
LONG
WINAPI
F3_Sle4442VerifyPSC
(
IN READERHANDLE hReader,
IN BYTE (&bPSCBytes)[3]
  );
```

Parameter:

hReader Quote F3_Connect return handle value
bPSCBytes Security code byte groups quantities

Return value:
Return 0 is success, other number are error code.

F3_Sle4442UpdatePSC

Update Security code:

```
LONG
WINAPI
F3_Sle4442UpdatePSC
(
IN READERHANDLE    hReader,
IN BYTE    (&bPSCBytes) [3]
);
```

Parameter:

hReader Quote F3_Connect return handle value
bPSCBytes Security code byte groups quantities

Return value:

Return 0 is success, other number are error code.

F3_Sle4442WriteErrorCounter

Write password incorrect counter:

```
LONG
WINAPI
F3_Sle4442WriteErrorCounter
(
IN READERHANDLE hReader,
IN BYTE bValue
);
```

Parameter:

hReader Quote F3_Connect return handle value
bValue Incorrect count value

Return value:
Return 0 is success, other number are error code.

RFID Card Operation Function

RFID Card Operation Function

F3_RfcActivate
RFID Card Activated

```
LONG
WINAPI
F3_RfcActivate
(
IN READERHANDLE hReader,
OUT PBYTE pbATRBuff,
IN OUT PDWORD pcbATRLength,
IN BYTE bFirstProtocol = RFC_PROTOCOL_NONE,
IN BYTE bSecondProtocol = RFC_PROTOCOL_NONE
);
```

Parameter:

hReader Quote F3_Connect return handle value

pbATRBuff Point to return reset info, if not NULL, pcbATRLength also can't be NULL
pcbATRLength Provide pbATRBuff Paramter length (quantity of bytes) and receive reader real return length.

bFirstProtocol First option protocol, available value:
RFC_PROTOCOL_NONE
RFC_PROTOCOL_TYPE_A
RFC_PROTOCOL_TYPE_B
F3_RfcActivate

bSecondProtocol Second option protocol, available value:
RFC_PROTOCOL_NONE
RFC_PROTOCOL_TYPE_A
RFC_PROTOCOL_TYPE_B

Return value:
Return 0 is success, other number are error code.

F3_RfcDeactivate

RFID Card Released

```
LONG WINAPI
F3_RfcDeactivate
(
IN READERHANDLE hReader
);
```

Parameter:

hReader Quote F3_Connect return handle value

Return value:
Return 0 is success, other number are error code.

F3_RfcGetStatus

Check RF Card Status

```
LONG
WINAPI
F3_RfcGetStatus
(
IN READERHANDLE hReader,
```

```
OUT PBYTE pbActivatedCard
);
```

Parameter:

hReader Quote F3_Connect return handle value

pbActivatedCard Return the activated RF card type,Value maybe:
RFCTYPE_UNKNOWN No RF card was activated
RFCTYPE_MIFARE_S50
RFCTYPE_MIFARE_S70
RFCTYPE_MIFARE_UL
RFCTYPE_TYPEA_CPU
RFCTYPE_TYPEB_CPU

Return value:
Return 0 is success, other number are error code.

Mifare Operation:

F3_MfVerifyPassword

Verify sector

```
LONG
WINAPI
F3_MfVerifyPassword
(
IN READERHANDLE hReader,
IN BYTE bSectorNumber,
IN BOOL fWithKeyA,
IN BYTE (&bKeyBytes)[6]
);
```

Parameter:

hReader Quote F3_Connect return handle value
bSectorNumber Sector number which want to verify
fWithKeyA Value is TRUE, verify KEY-A; Value is FALSE, verify KEY-B
bKeyBytes Groups quantity of password byte

Return value:
Return 0 is success, other number are error code.

F3_MfUpdatePassword

Update the sector password

```
LONG
WINAPI
F3_MfUpdatePassword

(
IN READERHANDLE hReader,
IN BYTE bSectorNumber,
IN BYTE (&bKeyBytes)[6]
);
```

Parameter

hReader Quote F3_Connect return handle value
bSectorNumber Sector number
bKeyBytes New password byte group number

Return value:
Return 0 is success, other number are error code.

F3_MfLoadPassword

Loading password from EEPROM and verify sector

