Capacitive Fingerprint Recognition Modu

BM92S2231-1
Arduino Library V1.0.1 Description

Revision: V1.00 Date: May 28, 2024

www.bestmodulescorp.com

Contents

Introduction

The BM92S2231-1 is a capacitive fingerprint recognition module from Best Modules, which uses the UART communication method. This document provides the description of the BM92S2231-1 Arduino Lib functions and how to install the Arduino Lib. The example uses the BMA92K223 module to demonstrate the functions of fingerprint enrollment and fingerprint recognition.

Applicable types:

Part No.	Description
BM92S2231-1	Capacitive Fingerprint Recognition Module
BMA92K223	Including adapter board, connection line and the BM92S2231-1

Arduino Library Functions

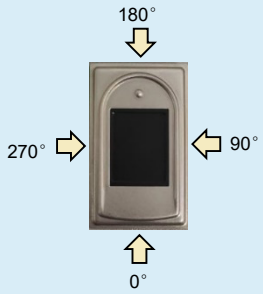
Arduino Lib Name: BM92S2231-1		Lib Version: V1.0.1
Constructors & Initialisation		
1	BM92S2231_1(uint8_t keyoutPin=2,HardwareSerial *theSerial = &Serial)	
	Description	Constructor, select the hardware UART communication
	Parameter	keyoutPin: Touch pin (default D2), connects to the Keyout pin of the BM92S2231-1 or the I/O (Keyout) pin of the BMA92K223 *theSerial: Select the Hardware UART interface (default Serial1 interface)
	Return Value	—
	Note	—
2	BM92S2231_1(uint8_t keyoutPin,uint8_t rxPin, uint8_t txPin)	
	Description	Constructor, select the software UART communication
	Parameter	keyout: Touch pin, connects to the Keyout pin of the BM92S2231-1 or the I/O (Keyout) pin of the BMA92K223 rxPin: RX pin, connects to the TX pin of the BM92S2231-1 or the BMA92K223 txPin: TX pin, connects to the RX pin of the BM92S2231-1 or the BMA92K223
	Return Value	—
	Note	—
3	void begin(unsigned long baud = 115200)	
	Description	Module initialisation
	Parameter	baud: Communication baud rate selection (default 115200 bps)
	Return Value	void
	Note	—
Performance Functions – Enrollment		
4	int8_t InputEnrollID(int16_t ID)	
	Description	Enter the fingerprint ID to be enrolled
	Parameter	ID: Fingerprint ID, range: 1~1000
	Return Value	Execution result: 0: ID is within range and not enrolled 4: ID has been enrolled 8: ID is out of range
	Note	—

5	uint8_t enroll()	
	Description	Enroll fingerprint
	Parameter	—
	Return Value	Execution result: 0x01: Press finger again 0x02: Enrollment succeeded 0x03: Enrollment failed 0x05: Try again with other areas of your finger 0x06: Please move your finger
	Note	Require to be used after “if(InputEnrollID ()==0)”
6	bool stopEnroll()	
	Description	Stop enrolling
	Parameter	—
	Return Value	Execution result: true: Succeeded false: Failed
	Note	—
Performance Functions – Fingerprint Recognition		
7	uint8_t getKeyout()	
	Description	Obtain the I/O (Keyout) pin current status
	Parameter	—
	Return Value	I/O (Keyout) pin level 0: Low, not pressed 1: High, pressed
	Note	—
8	int16_t identify()	
	Description	Recognise the enrolled fingerprint ID of the pressed finger
	Parameter	—
	Return Value	Execution result or ID: 0: No fingerprint pressing information 0xffff: Fingerprint detection failed Others: Fingerprint detection succeeded, matching the fingerprint ID of the pressed finger
	Note	—
9	bool stopIdentify()	
	Description	Stop detection
	Parameter	—
	Return Value	Execution result: true: Succeeded false: Failed
	Note	—
Performance Functions – Fingerprint Deletion		
10	uint8_t deleteID(uint16_t id)	
	Description	Delete the fingerprint with specfied ID
	Parameter	id: Fingerprint ID, range:1~500
	Return Value	Execution result: 0x00: Data reception error 0x01: Deletion succeeded 0x02: Deletion failed 0x03: ID to be deleted does not exist
	Note	—

