

CONTENTS

Pin Function	2
Reduce the Standby Power Consumption	3
Using the Module with an Arduino Uno	3
Logic Level Converter	3
Wire Connection	4
Download and Execute Example Code	5
Enroll.....	6
Identify	7
Get Enrolled ID Count.....	8
Delete ID	8
Using the SDK_DEMO Software	10
Wire Connection	10
Download and Execute SDK_DEMO	12
View Module Information.....	13
Enroll.....	13
Verify (1:1).....	15
Identify (1:N)	16

Pin Function

There are two 4-pin JST SH connectors on the GTM-5210F52 Fingerprint Sensor Module. These are the UART connector and the Touch IC connector. The pin assignments and functions are shown below.

UART Connector

Touch IC Connector



UART connector pin function:

Pin No.	Pin Name	Function
1	Tx	Serial interface (UART) transmitted data. The voltage level is 3.3V.
2	Rx	Serial interface (UART) received data. The voltage level is 3.3V.
3	GND	Ground
4	VCC	Supply voltage. The voltage range is 3.3V ~ 6V DC. This power supply is regulated by the module's internal voltage regulator and supplies power to the complete module with the exception of the touch IC on the module. The touch IC uses an individual power supply.

Touch IC connector pin function:

Pin No.	Pin Name	Function
1	VDD	Supply Voltage. Connect to 3.3V DC. This power supply is used to power only the touch IC of the fingerprint sensor module.
2	GND	Ground
3	ICPDA	In the normal mode, this pin is not used. In the programming mode, this pin is used as the in-circuit programming data/address pin.
4	ICPCK	In the normal mode, this pin is the touch IC output signal and is active high. When a finger touches the metal frame of the fingerprint sensor, this signal will remain high. In the programming mode, this pin is used as the in-circuit programming clock pin.

There is no need to program the touch IC to use this module as the touch IC is programmed during IC production.

This module is shipped without four wire cables. Users must prepare a cable assembly to use this module.

Reduce the Standby Power Consumption

The touch IC within this module is used to detect whether finger contact is made on the sensor. When a finger touches the fingerprint sensor metal frame, the touch IC connector pin 4 (ICPCK) output will be high, otherwise it remains at a low level.

In daily operation we only need to power on the fingerprint sensor module after finger contact is made on the sensor to reduce the standby power consumption. We can use external circuits or a microcontroller (such as an Arduino board) to detect the ICPCK signal and then supply power to the UART connector pin 4 (VCC) when the ICPCK signal is high.

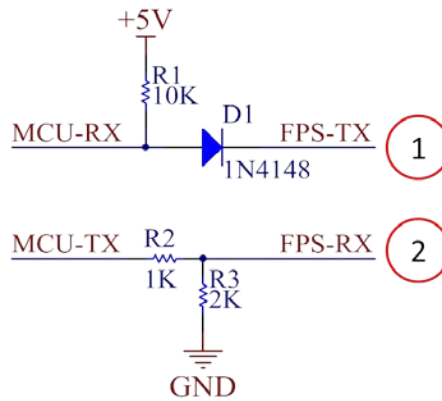
When applying this method to save power, remember to always supply power to the touch IC. However, if the touch IC function is not required, the touch IC connector can remain unused, allowing the fingerprint sensor module to keep working normally. In this case, remember to always supply power to the module at the UART connector pin 4 (VCC).

Using the Module with an Arduino Uno

This module has many protocol commands. External controllers can use these protocol commands for its operation. In this section we use the Arduino Uno board to work alongside the fingerprint sensor module, the GTM-5210F52.

Logic Level Converter

The Arduino Uno board operates at 5V while the fingerprint sensor module UART interface is at a 3.3V level. As a result of these level differences a logic level converter is required. Below is a simple logic level converter circuit. Note that this circuit is for 5V/3.3V TTL level converting. If a 5V controller board is not being used, do not use this circuit.



5V/3.3V logic Level Converter Circuit

Wire Connection

If you use the example codes downloaded from the DOCUMENTS page to operate the module, connect the wires as shown below.

Arduino UNO board:

Pin	Function	Wire Connection
4	Software serial RX	Connect to logic level converter circuit 'MCU-RX' signal
5	Software serial TX	Connect to logic level converter circuit 'MCU-TX' signal
6	Finger press detection	Connect to Touch IC connector pin 4 (ICPCK) of the fingerprint sensor module. Note: we don't use this signal in our example codes, so you may leave this pin unconnected.
5V	5V power supply	Connect to UART connector pin 4 (VCC) of the fingerprint sensor module. Also connect to logic level converter circuit '+5V' signal
3.3V	3.3V power supply	Connect to Touch IC connector pin 1 (VDD) of the fingerprint sensor module. Supply power to the touch IC.
GND	Ground	Connect to UART connector pin 3 (GND), Touch IC connector pin 2 (GND) and logic level converter circuit 'GND' signal. Connect all grounds together.