```
LONG
WINAPI
F3_MfLoadPassword
(
IN READERHANDLE hReader,
IN BYTE bSectorNumber,
IN BOOL fWithKeyA
);
```

Parameter:

hReader Quote F3_Connect return handle value
bSectorNumber Sector number which want to verify
fWithKeyA Value is TRUE, verify KEY-A; Value is FALSE, verify KEY-B

Return value:
Return 0 is success, other number are error code.

F3_MfDownloadPassword

Download password to EEPROM

```
LONG
WINAPI
F3_MfDownloadPassword
(
IN READERHANDLE hReader,
IN BYTE bSectorNumber,
IN BOOL fWithKeyA,
IN BYTE (&bKeyBytes)[6]
);
```

Parameter:

hReader Quote F3_Connect return handle value
bSectorNumber Sector number which want to verify
fWithKeyA Value is TRUE, verify KEY-A; Value is FALSE, verify KEY-B
bKeyBytes Groups quantity of password byte

Return value:
Return 0 is success, other number are error code.

F3_MfReadSector

Read sector block data

```
LONG
WINAPI
F3_MfReadSector
(
IN READERHANDLE hReader,
IN BYTE bSectorNumber,
IN BYTE bStartBlockNumber,
IN BYTE bBlocksToRead,
OUT PBYTE pbBuffer,
IN OUT PDWORD pcbLength
);
```

Parameter:

hReader Quote F3_Connect return handle value
bSectorNumber Sector number which want to operate
bStartBlockNumber The start block number

bBlocksToRead Block number which want to read
pbBuffer Point to return data, can't be NULL
pcbLength Provide pbBuffer Paramter length（number of byte）and receive card reader return real length

Return value:
Return 0 is success, other number are error code.

F3_MfWriteSector

Write sector's block data

```
LONG
WINAPI
F3_MfWriteSector

(
IN READERHANDLE hReader,
IN BYTE bSectorNumber,
IN BYTE bStartBlockNumber,
IN BYTE bBlocksToRead,
OUT PBYTE pbBuffer
);
```

Parameter:

hReader Quote F3_Connect return handle value
bSectorNumber Sector number which want to operate
bStartBlockNumber The start block number
bBlocksToRead Block number which want to write
pbBuffer Point to the data which want to write, can't be NULL

Return value:
Return 0 is success, other number are error code.

F3_MfInitializeValue

Initialized block value

```
LONG
WINAPI
F3_MfInitializeValue
(
```

IN READERHANDLE hReader,
IN BYTE bSectorNumber,
IN BYTE bBlockNumber,
IN UINT32 iValue
);

Parameter:

hReader Quote F3_Connect return handle value
bSectorNumber Sector number which want to operate
bBlockNumber Block number which want to operate
iValue value to the Initialize block

Return value:
Return 0 is success, other number are error code.

F3_MfReadValue

Read value block

LONG
WINAPI
F3_MfReadValue
(
IN READERHANDLE hReader,
IN BYTE bSectorNumber,
IN BYTE bBlockNumber,
OUT UINT32 *piValue
);

Parameter:

hReader Quote F3_Connect return handle value
bSectorNumber Sector number which want to operate
bBlockNumber Block number which want to operate
piValue Return block value

Return value:
Return 0 is success, other number are error code.

F3_MfIncrementValue

Increase value

```
LONG
WINAPI
F3_MfIncrementValue
(
IN READERHANDLE hReader,
IN BYTE bSectorNumber,
IN BYTE bBlockNumber,
IN UINT32 iValue
);
```
Parameter:

hReader Quote F3_Connect return handle value
bSectorNumber Sector number which want to operate
bBlockNumber Block number which want to operate
iValue Value which want to add to block

Return value:
Return 0 is success, other number are error code.

F3_MfDecrementValue

Minus value