11	bool deleteAll()	
	Description	Delete the fingerprints for all IDs
	Parameter	—
	Return Value	Execution result: true: Succeeded false: Failed
	Note	—
Performance Functions – General		
12	bool StandbyMode()	
	Description	Enter the standby mode
	Parameter	—
	Return Value	Enter standby mode status: true: Succeeded false: Failed
	Note	It can be woken up by touching the fingerprint detection area with finger
13	bool changeBaud(unsigned long baud)	
	Description	Change the module baud rate
	Parameter	baud: Communication baud rate selection 9600:9600bps 19200:19200bps 38400:38400bps 57600:57600bps 115200:115200bps (default) 128000:128000bps 230400:230400bps 256000:256000bps 460800:460800bps
	Return Value	Execution result: true: Succeeded false: Failed
	Note	When the module is powered on again or enters the standby mode and woken up again, the baud rate will be changed to the default baud rate
14	uint8_t getDeviceInformation()	
	Description	Return the obtained module information
	Parameter	—
	Return Value	Execution result: 0: Obtain failed 1: BM92S2131_1 2: BM92S2231_1 3: BM92S2331_1
	Note	
15	bool getModuleSettingsInformation(uint8_t inforArray[])	
	Description	Obtain the fractional threshold, detection angle and template number currently configured for the module
	Parameter	inforArray[]: Obtained parameter information inforArray[0]: Fractional threshold low bits inforArray[1]: Fractional threshold high bits inforArray[2]: Detection angle inforArray[3]: Template number
	Return Value	Execution result: true: Succeeded false: Failed
	Note	—

16	bool getImageSettingInformation(uint8_t inforArray[])	
	Description	Obtain the image threshold and image percentage currently configured for the module
	Parameter	inforArray[]: Obtained parameter information inforArray[0]: Image threshold low bits inforArray[1]: Image threshold high bits inforArray[2]: Image percentage low bits inforArray[3]: Image percentage high bits
	Return Value	Execution result: true: Succeeded false: Failed
	Note	—
17	bool userSet(uint16_t score,uint8_t checkAngle,uint8_t numberTemplates)	
	Description	User setting
	Parameter	score: Fractional threshold, If the fingerprint detection fraction is greater than the fractional threshold, the detection is successful ranges from 0~100, default 90 checkAngle: Detection angle (Note) 0x00: Full angle detection method 1, the finger pressing detection direction can be 0°,90°,180° and 270° 0x01: Single angle detection, the finger pressing detection direction must be consistent with the enrolled direction (0°) 0x03: Three angle detection, the finger pressing detection direction can be 0°,90° and 270° 0x04(default): Full angle detection method 2, the finger pressing detection direction can be 0°,90°,180° and 270° numberTemplates: Template number Ranges from 1~3, default 3
	Return Value	Execution result: true: Succeeded false: Failed
	Note	① The FAR (false acceptance rate) and FRR(false rejection rate) are determined by the template number together with the Fractional threshold. Template number 1: Fractional threshold 79 (FAR≤0.001%,FRR≤9%) Fractional threshold 73 (FAR≤0.01%,FRR≤7%) Template number 2: Fractional threshold 89 (FAR≤0.001%,FRR≤5%) Fractional threshold 78 (FAR≤0.01%, FRR≤3%) Template number 3: Fractional threshold 87 (FAR≤0.001%,FRR≤3%) Fractional threshold 80 (FAR≤0.01%,FRR≤2%) ② The difference between the full angle detection method 1 and 2: method 1 takes less time than method 2, but the recognition effect is slightly worse than method 2.
18	bool imageSet(uint16_t imageThreshold,uint16_t imagePercentage)	
	Description	Image setting
	Parameter	imageThreshold: Image threshold, ranges from 50~1000, default 128 imagePercentage: Image percentage, ranges from 0~100, default 60
	Return Value	Execution result: true: Succeeded false: Failed
	Note	—