Fingerprint Sensor Module GTM-5210F52

Connector	Pin	Pin Name	Wire Connection
UART	1	TX	Connect to logic level converter circuit 'FTP-TX' signal
	2	RX	Connect to logic level converter circuit 'FTP-RX' signal
	3	GND	Connect to Arduino UNO board GND pin

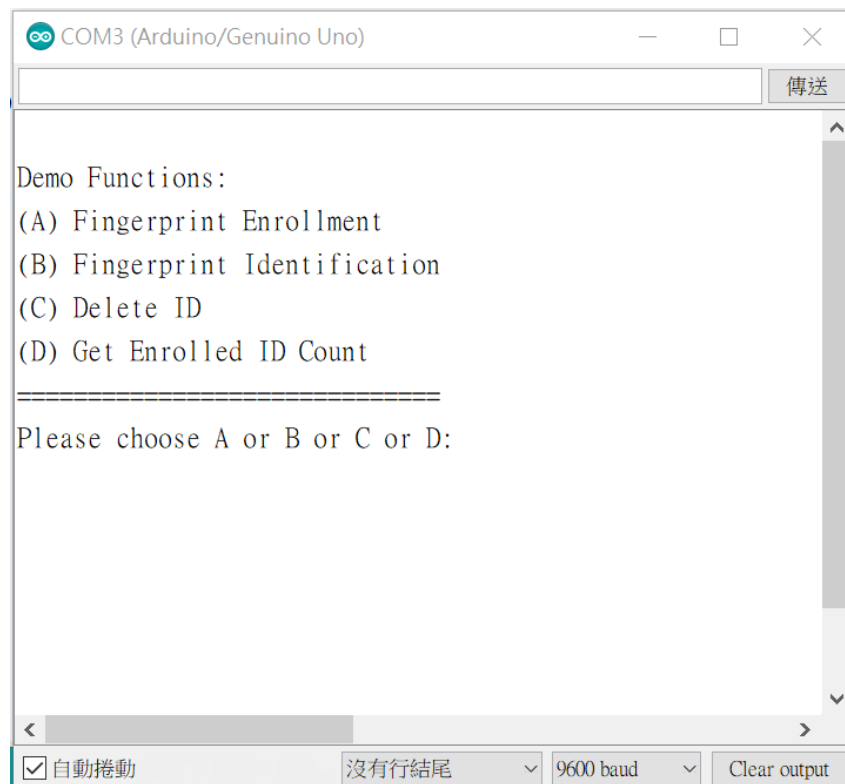
	4	VCC	Connect to Arduino UNO board 5V pin
Touch IC	1	VDD	Connect to Arduino UNO board 3.3V pin
	2	GND	Connect to Arduino UNO board GND
	3	ICPDA	Not used
	4	ICPCK	Connect to Arduino UNO board pin 6

Download and Execute Example Code

The open source community on the internet has Arduino open source example codes [\[link\]](#) for this series of fingerprint sensor module. You may refer to it. We develop an example code for GTM-5210F52 (FPS_GTM5210F52_demo) based on the open source program. Please see the product page's DOCUMENTS section. After downloading the demo code, add it to the examples directory of the open source example codes.

Launch the Arduino IDE, compile and upload the example code, open the Tools → Serial Monitor. The initial display is as shown below.

This example code demonstrates 4 functions: fingerprint enrollment, fingerprint identification, delete ID and Get enrolled ID count.



Enroll

Input A to execute the fingerprint enrollment function. The example code searches the first non-used ID starting from 0 and assigns this ID as the ID number of this enrollment.

