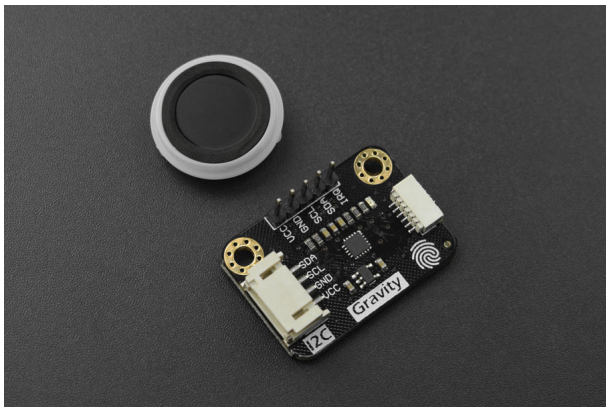
SKU:SEN0359 (https://www.dfrobot.com/product-2165.html)



(https://www.dfrobot.com/product-2165.html)

Introduction

This is a capacitive fingerprint sensor with fingerprint collecting, processing, storing, and comparison integrated all-in-one. Taking ID809 high-performance processor and semiconductor fingerprint sensor as the core, the sensor is equipped with built-in IDfinger6.0 fingerprint algorithms that enable it to complete all the fingerprint recognition independently. When working with our Arduino Library, it is able to realize functions like fingerprint registration, fingerprint deletion, etc. Also, we provide upper computer software that helps users to operate this sensor.
This fingerprint sensor comes with round breathing LEDs and has a simple structure, small size, and delicate appearance. The sensor offers fast recognition speed and high security. What's more, it supports 360-degree arbitrary angle recognition and deep self-learning function, high performance and low power consumption. Different from the SEN0348 Capacitive Fingerprint Sensor/Scanner (https://www.dfrobot.com/product-2051.html), this module adopts Gravity interfaces that make it more easier to connect. In addition, it is 3.3V and 5V compliant, and supports I2C communication, which is very suitable for working with controllers like UNO, micro:bit, and so on.

Features

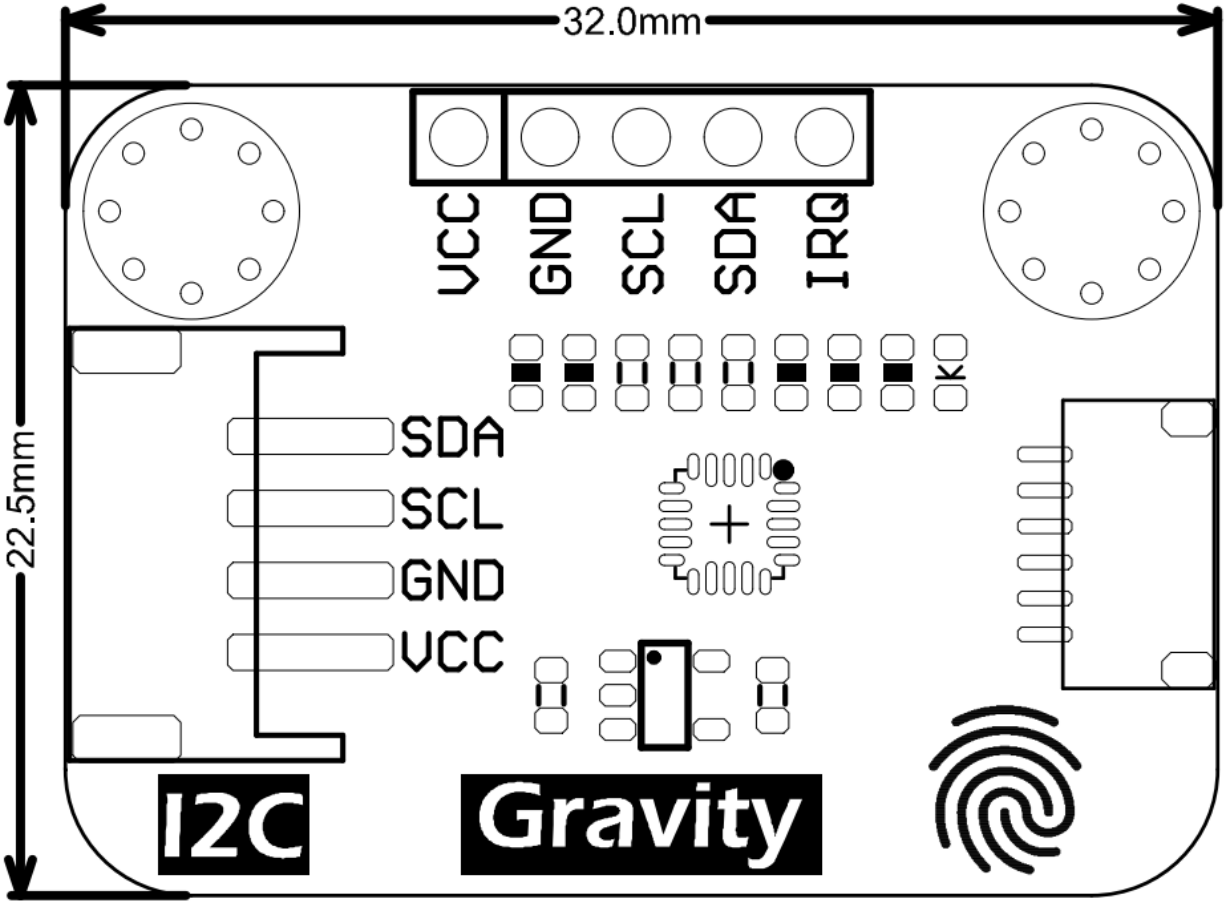
Features

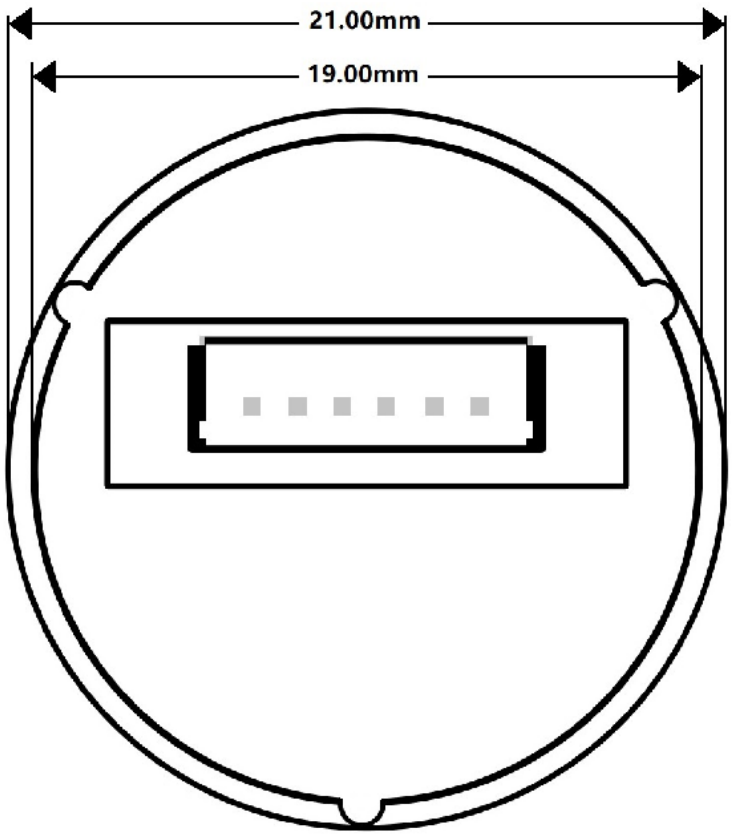
- 360-degree fingerprint entry and matching
- Self-learning function
- CNC metal ring, plus aperture