Note: Sensor direction

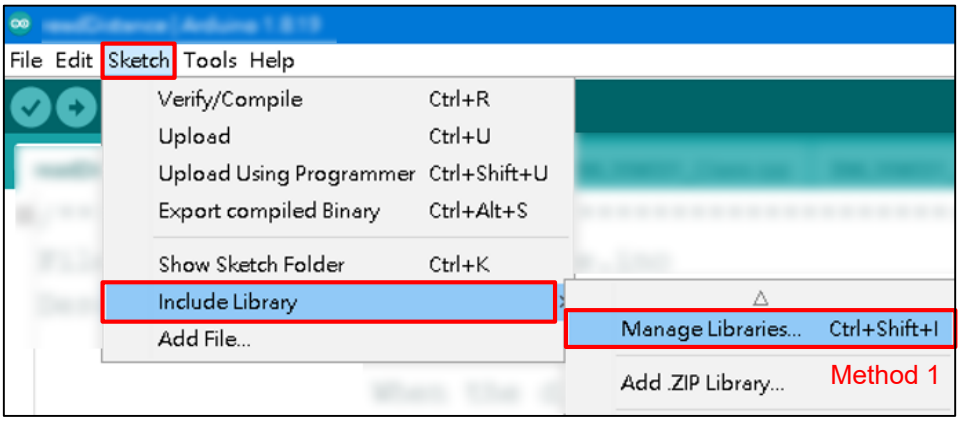


Arduino Lib Download and Installation

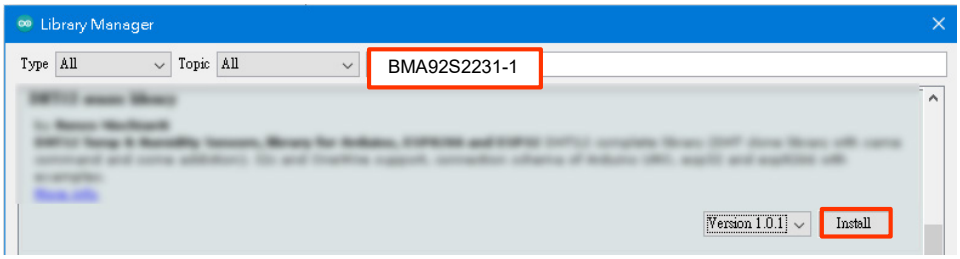
BM92S2231-1 A Library: Refer to the following two methods to install the BM92S2231-1 Arduino Library

Method 1: Search for installation

Arduino IDE → Sketch → Include Library → Manage Libraries... → Search BM92S2231-1 → Install



Search for Installation Step 1

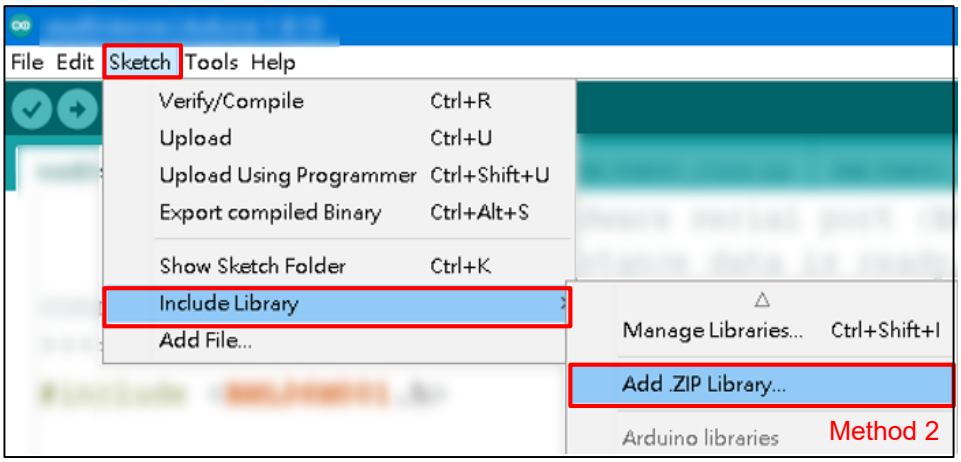


Search for Installation Step 2

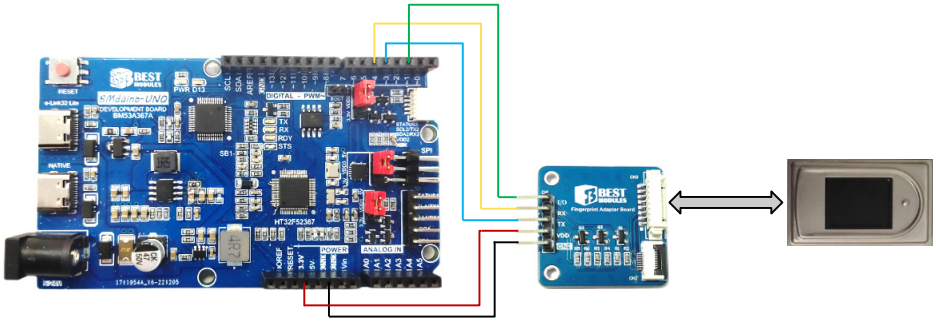
Method 2: Download before adding a ZIP library

Download method: Open the Best Modules official website (https://www.bestmodulescorp.com/bm92s2231-1.html) and download the BM92S2231-1 Library from "Arduino example program" under the "DOCUMENTS" menu.