```
LONG
WINAPI
F3_MfDecrementValue
(
IN READERHANDLE hReader,
IN BYTE bSectorNumber,
IN BYTE bBlockNumber,
IN UINT32 iValue
);
```

Parameter:

hReader Quote F3_Connect return handle value
bSectorNumber Sector number which want to operate
bBlockNumber Block number which want to operate
iValue Value which want to minus from the block

Return value:
Return 0 is success, other number are error code.

Card dispenser Stucture Status:

Channel information structure:

```
typedef struct _CRSTATUS
{
BYTE bLaneStatus;
BYTE bCardBoxStatus;
BOOL fCaptureBoxFull;
} CRSTATUS, *PCRSTATUS;
```

Members:

bLaneStatus Channel status, available value:

LS_NO_CARD_IN Have no card in the machine
LS_CARD_AT_GATE_POS Card in the export card position
LS_CARD_IN Have card in

bCardBoxStatus Card stacker status, available value:

CBS_EMPTY Card stacker is empty
CBS_INSUFFICIENT Card stacker card less
CBS_ENOUGH Card stacker have enough card insdie

fCaptureBoxFull Value is TRUE，Means collect stacker was full; Value is FALSE, meanshaven't full.

Error Code descriptions:

F3_S_SUCCESS Operate success

F3_E_PORT_UNAVAILABLE COM are not exist OR was occupied

F3_E_DEV_NOT_RECOGNIZED Device not detected, may cause by:
1 COM number incorrect
2 Baud rate incorrect
3 Address incorrect
4 COM cable has problem
(Remark: only after quote F3_Connect functions, then may return such error)

F3_E_COMM_ERROR Communicate error, may cause by:
1 The receive character haven't define in the communication protocol.
2 Return response info's head package,package tail or BBC incorrect.
3 Return response data length different from the communication protocol define.

F3_E_COMM_TIMEOUT Communication time-out
F3_E_UNKNOWN_ERROR Inner error, need to check
F3_E_MESSAGE_TOO_LONG message length over 1024 byte
F3_E_NO_MEMORY Has no enough RAM for present operate
F3_E_BUFFER_TOO_SMALL Receive return data's Buffer too small.
F3_E_INVALID_HANDLE invalid handle
F3_E_UNDEFINED_COMMAND Undifine command
F3_E_INVALID_PARAMETER Provide one or several parameter invalid or be NULL value.
F3_E_DISABLED_COMMAND Command can't execute in present status.
F3_E_UNSUPPORTED_COMMAND Un support command
F3_E_CONTACT_NO_RELEASE IC contact haven't released
F3_E_CARD_JAMMED Card jam
F3_E_SENSOR_ABNORMALITY Sensor abnormity
F3_E_TOO_LONG_CARD Card length too long
F3_E_TOO_SHORT_CARD Card length too short
F3_E_CARD_WITHDRAWN When collect card, the card was taken
F3_E_IC_SOLENOID_ERROR IC Electromagnetic coil error
F3_E_CANT_MOVED_TO_IC_POS Can't move card to IC contact position
F3_E_CARD_POSITION_CHANGE Card position was change by hand operate
F3_E_COUNTER_OVERFLOW Collect card counter overflow
F3_E_MOTOR_ABNORMALITY Motor abnormity
F3_E_POWER_SHORT IC Card power supplier short circuit
F3_E_ICC_ACTIVATION_ERROR IC card activated error
F3_E_ICC_NOT_ACTIVATED IC card didn't activate
F3_E_UNSUPPORTED_ICC Not support such IC card
F3_E_ICC_RECEPTION_ERROR Receive data from IC card error
F3_E_ICC_COMM_TIMEOUT IC card communication time out
F3_E_MISMATCH_EMV CPU/SAM card not compatiable with EMV2000 directive
F3_E_CARD_BOX_EMPTY Card stacker empty
F3_E_CAPTURE_BOX_FULL Card collect stacker was full
F3_E_WAITING_FOR_RESET Waiting for reset
F3_E_COMMAND_FAILURE Command execute fail
F3_E_DISAGREEMENT_OF_VC Verify code incorrect
F3_E_CARD_LOCKED Card was locked
F3_E_ADDRESS_OVERFLOW Operate address overflow
F3_E_LENGTH_OVERFLOW Operate length overflow