Application

- Fingerprint door-lock
- Drawer Lock
- Identity Recognition
- Authorization

Specification



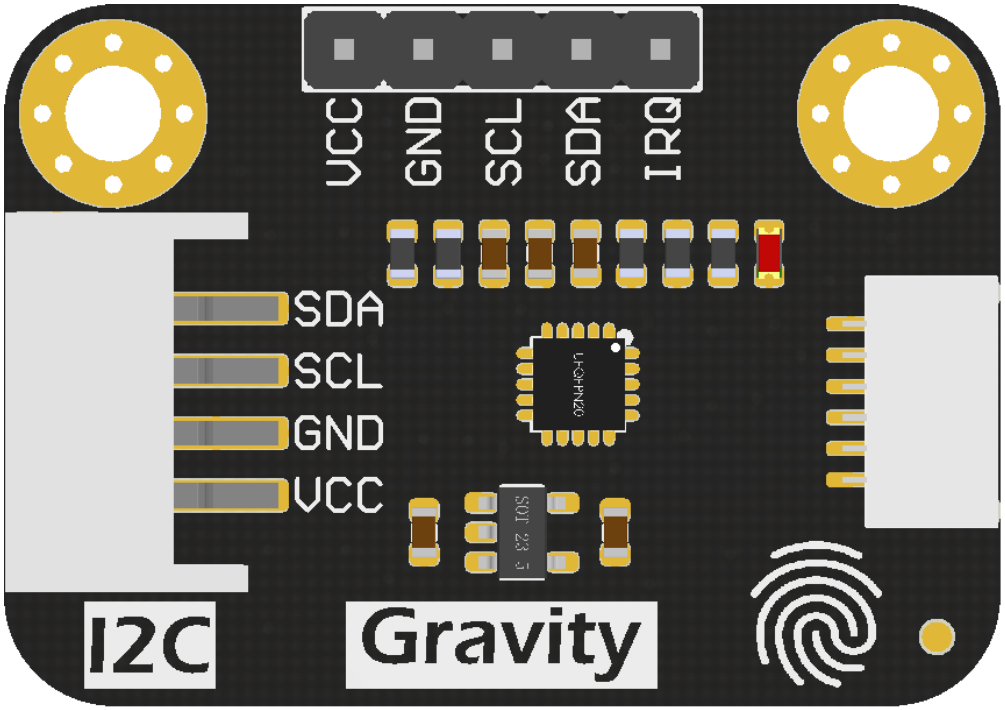


- Operating Voltage: 3.3V-5V
- Operating Current: <60mA
- Communication: I2C
- Storage Capacity: 80 fingerprints
- 1 : 1 verification time: 300~400ms

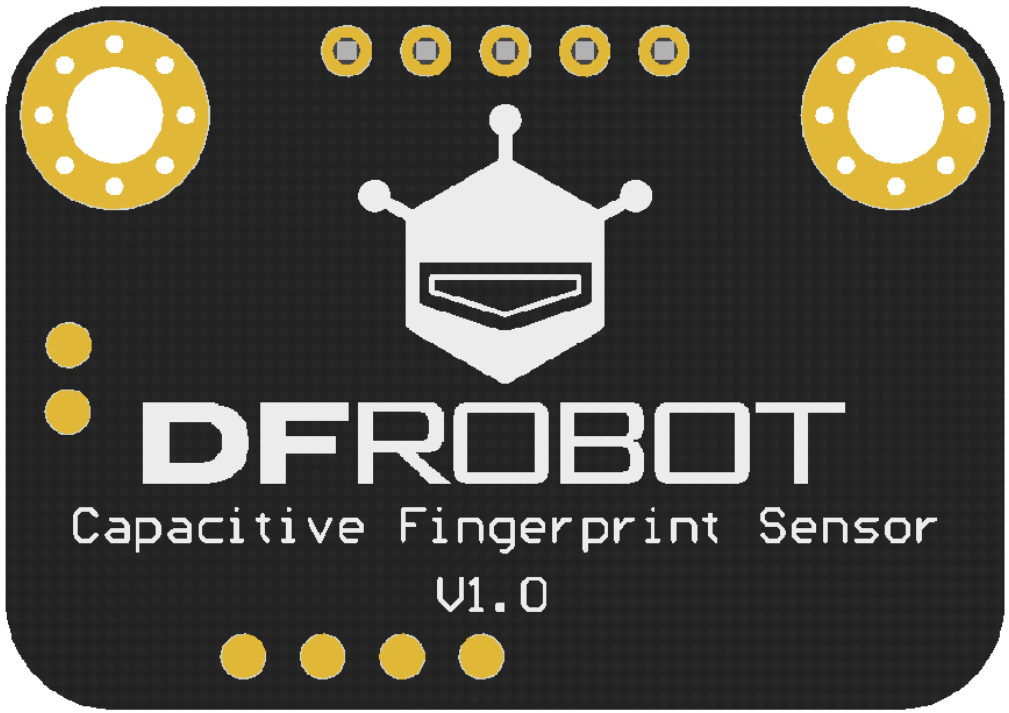
- Pixel Resolution: 508dpi
- Number of Pixels: 160x160
- Fingerprint Detection Area: 8.0mm x 8.0mm

- Working Environment: -40-60℃/<RH 90%
- Dimension: diameter 21mm/height 5mm
- Adapter Dimension: 22.5x32mm/0.89x1.26”

Board Overview



TOP



BOTTOM

Num	Label	Description

Num	Label	Description
1	VCC	+
2	GND	Ground
3	SCL	I2C Clock line
4	SDA	I2C Data line
5	IRQ	Finger sensing output: active High

Tutorial

Requirements

- **Hardware**
  - DFRduino UNO R3 (https://www.dfrobot.com/product-838.html) (or similar) x 1
  - Gravity: Capacitive Fingerprint Scanner/Sensor x1
  - Jumper wires
- **Software**
  - Arduino IDE (https://www.arduino.cc/en/Main/Software)
  - Download and install the **ID809 Library** (https://codeload.github.com/DFRobot/DFRobot_ID809_I2C/zip/master) (About how to install the library? (https://www.arduino.cc/en/Guide/Libraries#.UxU8mdzF9H0))

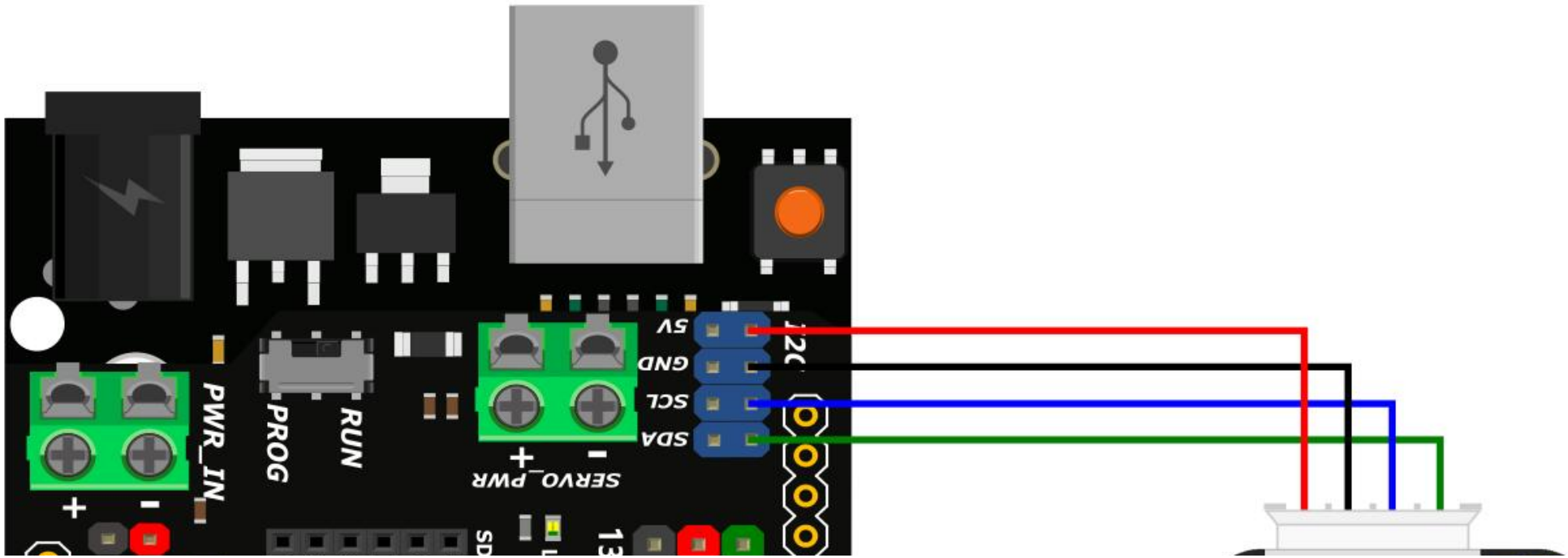
API Function List

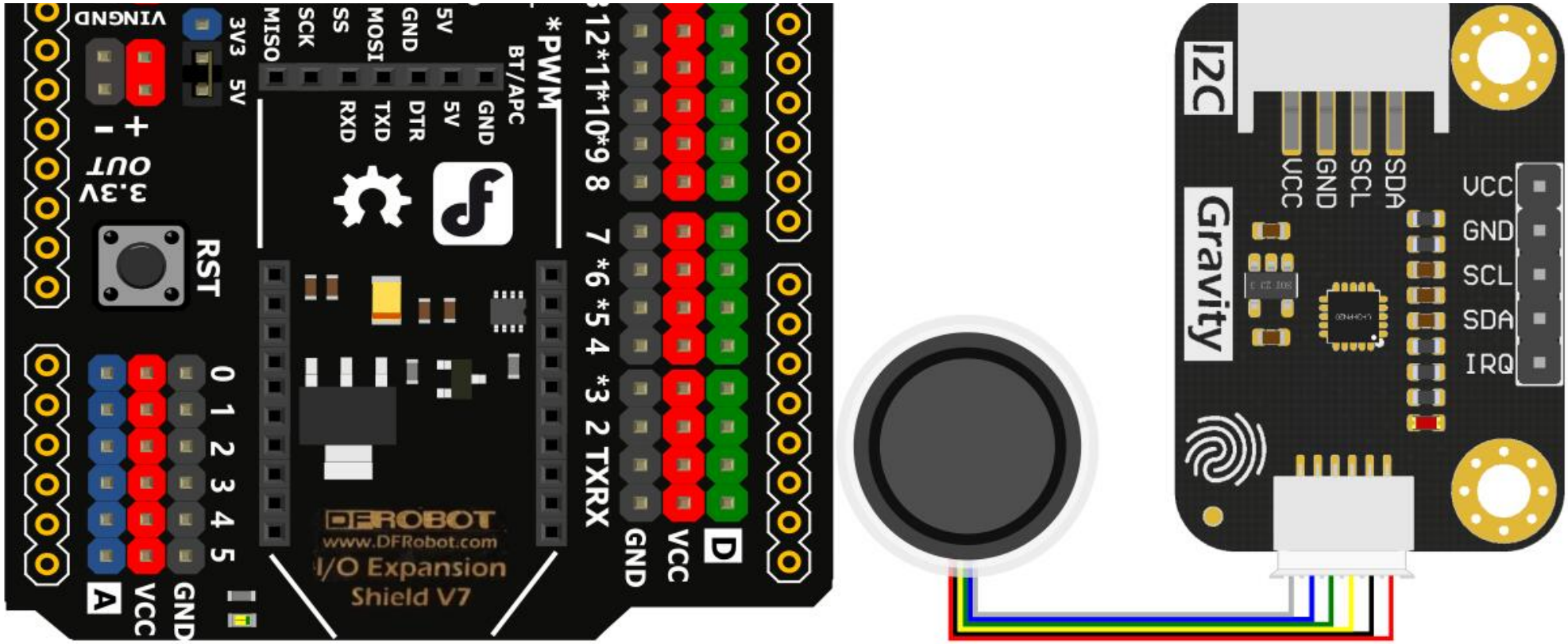
```
/**
   * @brief Test whether the module is properly connected
   * @return true or false
   */
  bool isConnected();

  /**
   * @brief Set LED light
   * @param mode:in typedef enum eLED_MODE_t
   * @param color:in typedef enum eLED_COLOR_t
   * @param blink Count 0 represents keeping breathing, blinking, this parameter is only valid in mode eBreathing, eFastBlink, eSlowBl
   * @return 0(succeed) or ERR_ID809
   */
  uint8_t ctrlLED(eLEDMode_t mode,eLEDColor_t color,uint8_t blinkCount);

  /**
   * @brief Detect whether the module is touched by a finger
   * @return 1(Yes) or 0(No)
   */
  uint8_t detectFinger();

  /**
   * @brief Get first registrable ID number
   * @return Registrable ID number or Error Code
   */
  uint8_t getEmptyID();

  /**
   * @brief Check whether the ID has been registered
   * @return 0(Registered), 1(Not yet) or ERR_ID809
   */
  uint8_t getStatusID(uint8_t ID);
```

```
uint8_t getStatusID(uint8_t ID);

/**
 * @brief Get the number of registered users
 * @return Number of registered users or ERR_ID809

 */
uint8_t getEnrollCount();

/**
 * @brief Get the list of registered users
 * @return 0(succeed) or ERR_ID809
 */
 uint8_t getEnrolledIDList(uint8_t* list);

/**
 * @brief Capture fingerprint
 * @return 0(succeed) or ERR_ID809
 */
uint8_t collectionFingerprint(uint16_t timeout);

/**
 * @brief Save fingerprint
 * @param Fingerprint ID
 * @return 0(succeed) or ERR_ID809
 */
uint8_t storeFingerprint(uint8_t ID);

/**
 * @brief Delete fingerprint
 * @param Finerprint ID or DELALL(delete all)
 * @return 0(succeed) or ERR_ID809
 */
uint8_t delFingerprint(uint8_t ID);

/**
 * @brief Match the fingerprint with all fingerprints
 * @return Successfully matched fingerprint ID, 0(Finerprint matching failed) or ERR_ID809
```

```
 */
uint8_t search();

/**
 * @brief Match the fingerprint with a designated fingerprint
 * @return Successfully matched fingerprint ID, 0(Finerprint matching failed) or ERR_ID809
 */
uint8_t verify(uint8_t ID);

/**
 * @brief Get error information
 * @return Text description of error information
 */
String getErrorDescription();
```

Connection Diagram





Sample Code 1 - Get Module Information

Get and serial print the module information.