Add .ZIP library: Arduino IDE → Sketch → Include Library → Add .ZIP Library...



Arduino Examples



Physical Connection Diagram

Note: The touch sensitivity of the module touch is high. When not using fingerprint detection, do not touch the module with your finger.

Example 1: enroll

Example 1 function: Enroll the fingerprint with one finger and display it on the serial monitor.

1. Open the example: Arduino IDE → File → Examples → Select Lib (BM92S2231-1) → Select example (enroll)

2. Example description:

a. Create object & Module initialisation and configuration

```
#include "BM92S2231-1.h"
BM92S2231_1 fps(2,4,5); // Uses the software UART communication.
                        // Keyout pin is connected to the D2, TX pin
                        // is connected to the D4, RX pin is connected
                        // to the D5
uint16_t ID=6; // Create variable, for storing enrolled ID
uint8_t idCheck = 1,enrollEnable=0; // Create variable, idCheck
                                    // for recording whether ID
                                    // detection is enabled (enable
                                    // detection: idCheck=1; disable
                                    // detection: idCheck=0),
                                    // enrollEnable for recording
                                    // whether the enrollment is
                                    // enabled (enable enrollment
                                    // enrollEnable=1; disable
                                    // enrollment: enrollEnable=0)
uint16_t enrollState; // Create variable, for recording the execution
                      // result of the enrollment
uint8_t idState=0xff;  // Create variable, for recording the usage of
                       // the IDs
uint16_t timeCnt=0; // Create variable, for recording the time that
                    // finger was not detected
void setup() {
  Serial.begin(115200); // Serial monitor initialisation
  delay(100);  // Wait for the module to stabilise
  fps.begin(); // Module initialisation
}
```