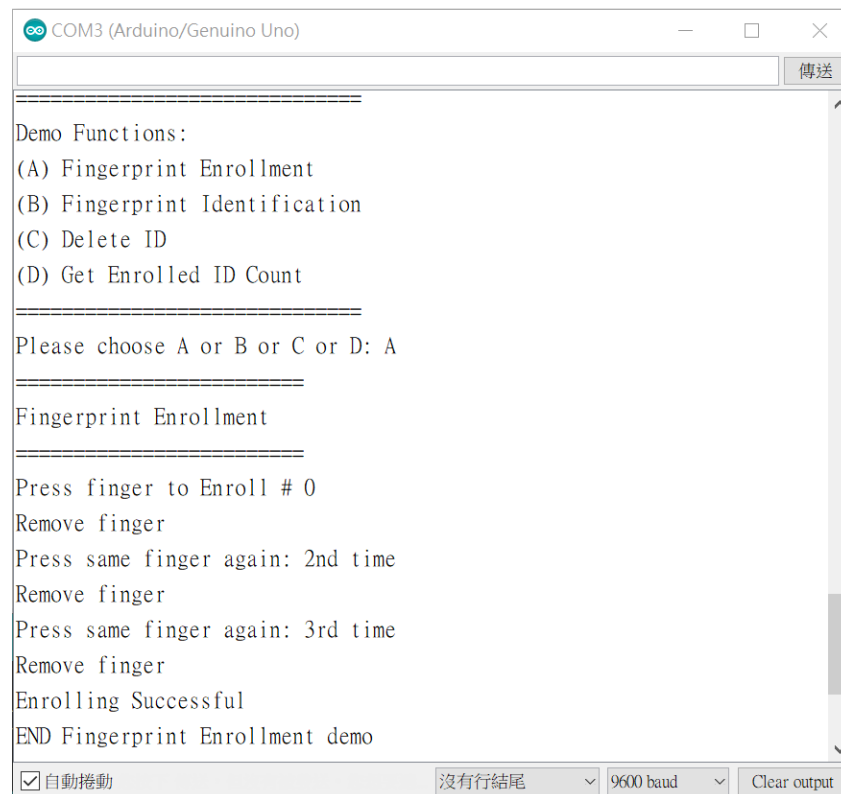
For each enrollment, the module will capture the fingerprint three times.

If the enrollment is successful, the module generates a template (498 bytes) for this fingerprint. However this demo code does not access any template.

The enrollment function is mainly implemented by the “CaptureFinger” and the “Enrollx” (x=1,2, or 3) commands of the fingerprint sensor module.

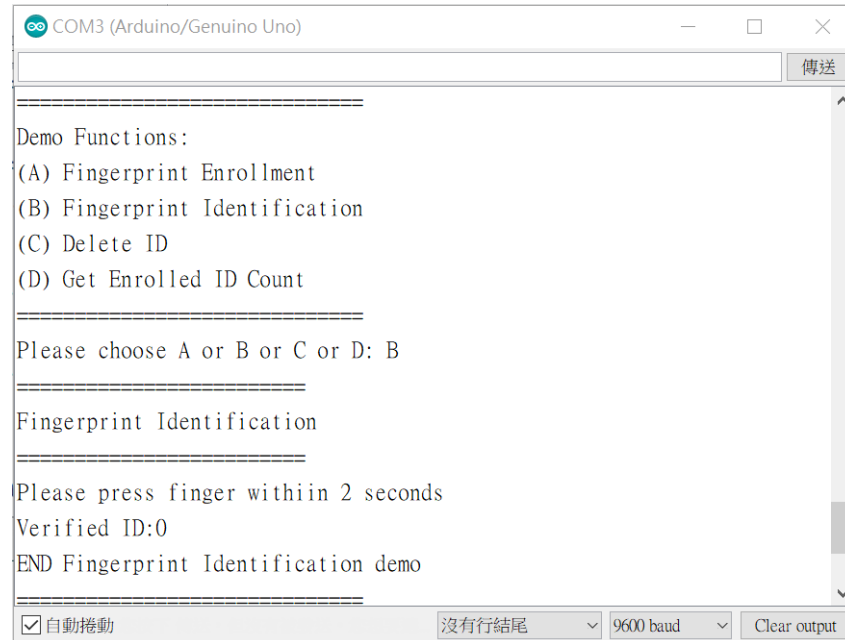
The “CaptureFinger” command is to capture fingerprint. If the finger is not pressed when executing this command, the error message is “Fail to capture x finger” (x is first, second, or third; according to which capture of an enrollment).

The “Enrollx” commands is to make template file. If error occurred at executing this command, the error message is “Enrolling Failed with error code: x”. Both “error code: 1” and “error code: 2” mean the same, failure to make template. Possible reason is the fingerprint not clear enough. You need to press finger tightly. Message “error code: 3” means the finger has been enrolled; there is already a copy in the module’s database. You may use “Delete ID” to erase this fingerprint.



Identify

Input B to execute fingerprint identification. If the identification is successful, the ID of this fingerprint will be shown.



```
COM3 (Arduino/Genuino Uno)
=====
Demo Functions:
(A) Fingerprint Enrollment
(B) Fingerprint Identification
(C) Delete ID
(D) Get Enrolled ID Count
=====
Please choose A or B or C or D: B
=====
Fingerprint Identification
=====
Please press finger within 2 seconds
Verified ID:0
END Fingerprint Identification demo
=====
 自動捲動  沒有行結尾  9600 baud  Clear output
```

After the LEDs turn on, the demo program waits 2 seconds to capture fingerprint. You may modify the program codes to adjust the waiting time.

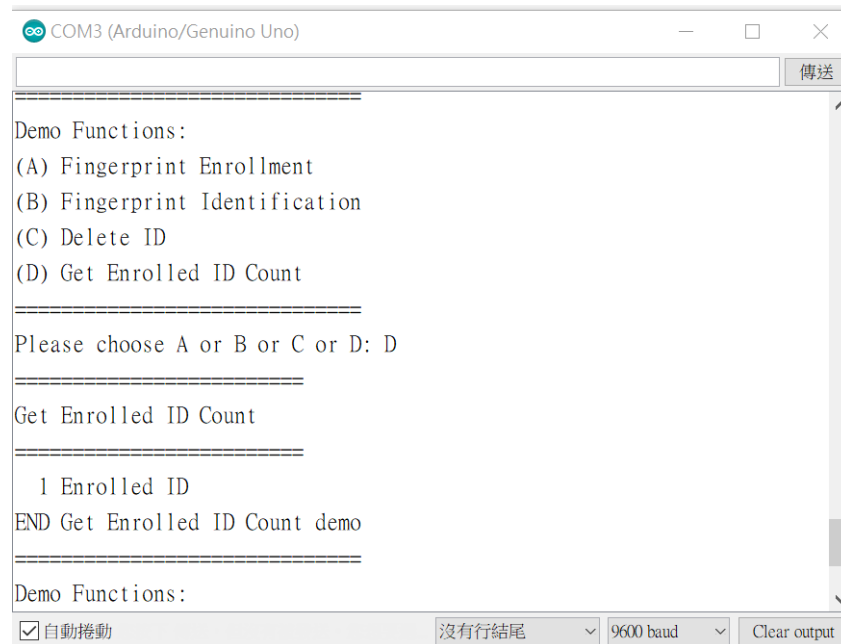
The identification function is mainly implemented by the “CaptureFinger” and the “identify” commands.

If the “CaptureFinger” command returns an error, the displayed error message is “Please press finger”, indicating there is no finger pressed when the module captures finger.

If the “identify” command returns an error, the error message will be “Finger not Found”, meaning that the module does not find this fingerprint in the database.

Get Enrolled ID Count

Input D to get the enrolled ID count.



```
COM3 (Arduino/Genuino Uno)
Demo Functions:
(A) Fingerprint Enrollment
(B) Fingerprint Identification
(C) Delete ID
(D) Get Enrolled ID Count
=====
Please choose A or B or C or D: D
=====
Get Enrolled ID Count
=====
  1 Enrolled ID
END Get Enrolled ID Count demo
=====
Demo Functions:
```

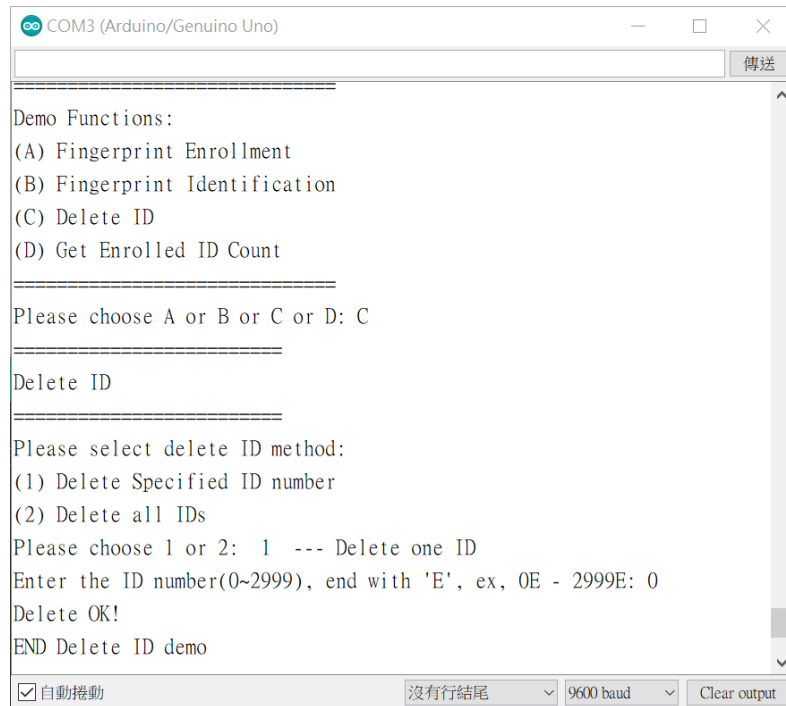
Serial monitor controls: 自動捲動 | 沒有行結尾 | 9600 baud | Clear output

The “GetEnrollCount” command is used to get the total enrolled ID number of the fingerprint sensor module.