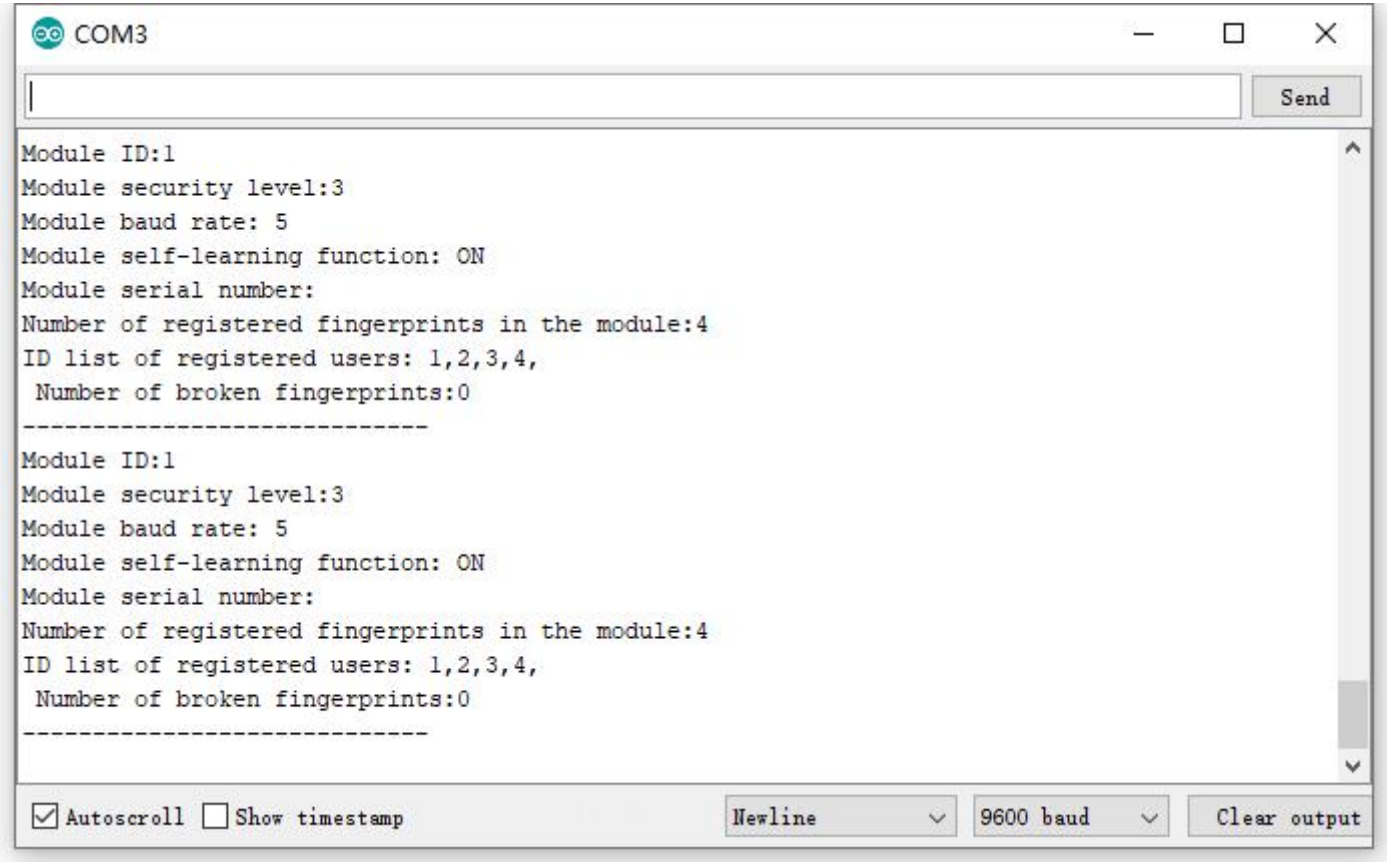
```
/*!
 * @file getDeviceInformation.ino
 * @brief Get fingerprint module information
 * @n Experiment Phenomenon: serial print module ID, security level, baud rate, etc.
 * @copyright  Copyright (c) 2010 DFRobot Co.Ltd (http://www.dfrobot.com)
 * @licence     The MIT License (MIT)
 * @author [Eddard](Eddard.liu@dfrobot.com)
 * @version  V1.0
 * @date  2020-03-19
 * @get from https://www.dfrobot.com
 * @url https://github.com/cdjq/DFRobot_ID809
*/

#include <DFRobot_ID809.h>

DFRobot_ID809_IIC fingerprint;
//DFRobot_ID809_UART fingerprint(115200);
//String desc;

void setup(){
  /*Init print serial port */
  Serial.begin(9600);
  /*Take FPSerial as communication port of the module*/
  fingerprint.begin();
  /*Wait for Serial to open*/
  while(!Serial);
  /*Test whether device can communicate properly with mainboard
    Return true or false
    */
  while(fingerprint.isConnected() == false){
    Serial.println("Communication with device failed, please check connection");
    /*Get error code information */
```

```
    /*Get error code information */
    //desc = fingerprint.getErrorDescription();
    //Serial.println(desc);
    delay(1000);
  }

}

uint8_t enrollCount;   //Number of registered users

void loop(){
  /*Set module ID, available range: 1-255*/
  //fingerprint.setDeviceID(/*Device ID = */1);
  Serial.print("Module ID:");
  /*Read module ID*/
  Serial.println(fingerprint.getDeviceID());

  /*Set module security level, range 1-5. Default level: 3
    Security Level          Recognition rate
      Level 1           FAR  0.1%
                        FRR  0.005%
      Level 2           FAR  0.003%
                        FRR  0 01%
      Level 3           FAR  0.001%
                        FRR  0.1%
      Level 4           FAR  0.003%
                        FRR  0.5%
      Level 5           FAR  0.0001%
                        FRR  1%
   */
  //fingerprint.setSecurityLevel(/*Security Level = */3);
  Serial.print("Module security level:");
  /*Read module security level*/
  Serial.println(fingerprint.getSecurityLevel());

  /*Set module baud rate, available range:
    e9600bps    e19200bps   e38400bps   e57600bps    e115200bps
       1           2            3          4             5
```

```
 */
//fingerprint.setBaudrate(fingerprint.e115200bps);
Serial.print("Module baud rate: ");
/*Read module baud rate*/
Serial.println(fingerprint.getBaudrate());


/*Set module self-learning function, 1(ON) 0(OFF)*/
//fingerprint.setAutoLearn(/*Auto Learn = */1);
Serial.print("Module self-learning function: ");
/*Read the state of module self-learning function
  Print ON if it is enabled, otherwise print OFF */
Serial.println(fingerprint.getSelfLearn()?"ON":"OFF");

/*Set module serial number, the number of characters of serial number should be small than 15 */
//fingerprint.setModuleSN(/*Module SN = */"DFRobot");
Serial.print("Module serial number:");
/*Read module serial number */
Serial.println(fingerprint.getModuleSN());

Serial.print("Number of registered fingerprints in the module:");
/*Get the number of registered users */
Serial.println(enrollCount = fingerprint.getEnrollCount());
/*Declare an array to hold ID list */
uint8_t list[80] = {0};
/*Get user list
 */
fingerprint.getEnrolledIDList(list);
Serial.print("ID list of registered users: ");
for(uint8_t i = 0; i < enrollCount; i++){
  Serial.print(list[i]);
  Serial.print(",");
}

Serial.print("\nNumber of broken fingerprints:");
/*Get the number of broken fingerprints */
Serial.println(fingerprint.getBrokenQuantity());
/*Get the ID of the first broken fingerprint*/
```

```
  //fingerprint.getBrokenID();
  Serial.println("------------------------------");

  delay(1000);
}
```

- Result



Sample Code 2 - Add Fingerprints

This sample automatically obtains the registrable ID, then collects the fingerprint three times, the yellow light flashes three times successfully. At last, save the fingerprint to the acquired unregistered ID, the green light is on for 1s, and then turns off.

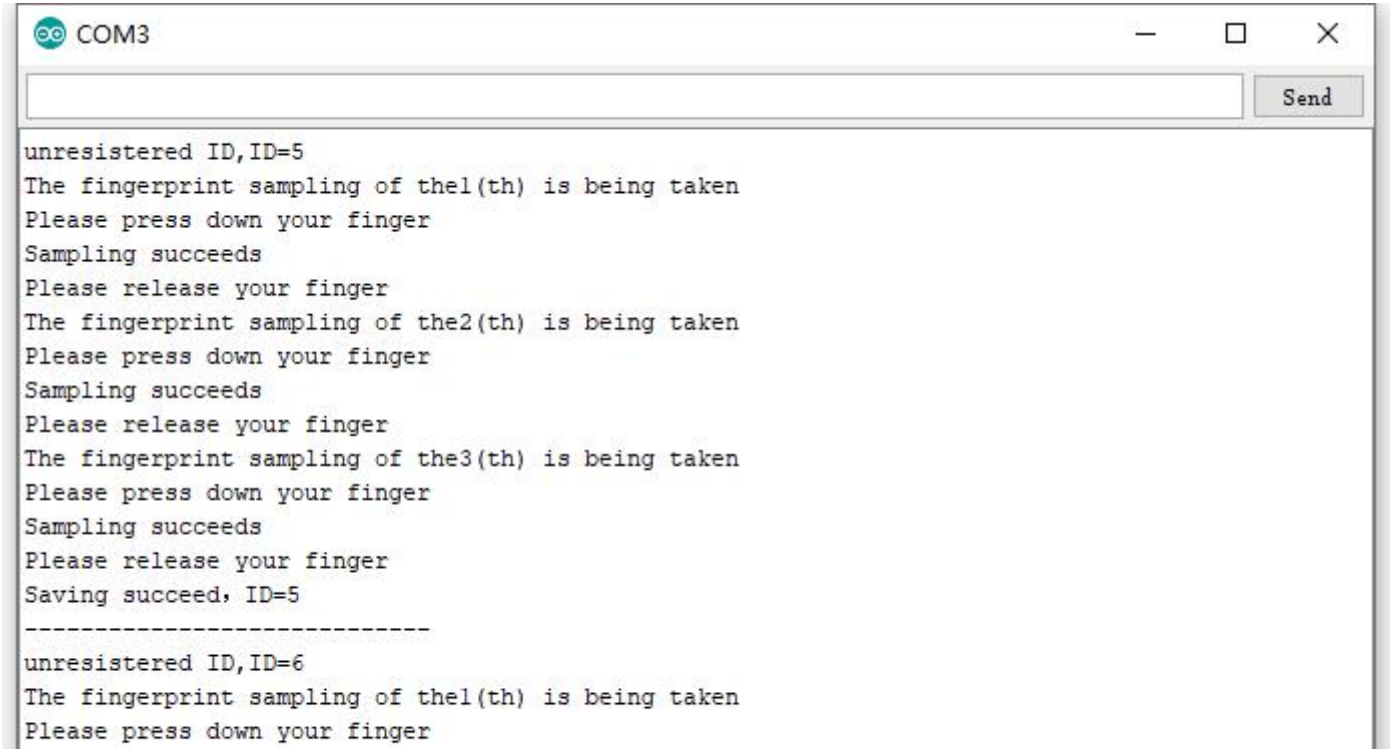
```
/*!
 * @file fingerprintRegistration.ino
 * @brief Fingerprint Acquisition and Saving
 * @n This module can be controlled by hardware serial or software serial
 * @n Experiment Phenomenon: auto retrieve unregistered ID, collect fingerprint 3 times.
 * @n           In collecting, set LED ring to breathing lighting in blue, and then to quick blink in yellow 3 times when completed
 * @n           At last, save the fingerprint in an unregistered ID, the green LED lights up for 1s and turns off.
 * @copyright  Copyright (c) 2010 DFRobot Co.Ltd (http://www.dfrobot.com)
 * @licence     The MIT License (MIT)
 * @author [Eddard](Eddard.liu@dfrobot.com)
 * @version  V1.0
 * @date  2020-03-19
 * @get from https://www.dfrobot.com
 * @url https://github.com/cdjq/DFRobot_ID809
*/

#include <DFRobot_ID809.h>

#define COLLECT_NUMBER 3  //Fingerprint sampling times, can be set to 2-3

DFRobot_ID809_IIC fingerprint;
//DFRobot_ID809_UART fingerprint(115200);
//String desc;

void setup(){
  /*Init print serial port */
  Serial.begin(9600);
  /*Take FPSerial as communication port of the module*/
  fingerprint.begin();
  /*Wait for Serial to open*/
  while(!Serial);
  /*Test whether the device can communicate properly with mainboard
```

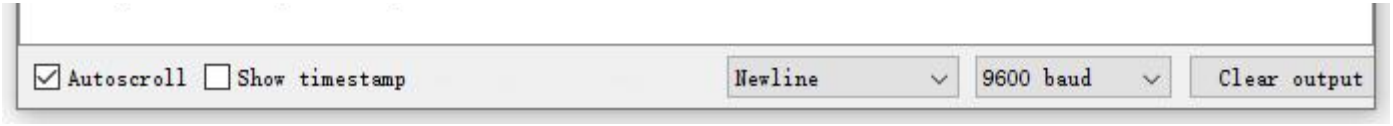
```
  /*Test whether the device can communicate properly with mainboard
    Return true or false
    */
  while(fingerprint.isConnected() == false){
    Serial.println("Communication with device failed, please check connection");

    /*Get error code information*/
    //desc = fingerprint.getErrorDescription();
    //Serial.println(desc);
    delay(1000);
  }
}

uint8_t ID,i,ret;

void loop(){
  /*Get an unregistered ID for saving fingerprint
    Return ID when succeeded
    Return ERR_ID809 if failed
   */
  if((ID = fingerprint.getEmptyID()) == ERR_ID809){
    while(1){
      /*Get error code information*/
      //desc = fingerprint.getErrorDescription();
      //Serial.println(desc);
      delay(1000);
    }
  }
  Serial.print("unresistered ID,ID=");
  Serial.println(ID);
  i = 0;   //Clear sampling times
  /*Fingerprint sampling 3 times*/
  while(i < COLLECT_NUMBER){
    /*Set fingerprint LED ring mode, color, and number of blinks
      Can be set as follows:
      Parameter 1:<LEDMode>
      eBreathing   eFastBlink   eKeepsOn    eNormalClose
      eFadeIn      eFadeOut     eSlowBlink
```

```
    Parameter 2:<LEDColor>
    eLEDGreen  eLEDRed      eLEDYellow   eLEDBlue
    eLEDCyan   eLEDMagenta  eLEDWhite
    Parameter 3:<Number of blinks> 0 represents blinking all the time
    This parameter will only be valid in mode eBreathing, eFastBlink, eSlowBlink

   */
  fingerprint.ctrlLED(/*LEDMode = */fingerprint.eBreathing, /*LEDColor = */fingerprint.eLEDBlue, /*blinkCount = */0);
  Serial.print("The fingerprint sampling of the");
  Serial.print(i+1);
  Serial.println("(th) is being taken");
  Serial.println("Please press down your finger");
  /*Capture fingerprint image, 10s idle timeout
    IF succeeded, return 0, otherwise, return ERR_ID809
   */
  if((fingerprint.collectionFingerprint(/*timeout = */10)) != ERR_ID809){
    /*Set fingerprint LED ring to quick blink in yellow 3 times */
    fingerprint.ctrlLED(/*LEDMode = */fingerprint.eFastBlink, /*LEDColor = */fingerprint.eLEDYellow, /*blinkCount = */3);
    Serial.println("Sampling succeeds");
    i++;   //Sampling times +1
  }else{
    Serial.println("Sampling failed");
    /*Get error code information*/
    //desc = fingerprint.getErrorDescription();
    //Serial.println(desc);
  }
  Serial.println("Please release your finger");
  /*Wait for finger to release
    Return 1 when finger is detected, otherwise return 0
   */
  while(fingerprint.detectFinger());
}

/*Save fingerprint in an unregistered ID */
if(fingerprint.storeFingerprint(/*Empty ID = */ID) != ERR_ID809){
  Serial.print("Saving succeed,ID=");
  Serial.println(ID);
  Serial.println("------------------------------");
```

```
    /*Set fingerprint LED ring to always ON in green */
    fingerprint.ctrlLED(/*LEDMode = */fingerprint.eKeepsOn, /*LEDColor = */fingerprint.eLEDGreen, /*blinkCount = */0);
    delay(1000);
    /*Turn off fingerprint LED ring */
    fingerprint.ctrlLED(/*LEDMode = */fingerprint.eNormalClose, /*LEDColor = */fingerprint.eLEDBlue, /*blinkCount = */0);

    delay(1000);
  }else{
    Serial.println("Saving failed");
    /*Get error code information*/
    //desc = fingerprint.getErrorDescription();
    //Serial.println(desc);
  }
}
```

- Result





Sample Code 3 - Fingerprint Matching

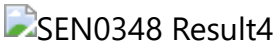
Collect the fingerprint image and compare it with the fingerprints in the fingerprint library. If the comparison succeeds, the green light will be on and the ID number will be printed. If it fails, the red light will be on and the matching failure will be prompted.

```
/*!
 * @file fingerprintMatching.ino
 * @brief Gather fingerprint and compare it with fingerprints in fingerprint library
 * @n Experiment Phenomenon: capture fingerprint image and compare it with all fingerprints in the fingerprint library
                If matched successfully, light up green LED and print ID number. Return 0 when failed.
 * @copyright  Copyright (c) 2010 DFRobot Co.Ltd (http://www.dfrobot.com)
 * @licence     The MIT License (MIT)
 * @author [Eddard](Eddard.liu@dfrobot.com)
 * @version  V1.0
 * @date  2020-03-19
 * @get from https://www.dfrobot.com
 * @url https://github.com/cdjq/DFRobot_ID809
*/
#include <DFRobot_ID809.h>

DFRobot_ID809_IIC fingerprint;
//DFRobot_ID809_UART fingerprint(115200);
//String desc;

void setup(){
  /*Init print serial port*/
  Serial.begin(9600);
  /*Take FPSerial as communication serial port of the fingerprint module*/
  fingerprint.begin();
  /*Wait for Serial to open*/
  while(!Serial);
  /*Test whether the device can properly communicate with mainboard
   Return true or false
    */
  while(fingerprint.isConnected() == false){
    Serial.println("Communication with device failed, please check connection");
    /*Get error code information*/
```

```
    /*Get error code information*/
    //desc = fingerprint.getErrorDescription();
    //Serial.println(desc);
    delay(1000);
  }

}

void loop(){
  uint8_t ret = 0;
  /*Set fingerprint LED ring mode, color, and number of blinks
    Can be set as follows:
    Parameter 1:<LEDMode>
    eBreathing   eFastBlink   eKeepsOn    eNormalClose
    eFadeIn      eFadeOut     eSlowBlink
    Parameter 2:<LEDColor>
    eLEDGreen  eLEDRed      eLEDYellow   eLEDBlue
    eLEDCyan   eLEDMagenta  eLEDWhite
    Parameter 3:<number of blinks> 0 represents blinking all the time
    This parameter will only be valid in mode eBreathing, eFastBlink, eSlowBlink
   */
  fingerprint.ctrlLED(/*LEDMode = */fingerprint.eBreathing, /*LEDColor = */fingerprint.eLEDBlue, /*blinkCount = */0);
  Serial.println("Please press down your finger");
  /*Capture fingerprint image, 10s idle timeout
    If succeed return 0, otherwise return ERR_ID809
   */
  if((fingerprint.collectionFingerprint(/*timeout=*/10)) != ERR_ID809){
    /*Set fingerprint LED ring to quick blink in yellow 3 times*/
    fingerprint.ctrlLED(/*LEDMode = */fingerprint.eFastBlink, /*LEDColor = */fingerprint.eLEDYellow, /*blinkCount = */3);
    Serial.println("Capturing succeeds");
  }else{
    Serial.println("Capturing fails");
    /*Get error code information*/
    //desc = fingerprint.getErrorDescription();
    //Serial.println(desc);
  }
  Serial.println("Please release your finger");
  /*Wait for finger to release
```

```
    Return 1 when finger is detected, otherwise return 0
   */
  while(fingerprint.detectFinger());

  /*Compare the captured fingerprint with all the fingerprints in the fingerprint library

    Return fingerprint ID(1-80) if succeed, return 0 when failed
   */
  ret = fingerprint.search();
  /*Compare the captured fingerprint with a fingerprint of specific ID
    Return fingerprint ID(1-80) if succeed, return 0 when failed
   */
  //ret = fingerprint.verify(/*Fingerprint ID = */1);
  if(ret != 0){
    /*Set fingerprint LED ring to always ON in green */
    fingerprint.ctrlLED(/*LEDMode = */fingerprint.eKeepsOn, /*LEDColor = */fingerprint.eLEDGreen, /*blinkCount = */0);
    Serial.print("Matching succeeds,ID=");
    Serial.println(ret);
  }else{
    /*Set fingerprint LED ring to always ON in red*/
    fingerprint.ctrlLED(/*LEDMode = */fingerprint.eKeepsOn, /*LEDColor = */fingerprint.eLEDRed, /*blinkCount = */0);
    Serial.println("Matching fails");
  }
  Serial.println("-----------------------------");
  delay(1000);
}
```

