

Programmer's Guide for Fingerprint's SDK



Wislon Technology Corp.

*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

- Wislon Technolgy OR200 Optical Sensor
- Support Windows 2000/XP/Vista

2. SDK Files

File Name	Descriptions
WISCMOS2.Inf, WISCMOS2.sys	Driver for OR200
PTSDK4_WISCMOS2_PTFV.h	Header file of Fingerprint Verification Algorithm and the sensor control of OR200
PTSDK4_WISCMOS2_PTFV.dll	DLL file of Fingerprint Verification Algorithm and the sensor control of OR200
WisCMOS2.dll, DEC207.dll	These DLL files are for controlling the OR200 Sensor and USB communication
PTFVLib.dll	DLL file of Fingerprint Verification Algorithm Kernel

3. SDK Functions

Function Name	Descriptions
bAPI4_OpenDevice	Open the fingerprint USB device (unused)
bAPI4_OpenSensor	Open fingerprint sensor
bAPI4_CloseSensor	Close fingerprint sensor
bAPI4_GetDeviceNum	Get the number of fingerprint sensors on the PC
bAPI4_SelectDevice	Open the selected fingerprint sensor
bAPI4_LightLedOff	Turn sensor led off
bAPI4_GetImage	Get the gray fingerprint image
bAPI4_GetBinaryImage	Get the binary fingerprint image
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
bAPI4_HMFVSetParas	Set parameter to fingerprint algorithm
bAPI4_HMFVGetParas	Get parameter from fingerprint algorithm

4. Parameters

4.1 Fingerprint Window Size

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

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

4.2 Resolution

```
#define SENSOR_RESOLUTION    500
```

Define the sensor image resolution.

4.3 Maximum Template Size

```
#define HMFV_MAX_TEMPLATE_SIZE 500
```

Define the maximum template size of an enrolled fingerprint.

4.4 Algorithm Kernel Status

```
#define HMFV_LIB_NOT_OPENED  -100
```

Value return while algorithm kernel function fails to call.

4.5 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
```

4.6 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
```

4.7 Security Level

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

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 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.

5.4 Get Device Number

Synopsis : bAPI4_GetDeviceNum ()

Description : Call this function to get the number of fingerprint sensor on the PC.

Parameter : None.

Return Value :

iDeviceNum	The number of fingerprint sensor on the PC.
------------	---

5.5 Select Device

Synopsis : bAPI4_SelectDevice (int iDevice)

Description : Call this function to open the selected fingerprint sensor.

Parameter :

iDevice	The number of the selected sensor to open.
---------	--

Return Value :

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

5.6 Light Led Off

Synopsis : bAPI4_LightLedOff ()

Description : Call this function to turn the sensor led off.

Parameter : None.

Return Value :

FALSE (0)	Turn sensor led off NG.
TRUE (1)	Turn sensor led off OK.

5.7 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.

Parameter :

picture	Pointer to an image buffer
timeout	Timeout period for getting image (millisecond)
iResolution	Image resolution (Default: <i>SENSOR_RESOLUTION</i>)
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.

Parameter :

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

Return Value :

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

5.8 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.

5.9 Fingerprint Algorithm

Synopsis : bAPI4_HMFVOpenLib ()

Description : Call this function to enable the fingerprint algorithm 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 algorithm 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.

5.10 Fingerprint Enrollment

Synopsis : bAPI4_HMFVStartEnroll(int iDefaultEnrolledTimes)

Description : Call this function to start the fingerprint enrollment.

Parameter :

iDefaultEnrolledTimes	Times to enroll fingerprint (<i>Recommended : 10</i>)
-----------------------	--

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 <i>(Refer to 4.5 Enrollment Status)</i>

Return Value :

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

5.11 Fingerprint Verification

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 :

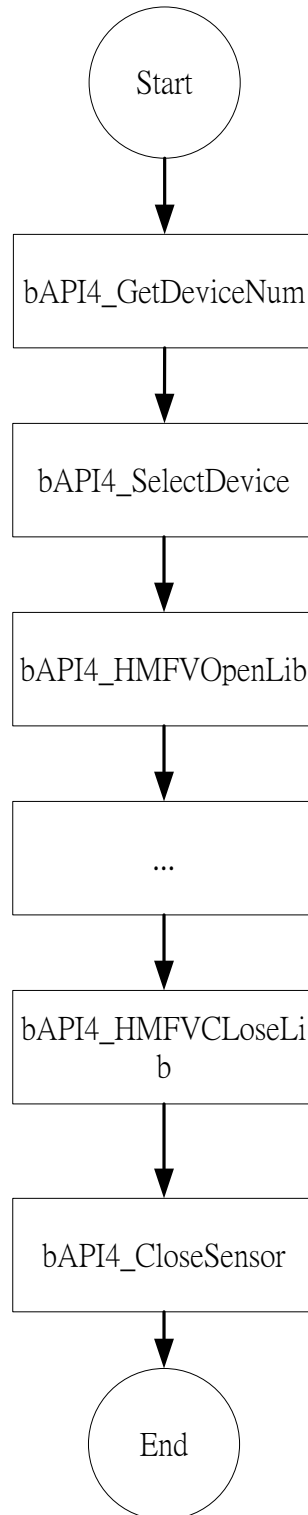
iResolution	Image Resolution
iWidth	Image Width
iHeight	Image Height
pFingerImage	Pointer to image buffer ready for verification
ppEnrolledfeatures	Pointer to Enrolled feature buffer address
iEnrolledNum	Number of Enrolled features to be verified
piMatchedID	Returned Matched ID
piStatus	Returned Verification Status <i>(Refer to 4.5 Verification Status)</i>

Return Value :

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

6. Control Flows

6.1 SDK Enable/ Disable Flow



6.2 Fingerprint Enrollment Flow