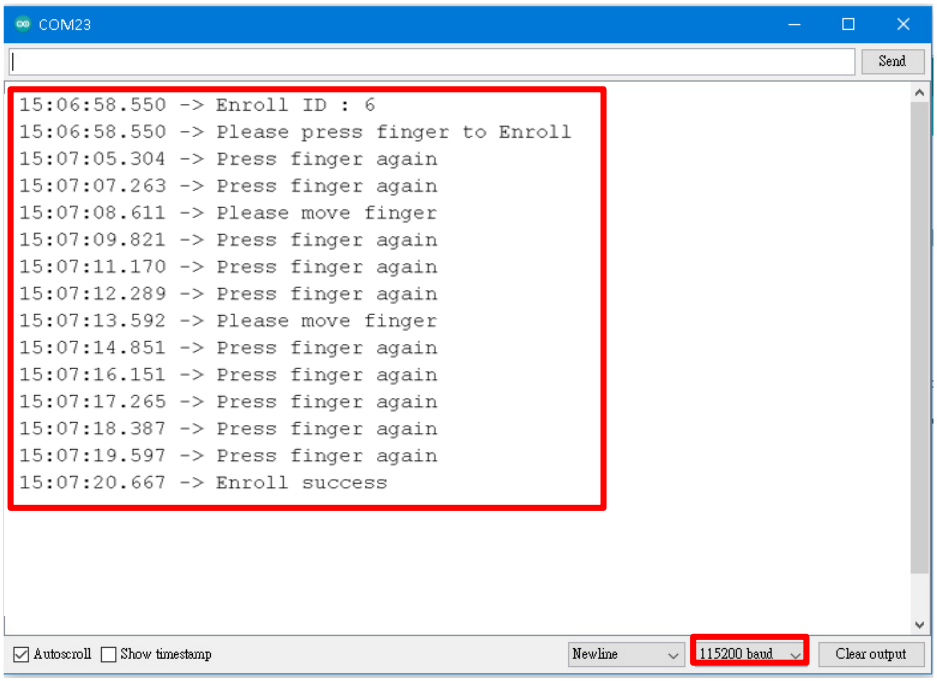
b. Enroll the fingerprint and display it on the serial monitor

```
void loop() {

     if(idCheck==1)
    {
      idState=fps.InputEnrollID(ID); // Detect the input ID and record
                                     // the current usage of the ID
      if(idState==0) // ID is within the range and not used
      {
        idCheck=0; // Exit the ID detection
        enrollEnable=1; // Enable the enrollment
        timeCnt=0; // Clear the fingerprint detection time to 0
        Serial.print(F("Enroll ID : ")); // Print the fingerprint ID to
                                         // be enrolled
        Serial.println(ID);
        Serial.println(F("Please press finger to Enroll"));
        // Print a prompt to enable the enrollment
      }
      else if(idState==4) // If ID has been used, increment by 1 and
                          // perform ID detection
      {
       ID ++;
       //Serial.println(F("ID already used"));
      }
```

```
      else if(idState==8) // ID out of range
      {
        idCheck=0;
       Serial.println(F("ID out of range"));
      }
      else // Communication error
      {
        idCheck=0;
       Serial.println(F("Communication error"));
      }
    }
    if(enrollEnable==1) // Enable the enrollment
    {

      enrollState=fps.enroll(); // Enable the fingerprint enrollment
      if(enrollState!=0) // enrollState is not 0, clear time count
                         // to 0
        timeCnt=0;

      if(enrollState==1) // enrollState=1, prompt to press finger
                         // again
       Serial.println(F("Press finger again"));
      else if(enrollState==2) // enrollState=2, enrollment succeeded
      {
       enrollEnable=0; // Disable the enrollment
       Serial.println(F("Enroll success")); // Prompt for enrollment
                                            // succeeded

       if(fps.stopEnroll()==false) // Stop enrolling
       Serial.println(F("Communication error"));
       else
       idCheck=1; // Enable the ID detection
       Serial.println(F(""));

      }

      else if(enrollState==3) // enrollState=3, enrollment failed
      {
        enrollEnable=0; // Disable the enrollment
        Serial.println(F("Enroll fail")); // Prompt for enrollment
                                          // failed

        if(fps.stopEnroll()==false) // Stop enrolling
        Serial.println(F("Communication error"));
        else
        idCheck=1; // Enable the ID detection

        Serial.println(F(""));
      }
```

```
        else if(enrollState==5) // enrollState=5, prompt to try again
                                // with other areas on the surface of
                                // your fingers
        Serial.println(F("Try again with other areas on the surface of
your fingers"));

        else if(enrollState==6) // enrollState=6, prompt to move the
                                // finger
         Serial.println(F("Please move finger"));

        else if(enrollState==0) // enrollState=0, no finger pressed
        {
          timeCnt++; // Time count is incremented by 1
          if(timeCnt>60) // Time count exceeds 60
          {
            Serial.println(F("No fingerprint detected"));
            // Prompt for no finger pressed
            enrollEnable=0; // Disable the enrollment

            if(fps.stopEnroll()==false) // Stop enrolling
            Serial.println(F("Communication error"));
          }
       }

    }

}
         else if(enrollState==5) // Require more fingerprint information
        Serial.println(F("Try again with other areas on the surface of
your fingers"));

        else if(enrollState==6||enrollState==7) // Require to move
                                               // finger
         Serial.println(F("Please move finger"));

        else if(enrollState==0)
        {
          timeCnt++;
          if(timeCnt>60) // Timeout detection, if there is no
                         // operation for 30s
          {
            Serial.println(F("No fingerprint detected"));
            enrollEnable=0;

            if(fps.stopEnroll()==false)
            Serial.println(F("Communication error"));
          }
       }

    }

}
```

3. Open the serial monitor and select the baud rate to be 115200. The serial monitor will display as follows:



Example 2: identify

Example 2 function: Recognise whether the fingerprint is a enrolled fingerprint, and display the enrolled fingerprint ID on the serial monitor.

1. Open the example: Arduino IDE → File → Examples → Select Lib (BM92S2231-1) → Select example (identify)
2. Example description:

a. Create object & Module initialisation and configuration

```
#include "BM92S2231-1.h" // Include the header file

BM92S2231_1 fps(2,4,5); // Uses the software UART communication.
                        // Keyout pin is connected to the D2, TX pin
                        // is connected to the D4, RX pin is connected
                        // to the D5
uint16_t result;
void setup() {
  Serial.begin(115200); // Serial initialisation
  delay(100);  // Power on and wait for initialisation to complete
  fps.begin(); // Initialise the module
  Serial.println(F("Please press your finger to start detection"));
}
```

b. Recognise whether the fingerprint is a enrolled fingerprint and display it on the serial monitor

```
void loop() {
  if(fps.getKeyout()!=BM92S2231_1_NO_KEY) // If a touch pressing is
                                          // detected
  {
    delay(100); // Wake-up time from Deep Sleep
    result=fps.identify(); // Recognise fingerprint
    if(result>0) // Recognition completed
    {
      if(result==0xffff) // Recognition failed
      {
      Serial.println(F("Detection failed"));
      }
      else // Recognition succeeded
      {
      Serial.print(F("Verified ID: "));
      Serial.println(result);
      }
    }
  }
```

3. Open the serial monitor and select the baud rate to be 115200. The serial monitor will display as follows: