Programmer's Guide for Fingerprint's SDK

Wison Technology Corp.

Version: 2.6

Addr: 11F-2, No. 289, Sec. 2, Guang-Fu Rd., Hsin-Chu 300, Taiwan, R.O.C. Tel: 886-3-5163339
Fax: 886-3-5163679

1. Features

- ➢ Wison Technolgy OR200N Optical Sensor
- ➢ Support Windows 2000/XP/Vista/Windows 7

2. SDK Files

File Name	Descriptions
OR200N.Inf, OR200N.sys	Driver for OR200N
PTSDK4_WISCMOS2_PTFV.h	Header file of Fingerprint Verification Algorithem and the sensor control of OR200N
PTSDK4_WISCMOS2_PTFV.dll PTSDK4_WISCMOS2_PTFV.lib	DLL files of Fingerprint Algorithem and the sensor control of OR200N
PTSDK4_WISCMOS2_PTFV.lib	Static link library for AP development

3. SDK Functions

Function Name	Descriptions
bAPI4_OpenDevice	Open the fingerprint USB device
bAPI4_OpenSensor	Open fingerprint sensor
bAPI4_GetDeviceNum	Get device number
bAPI4_SelectDevice	Select device
bAPI4_CloseSensor	Close fingerprint sensor
bAPI4_GetImage	Get the gray fingerprint image
bAPI4_GetBinaryImage	Get the binary fingerprint image
bAPI4_StartRealtimeImage	Start real time image capturing
bAPI4_GrabRealtimeImage	Grab real time image without quality check
bAPI4_StopImage	Stop the get image
bAPI4_CheckSensorStatus	Check sensor status
bAPI4_HMFVOpenLib	Enable the fingerprint algorithm functions
bAPI4_HMFVCLoseLib	Disable the fingerprint algorithm functions
bAPI4_HMFVStartEnroll	Start the fingerprint enrollment
bAPI4_HMFVEnroll	Fingerprint enrollment
bAPI4_HMFVVerify	Fingerprint verification by image
bAPI4_HMFVExtract	Fingerprint feature extraction (Reserved)
bAPI4_HMFVVerifyTemplate	Fingerprint verification by feature (Reserved)
bAPI4_HMFVSetParas	Set parameter to fingerprint algorithm
bAPI4_HMFVGetParas	Get parameter from fingerprint algorithm
bAPI4_ReadPID	Read Product ID from EEPROM
bAPI4_ReadSN	Read Serial Number from EEPROM
bAPI4_ReadMFS	Read Manufacturer Information from EEPROM
bAPI4_ReadPDS	Read Product Information from EEPROM
bAPI4_ReadInfo	***Read Customer Defined Information from EEPROM***
bAPI4_WriteInfo	***Write Customer Defined Information to EEPROM***

4. Parameters

4.1 Fingerprint Window Size

Define the fingerprint window size, unit in Pixel, in this case, the fingerprint window size is 256 pixels * 288 pixels

```
#define         SENSOR_WIDTH                256
#define         SENSOR_HEIGHT               288
```

4.2 Resolution

Define the sensor image resolution.

```
#define         SENSOR_RESOLUTION           500
```

4.3 Maximum Template Size

Define the maximum template size of an enrolled fingerprint.

```
#define         HMFV_MAX_TEMPLATE_SIZE      500
```

4.4 Algorithm Kernel Status

Value return while algorithm kernel function fails to call.

```
#define         HMFV_LIB_NOT_OPENED         -100
#define         HMFV_STS_MALLOC_FAIL        -101
```

4.5 Get Image Status

Value return while get image function time out.

```
#define         HMFV_STS_GI_TIMEOUT         -3
```

4.6 Enrollment Status

```
#define         HMFV_STS_EN_CONTINUE        1
#define         HMFV_STS_EN_SUCCESS         0
#define         HMFV_STS_EN_FAIL            -1
#define         HMFV_STS_EN_NOINIT          -2
#define         HMFV_STS_EN_TOOMANY_POORIMG -3
#define         HMFV_STS_EN_TOOMAY_TRIALS   -4
```