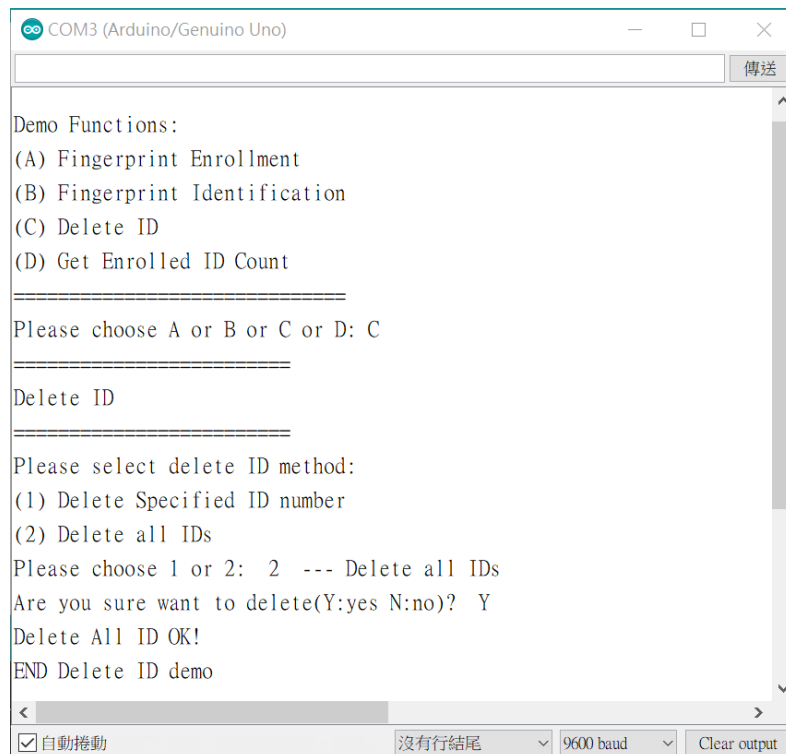
Delete ID

Input C to delete the ID from the module. There are two options, delete one ID or delete all IDs. When deleting one ID, the ID number should be specified. Note to add an E after the number due to the example code design.

The module’s command “DeleteID” and “DeleteAll” are used to implemented this function.



Delete all IDs:

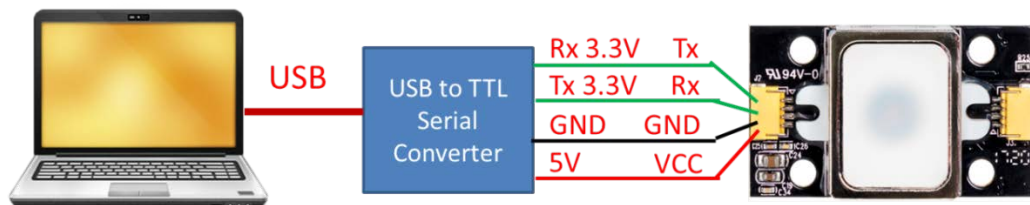


Using the SDK_DEMO Software

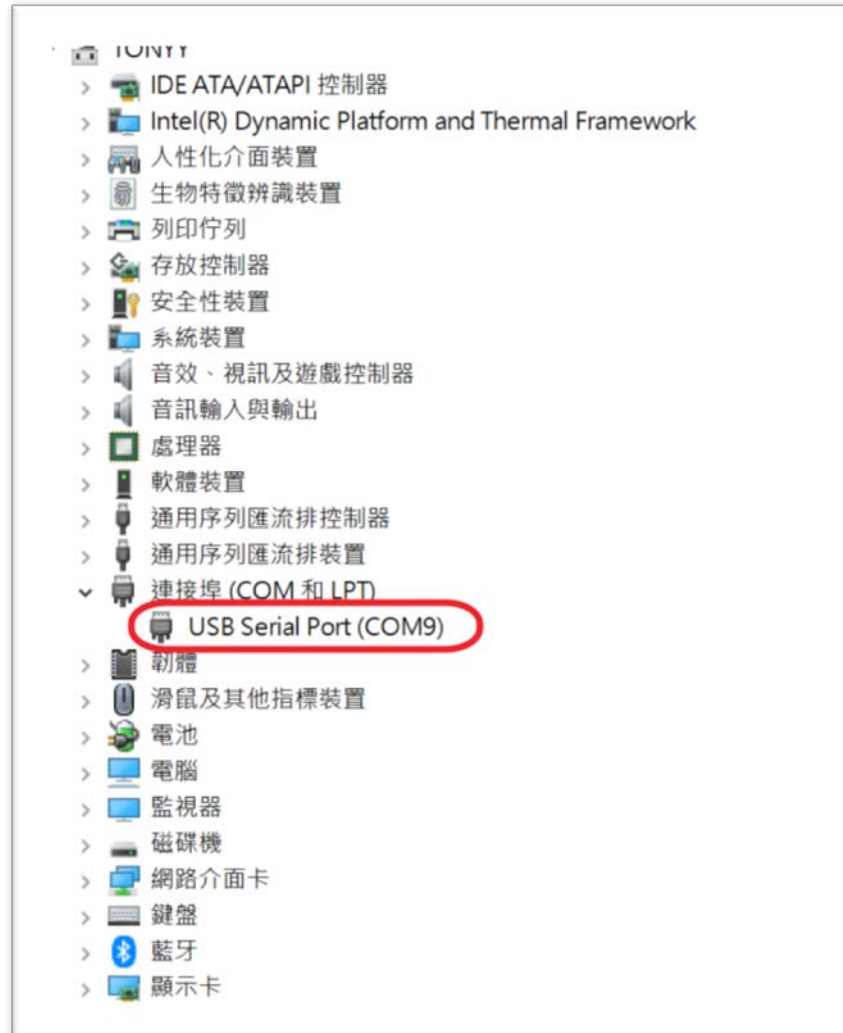
The SDK_DEMO is executable software running on a computer to demonstrate basic operations of the GTM-5210F52 fingerprint sensor module. With the SDK_DEMO it is not necessary to write program code to begin using this module, it is only necessary to make the correct wire connection to the module. The SDK_DEMO software is in the product page's DOCUMENTS section.