- Result



Sample Code 4 - Fingerprint Deletion

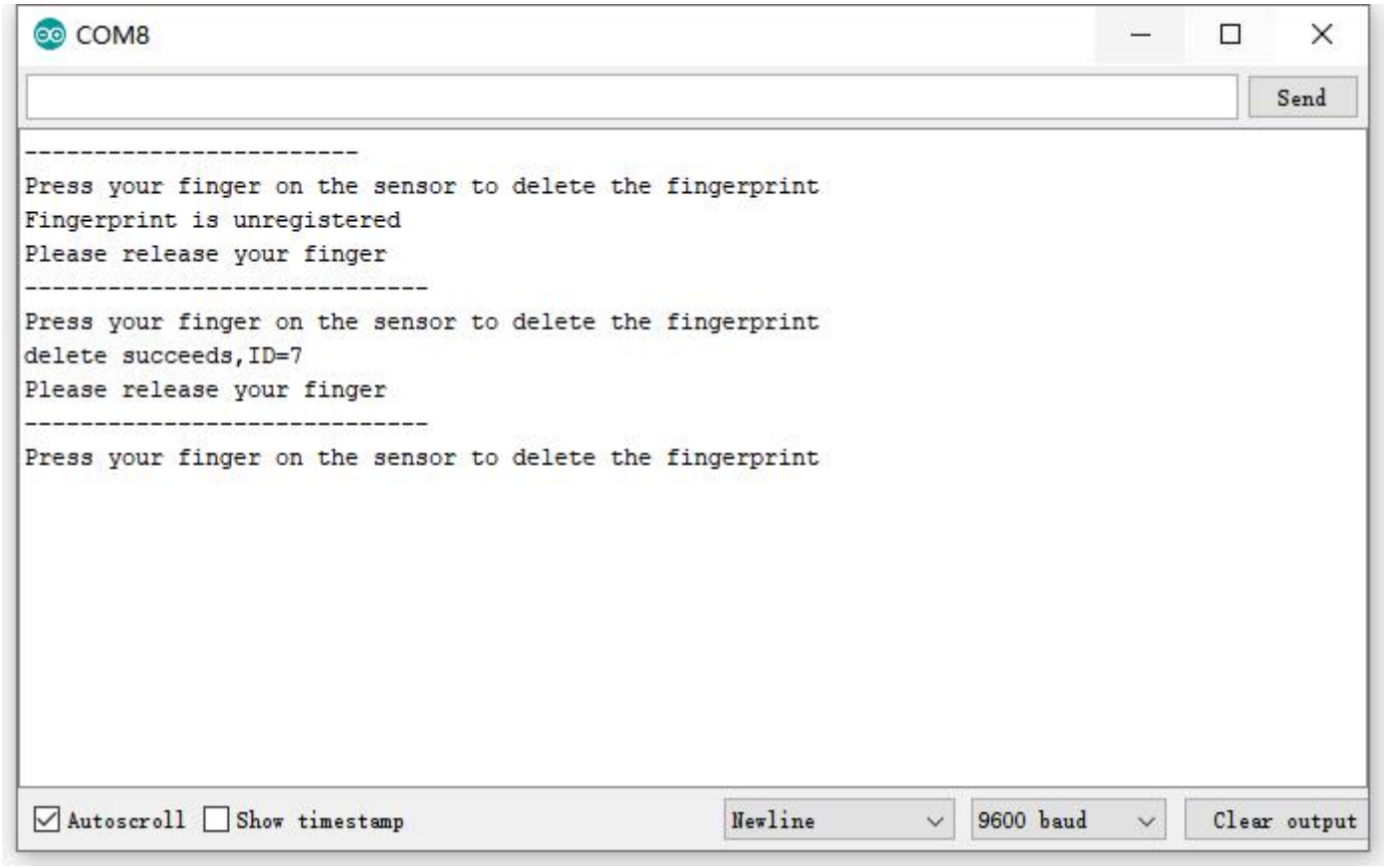
Run the codes, press your finger on the sensor to delete your fingerprint.

```
/*!
 * @file fingerprintDeletion.ino
 * @brief Delete a specific fingerprint
 * @n Experiment phenomenon: press your finger on the sensor, if this fingerprint is registered, delete it and LED turns on in green.
      If it is unregistered or fingerprint collection fails, LED light turns on in red.
 * @copyright  Copyright (c) 2010 DFRobot Co.Ltd (http://www.dfrobot.com)
 * @licence     The MIT License (MIT)
 * @author [Eddard](Eddard.liu@dfrobot.com)
 * @version  V1.0
 * @date  2020-03-19
 * @get from https://www.dfrobot.com
 * @url https://github.com/DFRobot/DFRobot_ID809
*/
#include <DFRobot_ID809.h>

DFRobot_ID809_IIC fingerprint;
//DFRobot_ID809_UART fingerprint(115200);
//String desc;

void setup(){
  /*Init print serial port*/
  Serial.begin(9600);
  /*Take FPSerial as communication serial port of the fingerprint module*/
  fingerprint.begin();
  /*Wait for Serial to open*/
  while(!Serial);
  /*Test whether the device can properly communicate with mainboard
    Return true or false
    */
  while(fingerprint.isConnected() == false){
    Serial.println("Communication with device failed, please check connection");
    /*Get error code information*/
```

```
    /*Get error code information*/
    //desc = fingerprint.getErrorDescription();
    //Serial.println(desc);
    delay(1000);
  }

}

void loop(){
  uint8_t ret = 0;
  Serial.println("Press your finger on the sensor to delete the fingerprint");
  /*Set fingerprint LED ring mode, color, and number of blinks
    Can be set as follows:
    Parameter 1:<LEDMode>
    eBreathing   eFastBlink   eKeepsOn    eNormalClose
    eFadeIn      eFadeOut     eSlowBlink
    Parameter 2:<LEDColor>
    eLEDGreen  eLEDRed      eLEDYellow   eLEDBlue
    eLEDCyan   eLEDMagenta  eLEDWhite
    Parameter 3:<Number of blinks> 0 represents blinking all the time
    This parameter will only be valid in mode eBreathing, eFastBlink, eSlowBlink
   */
  fingerprint.ctrlLED(/*LEDMode = */fingerprint.eBreathing, /*LEDColor = */fingerprint.eLEDBlue, /*blinkCount = */0);
  /*Capture fingerprint image, 10s idle timeout
    If succeed return 0, otherwise return ERR_ID809
   */
  if((fingerprint.collectionFingerprint(/*timeout=*/10)) != ERR_ID809){
    /*Compare the captured fingerprint with all the fingerprints in the fingerprint library
      Return fingerprint ID(1-80) if succeed, return 0 when failed
     */
    ret = fingerprint.search();
    if(ret != 0){
      /*Delete the fingerprint of this ID*/
      fingerprint.delFingerprint(/*Fingerprint ID = */ret);
      //fingerprint.delFingerprint(DELALL);  //Delete all fingerprints
      Serial.print("delete succeeds,ID=");
      Serial.println(ret);
      /*Set fingerprint LED ring to always ON in green */
```

```
      fingerprint.ctrlLED(/*LEDMode = */fingerprint.eKeepsOn, /*LEDColor = */fingerprint.eLEDGreen, /*blinkCount = */0);
    }else{
      Serial.println("Fingerprint is unregistered");
      /*Set fingerprint LED ring to always ON in red*/
      fingerprint.ctrlLED(/*LEDMode = */fingerprint.eKeepsOn, /*LEDColor = */fingerprint.eLEDRed, /*blinkCount = */0);

    }
  }else{
    Serial.println("Capturing fails");
    /*Get error code information*/
    //desc = fingerprint.getErrorDescription();
    //Serial.println(desc);
    /*Set fingerprint LED ring to always ON in red*/
    fingerprint.ctrlLED(/*LEDMode = */fingerprint.eKeepsOn, /*LEDColor = */fingerprint.eLEDRed, /*blinkCount = */0);
  }
  Serial.println("Please release your finger");
  /*Wait for finger to release
    Return 1 when finger is detected, otherwise return 0
   */
  while(fingerprint.detectFinger());
    /*Check whether the fingerprint ID has been registered
    Return 1 if registered, otherwise return 0
   */
//  if(fingerprint.getStatusID(/*Fingerprint ID = */ret)){
//    Serial.println("ID has been registered");
//  }else{
//    Serial.println("ID is unregistered");
//  }
  Serial.println("----------------------------");
  delay(1000);
}
```

- Result



Sample Code 5 - Comprehensive Examples

This example is a comprehensive application of fingerprint module, in which the module will enter into different modes according to the length of finger pressing time. When a finger pressed down, the blue light flashes 3 times to enter the fingerprint comparison mode; the yellow light flashes 3 times to enter the fingerprint registration mode; the red light flashes 3 times to enter the fingerprint deletion mode to delete the fingerprint. This experiment needs to connect IRQ pin to D6.

```
/*!
 * @file comprehensiveExample.ino
 * @brief Comprehensive Example
 * @n This module can be controlled by hardware serial or software serial. Pin IRQ should be connected to D6 in this experiment.
 * @n Experiment Phenomenon: when finger press down, the blue LED blinks quickly 3 times, which means it enters fingerprint
 * @n comparison mode.
 * @n                        The yellow LED blinks quickly 3 times for entering fingerprint registration mode
 * @n                        The red LED blinks quickly 3 times for entering fingerprint deletion mode, and delete this fringerprint
 * @n                        Enter sleep mode when idle for 5 seconds
 * @copyright  Copyright (c) 2010 DFRobot Co.Ltd (http://www.dfrobot.com)
 * @licence     The MIT License (MIT)
 * @author [Eddard](Eddard.liu@dfrobot.com)
 * @version  V1.0
 * @date  2020-03-19
 * @get from https://www.dfrobot.com
 * @url https://github.com/cdjq/DFRobot_ID809
*/

#include <DFRobot_ID809.h>

#define COLLECT_NUMBER 3  //Fingerprint sampling times, can be set to 1-3
#define IRQ         6  //IRQ pin

DFRobot_ID809_IIC fingerprint;
//DFRobot_ID809_UART fingerprint(115200);
//String desc;

void setup(){
  /*Init print serial port*/
  Serial.begin(9600);
  /*Take FPSerial as communication serial port of the fingerprint module*/
  fingerprint.begin();
```

```
  fingerprint.begin();
  /*Wait for Serial to open*/
  while(!Serial);
  /*Test whether the device can properly communicate with mainboard
    Return true or false

    */
  while(fingerprint.isConnected() == false){
    Serial.println("Communication with device failed, please check connection");
    /*Get error code information*/
    //desc = fingerprint.getErrorDescription();
    //Serial.println(desc);
    delay(1000);
  }
}

//Blue LED Comparison mode  Yellow LED Registration mode  Red Deletion mode
void loop(){
  if(digitalRead(IRQ)){
    uint16_t i = 0;
    /*Capture fingerprint image, 5s idle timeout
      Return 0 if succeed, otherwise return ERR_ID809
     */
    if((fingerprint.collectionFingerprint(/*timeout=*/5)) != ERR_ID809){
      /*Get the time finger pressed down*/
      /*Set fingerprint LED ring mode, color, and number of blinks
        Can be set as follows:
        Parameter 1:<LEDMode>
        eBreathing   eFastBlink   eKeepsOn    eNormalClose
        eFadeIn      eFadeOut     eSlowBlink
        Paramerer 2:<LEDColor>
        eLEDGreen  eLEDRed      eLEDYellow   eLEDBlue
        eLEDCyan   eLEDMagenta  eLEDWhite
        Parameter 3:<number of blinks> 0 represents blinking all the time
        This parameter will only be valid in mode eBreathing, eFastBlink, eSlowBlink
       */
      fingerprint.ctrlLED(/*LEDMode = */fingerprint.eFastBlink, /*LEDColor = */fingerprint.eLEDBlue, /*blinkCount = */3);  //blue LED
      /*Wait for finger to relase */
```

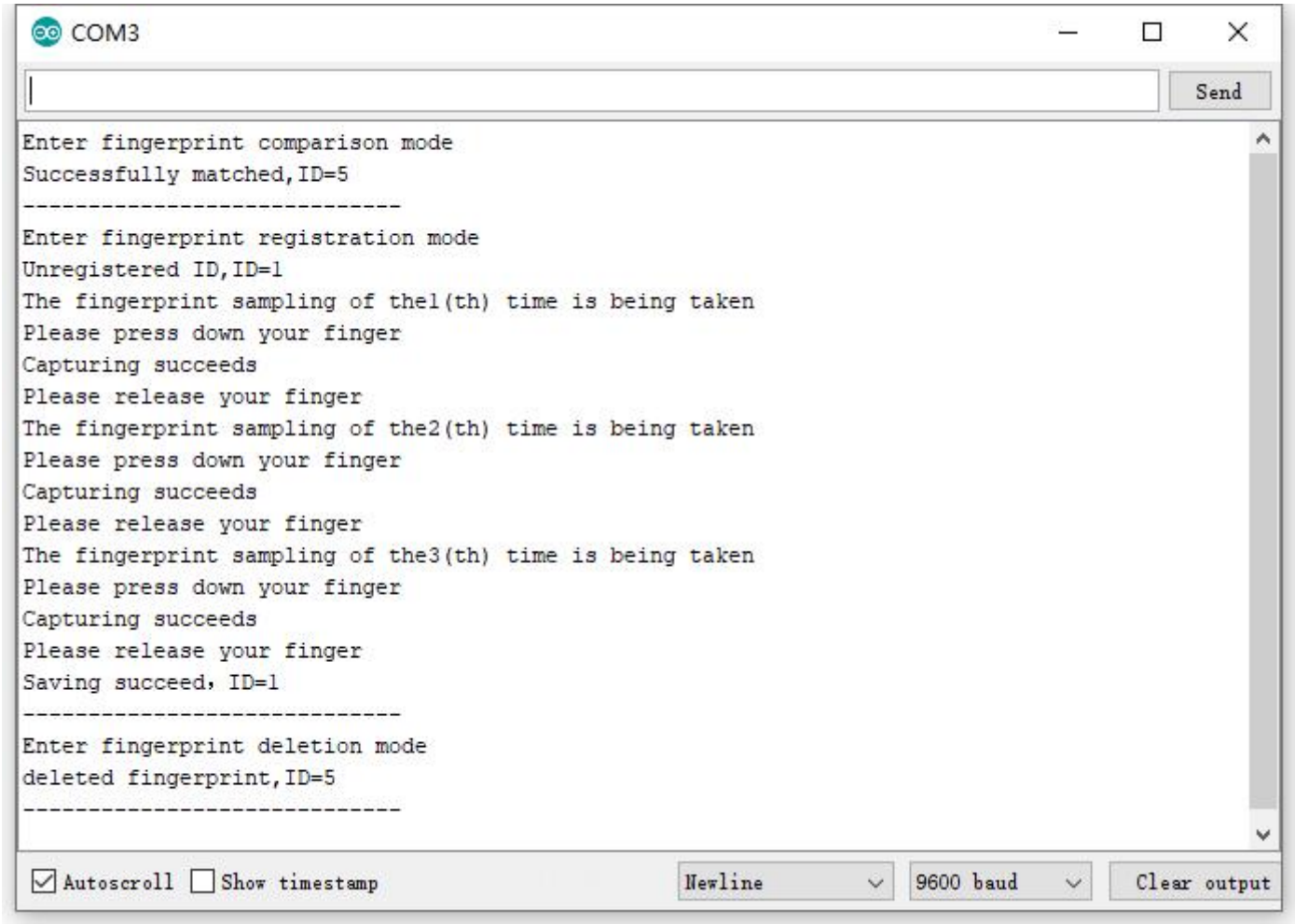
```
      while(fingerprint.detectFinger()){
        delay(50);
        i++;
        if(i == 15){             //Yellow LED blinks quickly 3 times, means it's in fingerprint regisrtation mode now
          /*Set fingerprint LED ring to always ON in yellow*/

          fingerprint.ctrlLED(/*LEDMode = */fingerprint.eFastBlink, /*LEDColor = */fingerprint.eLEDYellow, /*blinkCount = */3);
        }else if(i == 30){      //Red LED blinks quickly 3 times, means it's in fingerprint deletion mode now
          /*Set fingerprint LED ring to always ON in red*/
          fingerprint.ctrlLED(/*LEDMode = */fingerprint.eFastBlink, /*LEDColor = */fingerprint.eLEDRed, /*blinkCount = */3);
        }
      }
    }
    if(i == 0){
      /*Fingerprint capturing failed*/
    }else if(i > 0 && i < 15){
      Serial.println("Enter fingerprint comparison mode");
      /*Compare fingerprints*/
      fingerprintMatching();
    }else if(i >= 15 && i < 30){
      Serial.println("Enter fingerprint registration mode");
      /*Registrate fingerprint*/
      fingerprintRegistration();
    }else{
      Serial.println("Enter fingerprint deletion mode");
      /*Delete this fingerprint*/
      fingerprintDeletion();
    }
  }
}

//Compare fingerprints
void fingerprintMatching(){
  /*Compare the captured fingerprint with all fingerprints in the fingerprint library
    Return fingerprint ID number(1-80) if succeed, return 0 when failed
   */
  uint8_t ret = fingerprint.search();
  if(ret != 0){
```

```
    /*Set fingerprint LED ring to always ON in green*/
    fingerprint.ctrlLED(/*LEDMode = */fingerprint.eKeepsOn, /*LEDColor = */fingerprint.eLEDGreen, /*blinkCount = */0);
    Serial.print("Successfully matched,ID=");
    Serial.println(ret);
  }else{
    /*Set fingerprint LED Ring to always ON in red*/
    fingerprint.ctrlLED(/*LEDMode = */fingerprint.eKeepsOn, /*LEDColor = */fingerprint.eLEDRed, /*blinkCount = */0);
    Serial.println("Matching failed");
  }
  delay(1000);
  /*Turn off fingerprint LED Ring*/
  fingerprint.ctrlLED(/*LEDMode = */fingerprint.eNormalClose, /*LEDColor = */fingerprint.eLEDBlue, /*blinkCount = */0);
  Serial.println("-----------------------------");
}

//Fingerprint Registration
void fingerprintRegistration(){
  uint8_t ID,i;
  /*Compare the captured fingerprint with all fingerprints in the fingerprint library
    Return fingerprint ID number(1-80) if succeed, return 0 when failed
    Function: clear the last captured fingerprint image
   */
  fingerprint.search();       //Can add "if else" statement to judge whether the fingerprint has been registered.
  /*Get a unregistered ID for saving fingerprint
    Return ID number when succeed
    Return ERR_ID809 if failed
   */
  if((ID = fingerprint.getEmptyID()) == ERR_ID809){
    while(1){
      /*Get error code imformation*/
      //desc = fingerprint.getErrorDescription();
      //Serial.println(desc);
      delay(1000);
    }
  }
  Serial.print("Unregistered ID,ID=");
  Serial.println(ID);
```

```
i = 0;   //Clear sampling times
/*Fingerprint Sampling 3 times */
while(i < COLLECT_NUMBER){
  /*Set fingerprint LED ring to breathing lighting in blue*/
  fingerprint.ctrlLED(/*LEDMode = */fingerprint.eBreathing, /*LEDColor = */fingerprint.eLEDBlue, /*blinkCount = */0);

  Serial.print("The fingerprint sampling of the");
  Serial.print(i+1);
  Serial.println("(th) time is being taken");
  Serial.println("Please press down your finger");
  /*Capture fingerprint image, 10s idle timeout
    If succeed return 0, otherwise return ERR_ID809
   */
  if((fingerprint.collectionFingerprint(/*timeout = */10)) != ERR_ID809){
    /*Set fingerprint LED ring to quick blink in yellow 3 times*/
    fingerprint.ctrlLED(/*LEDMode = */fingerprint.eFastBlink, /*LEDColor = */fingerprint.eLEDYellow, /*blinkCount = */3);
    Serial.println("Capturing succeeds");
    i++;   //Sampling times +1
  }else{
    Serial.println("Capturing fails");
    /*Get error code information*/
    //desc = fingerprint.getErrorDescription();
    //Serial.println(desc);
  }
  Serial.println("Please release your finger");
  /*Wait for finger to release
    Return 1 when finger is detected, otherwise return 0
   */
  while(fingerprint.detectFinger());
}

/*Save fingerprint information into an unregistered ID*/
if(fingerprint.storeFingerprint(/*Empty ID = */ID) != ERR_ID809){
  Serial.print("Saving succeed,ID=");
  Serial.println(ID);
  /*Set fingerprint LED ring to always ON in green*/
  fingerprint.ctrlLED(/*LEDMode = */fingerprint.eKeepsOn, /*LEDColor = */fingerprint.eLEDGreen, /*blinkCount = */0);
  delay(1000);
```

```
    /*Turn off fingerprint LED ring */
    fingerprint.ctrlLED(/*LEDMode = */fingerprint.eNormalClose, /*LEDColor = */fingerprint.eLEDBlue, /*blinkCount = */0);
  }else{
    Serial.println("Saving failed");
    /*Get error code information*/

    //desc = fingerprint.getErrorDescription();
    //Serial.println(desc);
  }
  Serial.println("-----------------------------");
}


//Fingerprint deletion
void fingerprintDeletion(){
  uint8_t ret;
  /*Compare the captured fingerprint with all the fingerprints in the fingerprint library
    Return fingerprint ID(1-80) if succeed, return 0 when failed
   */
  ret = fingerprint.search();
  if(ret){
    /*Set fingerprint LED ring to always ON in green*/
    fingerprint.ctrlLED(/*LEDMode = */fingerprint.eKeepsOn, /*LEDColor = */fingerprint.eLEDGreen, /*blinkCount = */0);
    fingerprint.delFingerprint(ret);
    Serial.print("deleted fingerprint,ID=");
    Serial.println(ret);
  }else{
    /*Set fingerprint LED ring to always ON in red*/
    fingerprint.ctrlLED(/*LEDMode = */fingerprint.eKeepsOn, /*LEDColor = */fingerprint.eLEDRed, /*blinkCount = */0);
    Serial.println("Matching failed or the fingerprint hasn't been registered");
  }
  delay(1000);
  /*Turn off fingerprint LED ring*/
  fingerprint.ctrlLED(/*LEDMode = */fingerprint.eNormalClose, /*LEDColor = */fingerprint.eLEDBlue, /*blinkCount = */0);
  Serial.println("-----------------------------");
}
```

- Result



FAQ

For any questions, advice or cool ideas to share, please visit the **DFRobot Forum** (https://www.dfrobot.com/forum/).

More Documents

- Schematics Diagram (https://dfimg.dfrobot.com/nobody/wiki/dba17c874730936d437b8f4c31250934.pdf)
- Dimension Diagram (https://dfimg.dfrobot.com/nobody/wiki/9da85c7b0f09627eec7017b2e5be6c22.pdf)

 Get **Gravity: Capacitive Fingerprint Scanner/Sensor** (https://www.dfrobot.com/product-2165.html) from DFRobot Store or **DFRobot Distributor**. (https://www.dfrobot.com/index.php?route=information/distributorslogo)

Turn to the Top