4.7 Verification Status

```
#define        HMFV_STS_VF_SUCCESS          2
#define        HMFV_STS_VF_FAIL             1
#define        HMFV_STS_VF_POORIMG          -1
#define        HMFV_STS_VF_ERROR            -2
```

4.8 Security Level

```
#define        PARAID_MATCHTHRESHOLD        100
#define        LOW_MATCH_TH                 0
#define        MID_MATCH_TH                 1
#define        HIGH_MATCH_TH                2
```

4.9 EEPROM Field Length

```
#define        PID_VALUE_SIZE               2
#define        SN_VALUE_SIZE                8
#define        MFS_VALUE_SIZE               16
#define        PDS_VALUE_SIZE               16
#define        INFO_VALUE_SIZE              32
```

5. Functions

5.1 Open Device

Synopsis : bAPI4_OpenDevice ()

Description : Call this function to open the fingerprint USB device. This is unused.

Parameter : None.

Return Value :

FALSE (0)	Open device NG.
TRUE (1)	Open device OK.

5.2 Open Sensor

Synopsis : bAPI4_OpenSensor ()

Description : Call this function to open the fingerprint sensor.

Parameter : None.

Return Value :

FALSE (0)	Open sensor NG.
TRUE (1)	Open sensor OK.

5.3 Get Device Number

Synopsis : bAPI4_GetDeviceNum ()

Description : Call this function to get the number of device.

Parameter : None.

Return Value :

Integer Value	Number of device.

5.4 Select Device

Synopsis : bAPI4_SelectDevice (int iDevice)

Description : Call this function to select the device to open it.

Parameter : None.

Return Value :

FALSE (0)	Select device NG.
TRUE (1)	Select device OK.

5.5 Close Sensor

Synopsis : bAPI4_CloseSensor ()

Description : Call this function to close the fingerprint sensor.

Parameter : None.

Return Value :

FALSE (0)	Open sensor NG.
TRUE (1)	Open sensor OK.

5.6 Get Fingerprint Image

Synopsis : bAPI4_GetImage (BYTE *picture, int timeout, int iResolution,
int *piWidth, int *piHeight)

Description : Call the bAPI4_GetImage to get the **gray** fingerprint image with quality check.

Parameter :

picture	Pointer to an image buffer
timeout	Timeout period for getting image (millisecond)
iResolution	Image resolution ***(Default: SENSOR_RESOLUTION)***
piWidth	Returned image width
piHeight	Returned image height

Return Value :

FALSE (0)	Get gray image NG.
TRUE (1)	Get gray image OK.

Synopsis : bAPI4_GetBinaryImage (BYTE *picture, int timeout, int iResolution,
int *piWidth, int *piHeight)

Description : Call the bAPI4_GetBinaryImage to get the **binary** fingerprint image with quality check.

Parameter :

picture	Pointer to an image buffer
timeout	Timeout period for getting image (millisecond)
iResolution	Image resolution ***(Default: SENSOR_RESOLUTION)***
piWidth	Returned image width
piHeight	Returned image height

Return Value :

FALSE (0)	Get binary image NG.
TRUE (1)	Get binary image OK.

Synopsis : bAPI4_StartRealtimeImage (int iResolution, int *piWidth, int *piHeight)

Description : Call this function to start real time image capturing. (**gray** image only)

Parameter :

iResolution	Image resolution ***(Default: SENSOR_RESOLUTION)***
piWidth	Returned image width
piHeight	Returned image height

Return Value :

FALSE (0)	Start real time image capturing NG.
TRUE (1)	Start real time image capturing OK.

Synopsis : bAPI4_GrabRealtimeImage (BYTE *picture)

Description : Call this function to get the **gray** fingerprint image without quality check for real time display.

Parameter :

picture	Pointer to an image buffer

Return Value :

FALSE (0)	Get real time image NG.
TRUE (1)	Get real time image OK.

Synopsis : bAPI4_StopImage ()

Description : Call the bAPI4_StopImage to stop getting image.

Parameter : None.

Return Value :

FALSE (0)	Stop getting image NG.
TRUE (1)	Stop getting image OK.

5.7 Check Sensor Status

Synopsis : bAPI4_CheckSensorStatus (BOOL *pbStatus)

Description : Call this function to check the sensor status.

Parameter :

pbStatus	Returned current status FALSE (0) : sensor is not found TRUE (1) : sensor is ready

Return Value :

FALSE (0)	Check sensor status NG.
TRUE (1)	Check sensor status OK.

5.8 Fingerprint Algorithm

Synopsis : bAPI4_HMFVOpenLib ()

Description : Call this function to enable the fingerprint alogorithm functions.

Parameter : None.

Return Value :

FALSE (0)	Open algorithm kernel functions NG.
TRUE (1)	Open algorithm kernel functions OK.

Synopsis : bAPI4_HMFVCLoseLib ()

Description : Call this function to disable the fingerprint alogorithm functions.

Parameter : None.

Return Value :

FALSE (0)	Close algorithm kernel functions NG.
TRUE (1)	Close algorithm kernel functions OK.

Synopsis : bAPI4_HMFVSetParas(int iParaID, int iParaValue)

Description : Call this function to set parameters to algorithm kernel.

Parameter :

iParaID	Parameter ID
iParaValue	Parameter Value

Return Value :

FALSE (0)	Set parameter NG.
TRUE (1)	Set parameter OK.

Synopsis : bAPI4_HMFVGetParas(int iParaID, int *piParaValue)

Description : Call this function to get parameters from algorithm kernel.

Parameter :

iParaID	Parameter ID
iParaValue	Returned Parameter Value

Return Value :

FALSE (0)	Get parameter NG.
TRUE (1)	Get parameter OK.

5.9 Fingerprint Enrollment

Synopsis : bAPI4_HMFVStartEnroll(int iDefaultEnrolledTimes)

Description : Call this function to start the fingerprint enrollment.

Parameter :

iDefaultEnrolledTimes	Times to enroll fingerprint ***(Recommended : 10)***

Return Value :

FALSE (0)	Starting enrollment NG.
TRUE (1)	Starting enrollment OK.

Synopsis : bAPI4_HMFVEnroll(int iResolution, int iWidth, int iHeight,
BYTE * pFingerImage,
BYTE *pEnrolledFeatures, DWORD *pwEnRetSize,
int *piStatus)

Description : Call this function to do fingerprint enrollment.

Parameter :

iResolution	Image Resolution
iWidth	Image Width
iHeight	Image Height
pFingerImage	Pointer to image buffer ready for enrollment
pEnrolledFeatures	Pointer to buffer for keeping returned features
pwEnRetSize	Returned enrolled feature size
piStatus	Returned enrollment status ***(Refer to 4.5 Enrollment Status)***

Return Value :

FALSE (0)	Fingerprint enrollment NG.
TRUE (1)	Fingerprint enrollment OK.

5.10 Fingerprint Verification by Image

Synopsis : bAPI4_HMFVVerify(int iResolution, int iWidth, int iHeight,
BYTE *pFingerImage,
BYTE **ppEnrolledfeatures, int iEnrolledNum,
int *piMatchedID,
int *piStatus)

Description : Call this function to do fingerprint verification.

Parameter :

iResolution	Image Resolution
iWidth	Image Width
iHeight	Image Height
pFingerImage	Pointer to image buffer ready for verification
ppEnrolledfeatures	Pointer to Enrolled feature buffer address
iEnrolledNum	Number of Enrolled features to be verified
piMatchedID	Returned Matched ID
piStatus	Returned Verification Status ***(Refer to 4.6 Verification Status)***

Return Value :

FALSE (0)	Fingerprint Verification by Image NG.
TRUE (1)	Fingerprint Verification by Image OK.

5.11 Fingerprint Feature Extraction

Synopsis : bAPI4_HMFVExtract(int iResolution, int iWidth, int iHeight,
BYTE *pFingerImage,
BYTE *pFeatures, DWORD *pwFeatSize,
int *piQuality)

Description : Call this function to do fingerprint feature extraction.

Parameter :

iResolution	Image Resolution
iWidth	Image Width
iHeight	Image Height
pFingerImage	Pointer to image buffer ready for feature extraction
pFeatures	Pointer to extracted feature buffer address
pwFeatSize	Size of extracted feature
piQuality	Returned Quality of extracted feature

Return Value :

FALSE (0)	Fingerprint Feature Extraction NG.
TRUE (1)	Fingerprint Feature Extraction OK.

5.12 Fingerprint Verification by Feature

Synopsis : bAPI4_HMFVVerifyTemplate(BYTE *pTemplate,
BYTE **ppEnrolledfeatures, int iEnrolledNum,
int *piMatchedID, int *piStatus)

Description : Call this function to do fingerprint verification by extracted feature.

Parameter :

pTemplate	Pointer to extracted feature buffer ready for verification
ppEnrolledfeatures	Pointer to Enrolled feature buffer address
iEnrolledNum	Number of Enrolled features to be verified
piMatchedID	Returned Matched ID
piStatus	Returned Verification Status ***(Refer to 4.5 Verification Status)***

Return Value :

FALSE (0)	Fingerprint Verification by Feature NG.
TRUE (1)	Fingerprint Verification by Feature OK.

5.13 Read/Write EEPROM (OR200E Only)

Synopsis : bAPI4_ReadPID (BYTE *pBuf)

Description : Call this function to read the Product ID from EEPROM.

Parameter :

pBuf	Pointer to data buffer ***(Refer to 4.9 EEPROM Field Length)***

Return Value :

FALSE (0)	Read EEPROM NG.
TRUE (1)	Read EEPROM OK.

Synopsis : bAPI4_ReadSN (BYTE *pBuf)

Description : Call this function to read the Serial Number from EEPROM.

Parameter :

pBuf	Pointer to data buffer ***(Refer to 4.9 EEPROM Field Length)***

Return Value :

FALSE (0)	Read EEPROM NG.
TRUE (1)	Read EEPROM OK.

Synopsis : bAPI4_ReadMFS (BYTE *pBuf)

Description : Call this function to read the Manufacturer Information from EEPROM.

Parameter :

pBuf	Pointer to data buffer ***(Refer to 4.9 EEPROM Field Length)***

Return Value :

FALSE (0)	Read EEPROM NG.
TRUE (1)	Read EEPROM OK.

Synopsis : bAPI4_ReadPDS (BYTE *pBuf)

Description : Call this function to read the Product Information from EEPROM.

Parameter :

pBuf	Pointer to data buffer ***(Refer to 4.9 EEPROM Field Length)***

Return Value :

FALSE (0)	Read EEPROM NG.
TRUE (1)	Read EEPROM OK.

Synopsis : bAPI4_ReadInfo (BYTE *pBuf)

Description : Call this function to read the Customer defined Information from EEPROM.

Parameter :

pBuf	Pointer to data buffer The maximum allowed size to read is 32 bytes ***(Refer to 4.9 EEPROM Field Length)***

Return Value :

FALSE (0)	Read EEPROM NG.
TRUE (1)	Read EEPROM OK.

Synopsis : bAPI4_WriteInfo (BYTE *pBuf)

Description : Call this function to write the Customer defined Information to EEPROM.

Parameter :

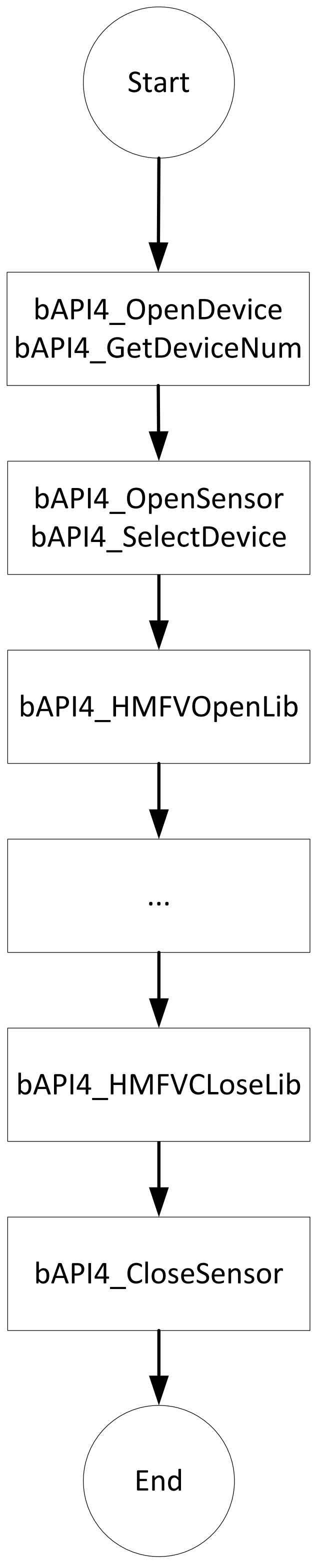
pBuf	Pointer to data buffer The maximum allowed size to write is 32bytes ***(Refer to 4.9 EEPROM Field Length)***

Return Value :

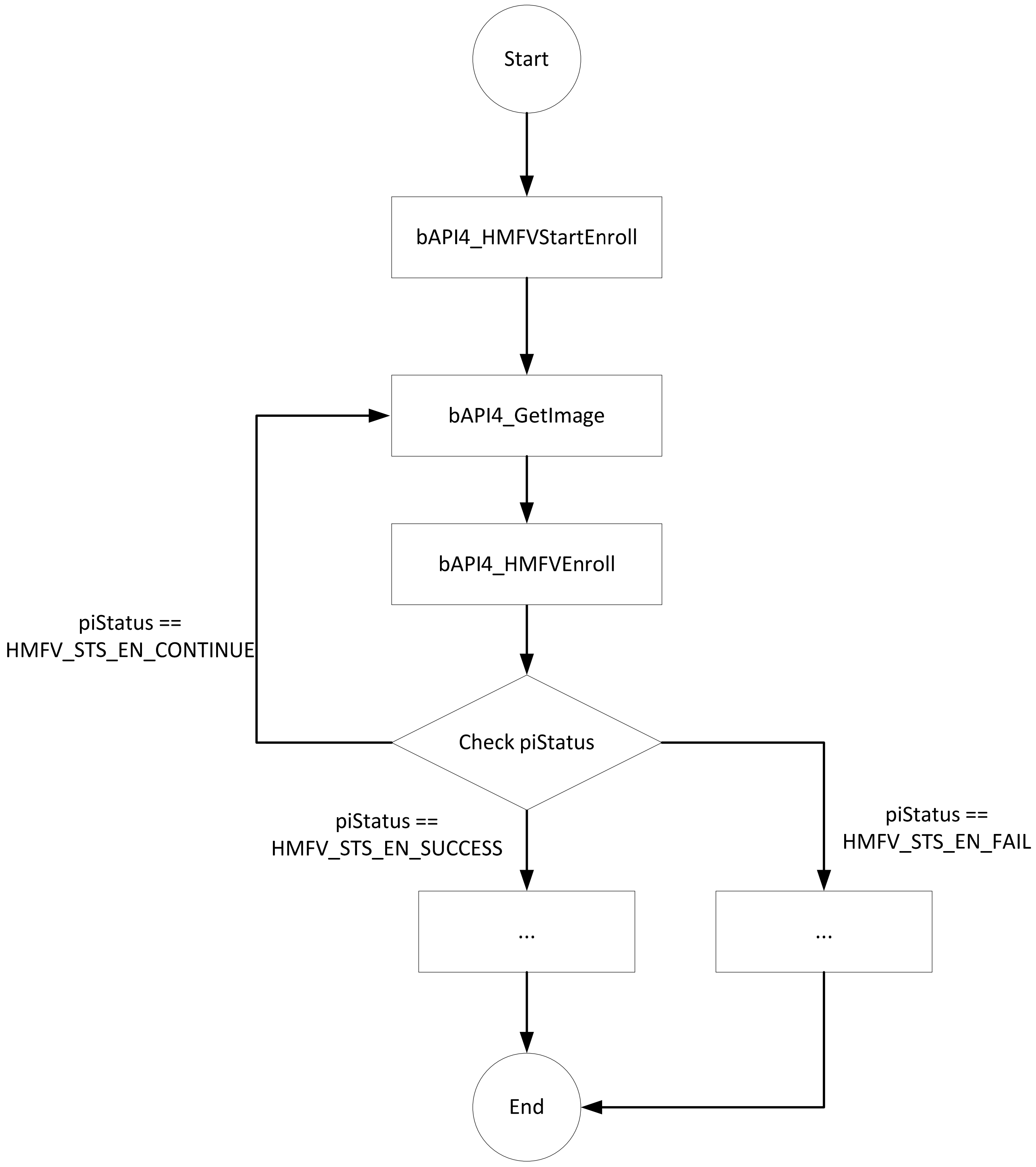
FALSE (0)	Write EEPROM NG.
TRUE (1)	Write EEPROM OK.

6. Control Flows

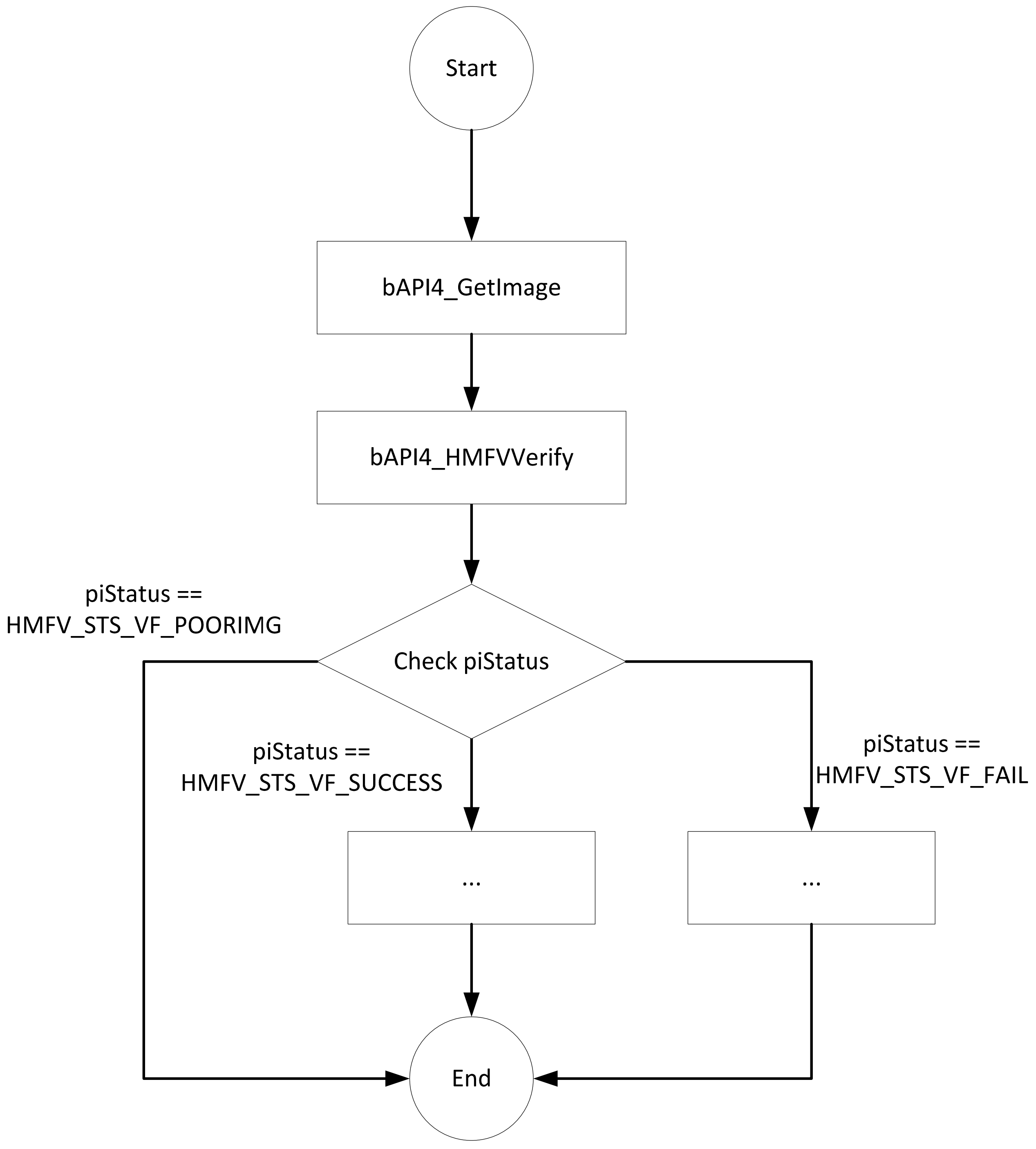
6.1 SDK Enable/ Disable Flow



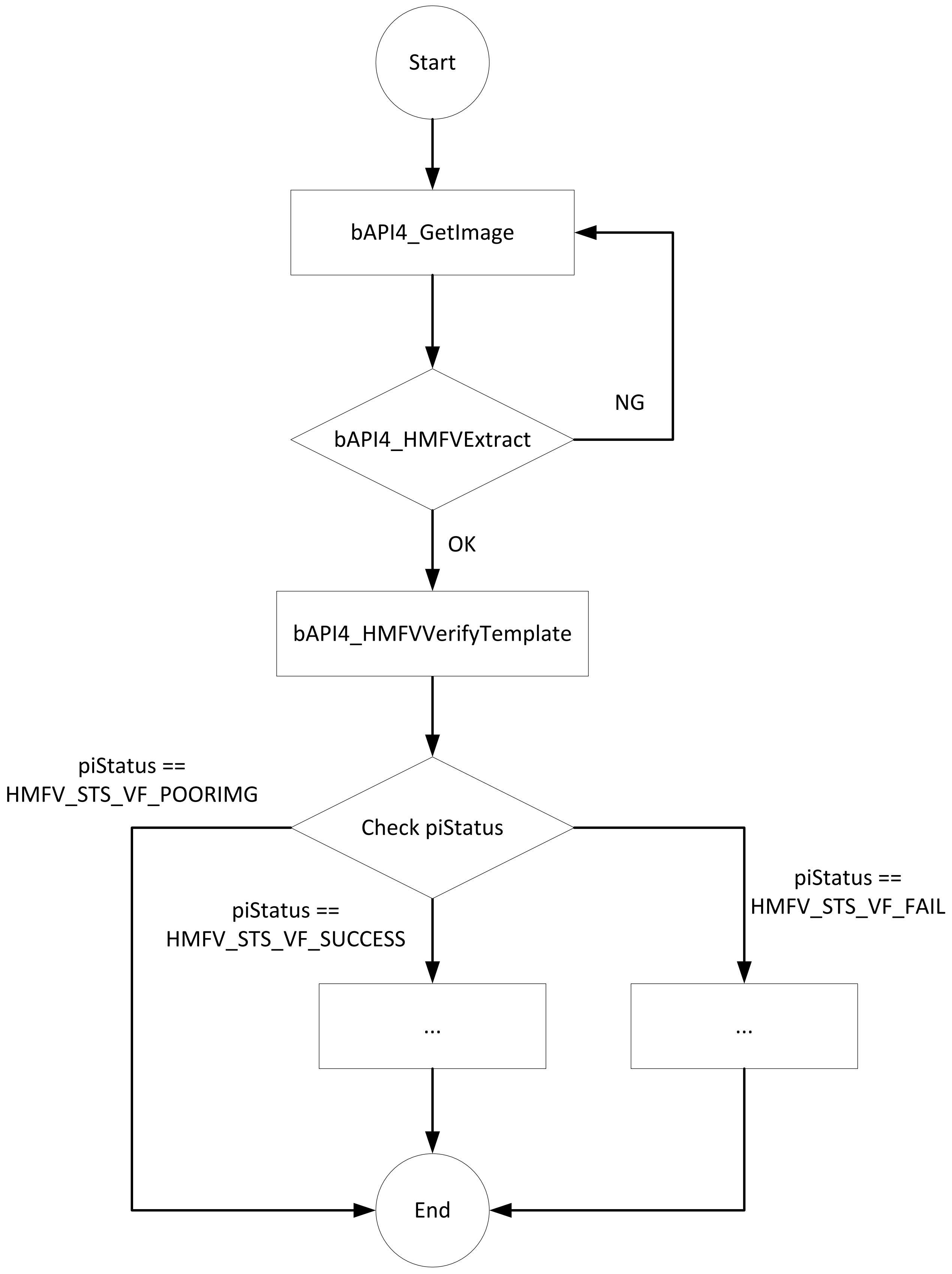
6.2 Fingerprin Enrollment Flow


Start
bAPI4_HMFVStartEnroll
bAPI4_GetImage
bAPI4_HMFVEnroll
piStatus == HMFV_STS_EN_CONTINUE
Check piStatus
piStatus == HMFV_STS_EN_SUCCESS
piStatus == HMFV_STS_EN_FAIL
...
...
End

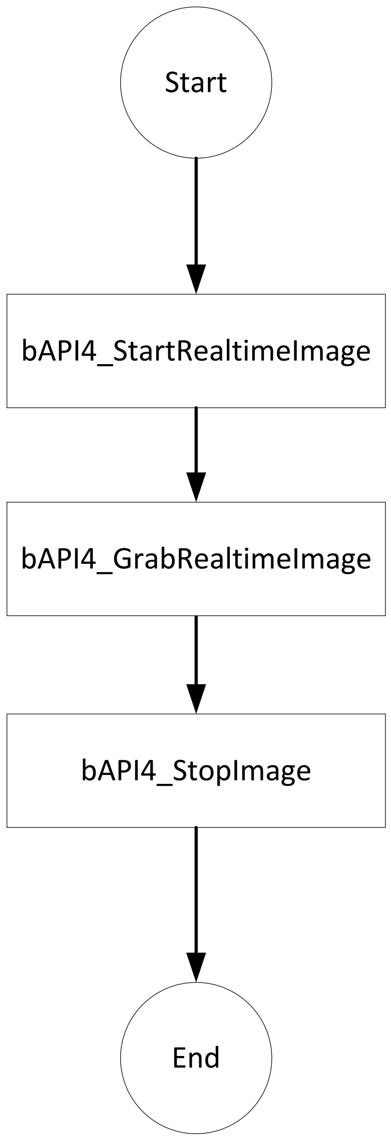
6.3 Fingerprint Verification by Image Flow


Start
bAPI4_GetImage
bAPI4_HMFVVerify
Check piStatus
piStatus == HMFV_STS_VF_POORIMG
piStatus == HMFV_STS_VF_SUCCESS
piStatus == HMFV_STS_VF_FAIL
...
...
End

6.4 Fingerprint Verification by Feature Flow (Reserved)


Start
bAPI4_GetImage
bAPI4_HMFVExtract
NG
OK
bAPI4_HMFVVerifyTemplate
piStatus == HMFV_STS_VF_POORIMG
Check piStatus
piStatus == HMFV_STS_VF_SUCCESS
piStatus == HMFV_STS_VF_FAIL
...
...
End

6.5 Get Pure Image without Quality Check


Start
bAPI4_StartRealtimeImage
bAPI4_GrabRealtimeImage
bAPI4_StopImage
End