Wire Connection

We use an USB to TTL Serial converter to connect the fingerprint sensor module. Set the converter to work at 3.3V, and connect the individual 5V pin to the fingerprint module UART connector pin 4 (VCC). If the converter does not have enough power to drive the fingerprint sensor module, connect the VCC pin to another power source (3.3V to 6V DC)

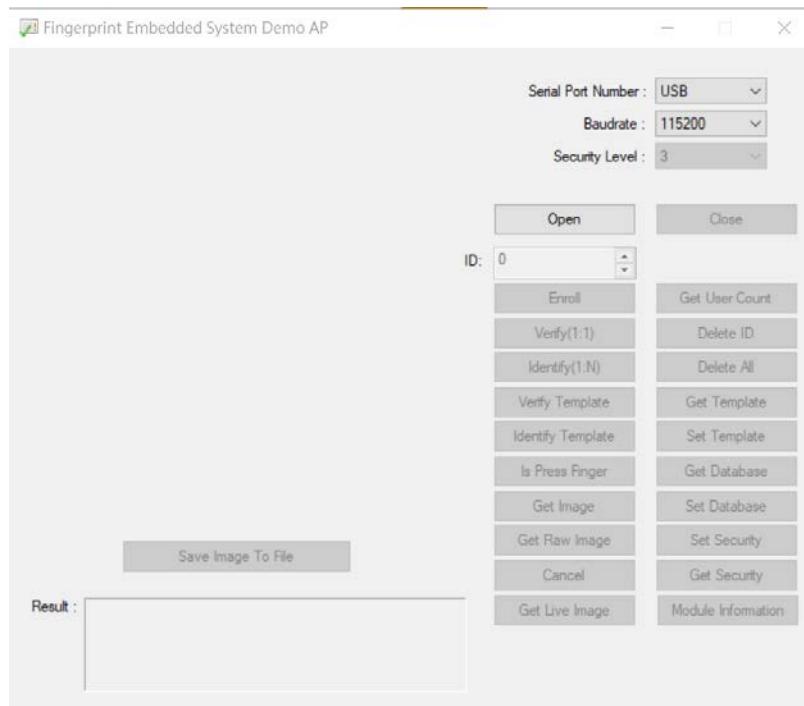


After the wire connections are complete, in the device manager of a windows computer, the USB to TTL serial converter can be viewed. Note that the COM port number is not fixed as it will change according to the computer configuration.

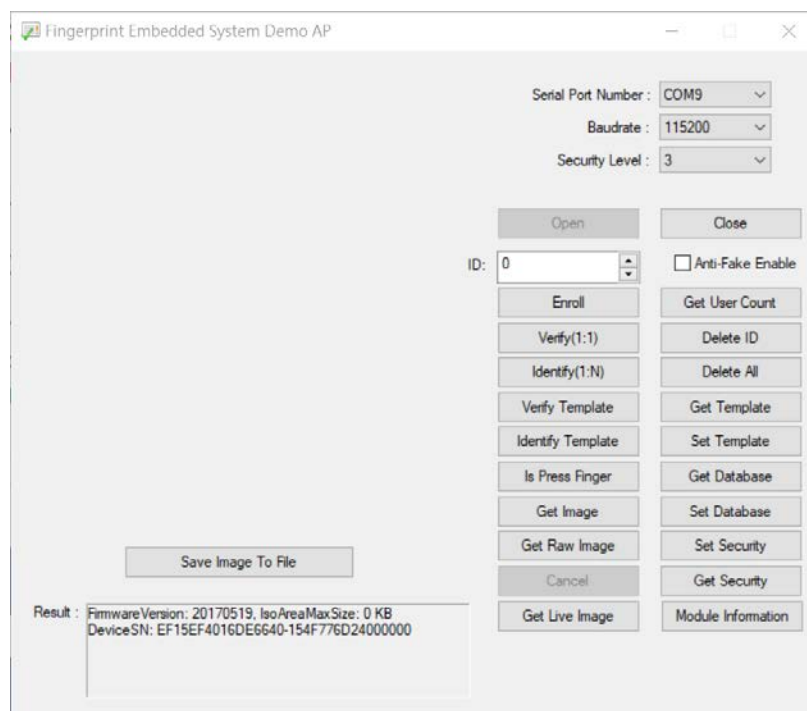


Download and Execute SDK_DEMO

Execute the SDK_DEMO software, the initial display is as shown below.

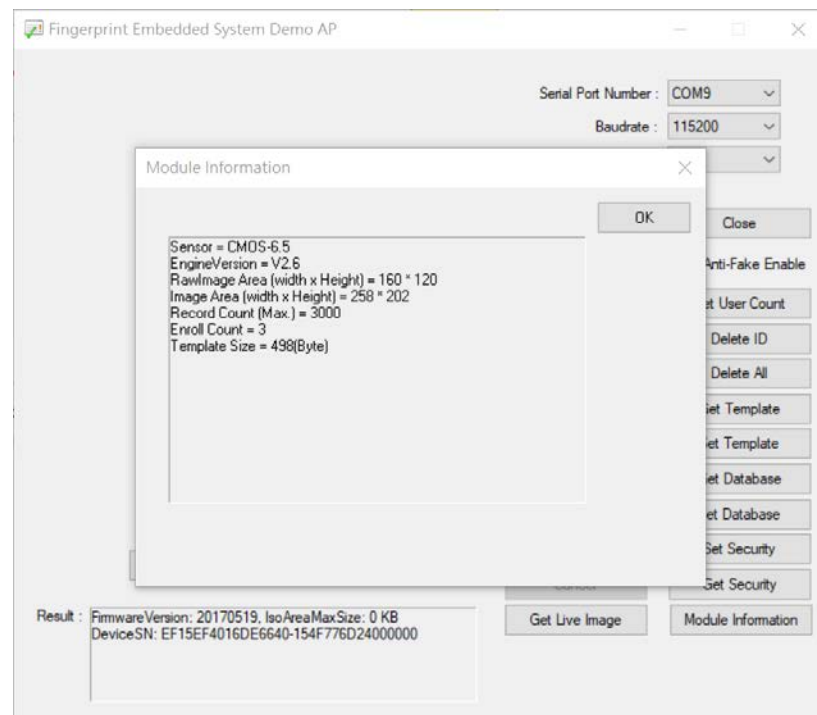


After changing the 'Serial Port Number' to COM9 and then click 'Open' to initialize the module, the display is as shown below.



View Module Information

'Module Information' can be selected to view more detailed module information.



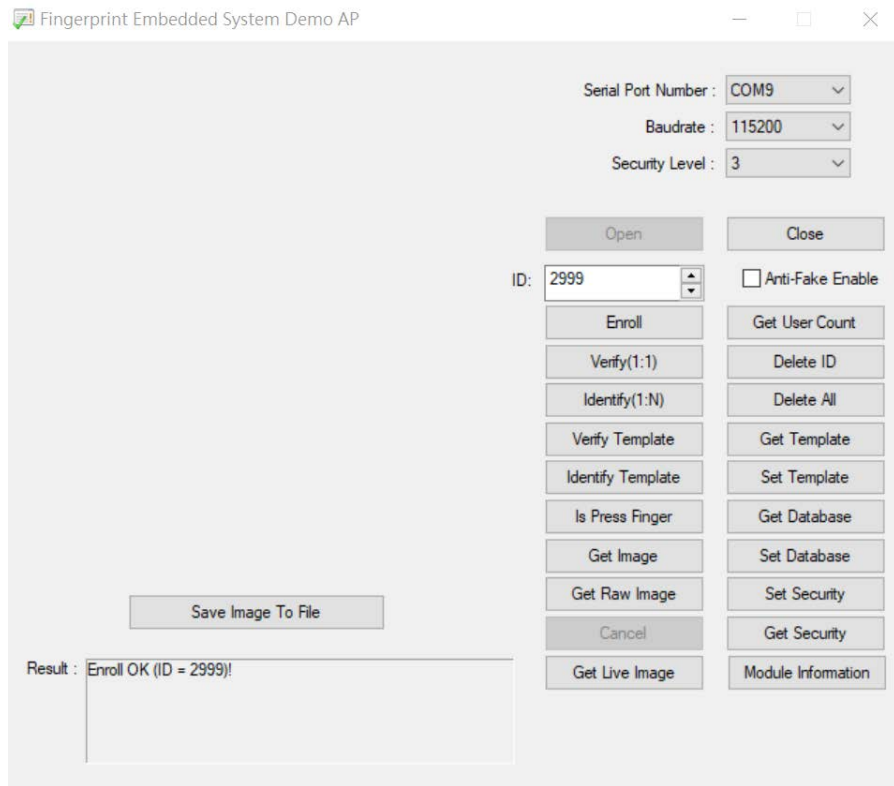
Enroll

Click 'Enroll' to enroll a fingerprint. Before clicking, the value in the 'ID' field can be changed to specify the enrollment ID number. The capacity of the module is 3000 fingerprints and the ID number is from 0 to 2999. Here we use ID 2999 as a demo.

After click 'Enroll', the LEDs inside the module will turn on and the message in the 'Result:' field shows "input finger 1 !". Now press a finger onto the fingerprint sensor plate. A successful enrollment needs to scan the fingerprint for three times. Follow the instructions shown in 'Result:' to complete the enrolling process.

When pressing the finger, it is recommended to press firmly to ensure a successful fingerprint capture. Pressing the finger loosely can result in a failed enrollment.

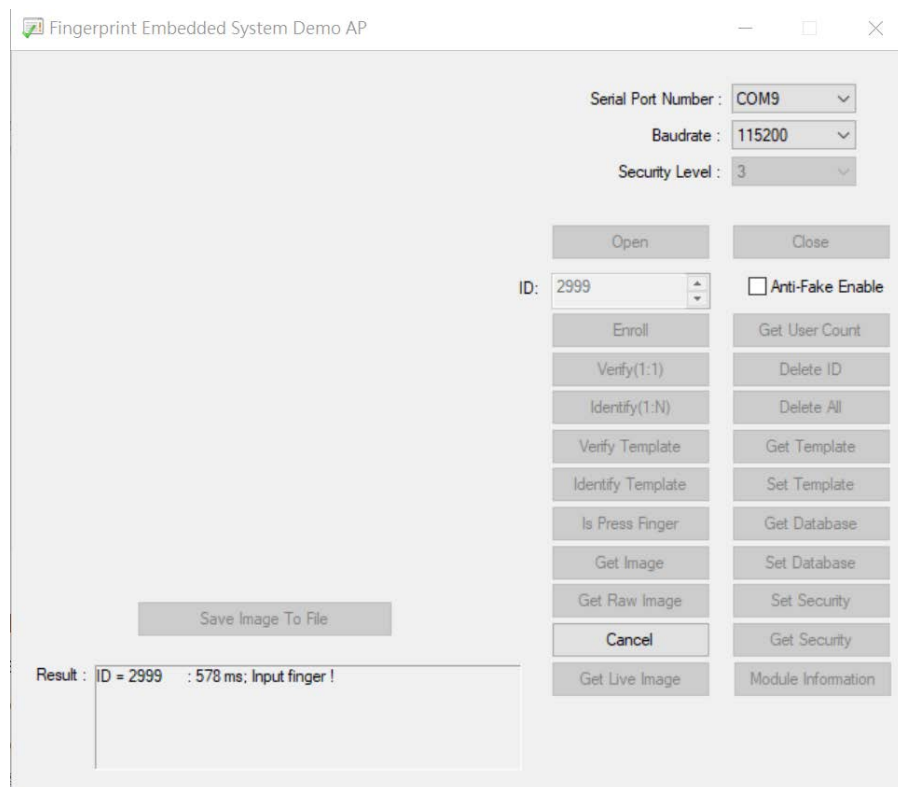
As this module is capable of 360° recognition, the finger may be pressed in any direction.



Verify (1:1)

The function of 'Verify(1:1)' is to verify that the ID number specified in the 'ID' field is the same fingerprint as the finger pressed.

After clicking 'Verify(1:1)', the LEDs turn on, if the finger is correct, the 'Result:' will show the ID number and the time used for this verification.



Identify (1:N)

The function of 'Identify(1:N)' is to identify that the ID number of the pressed finger. It will compare the captured fingerprint with the templates in the database to locate the ID number. If the finger has not been enrolled, the message is "NG".

