電容式指紋辨識模組

BM92S2231-1
Arduino Library V1.0.1 說明

版本：V1.00　日期：2024-05-28

www.bestmodulescorp.com

目錄

簡介 3
Arduino Library 函式 3
Arduino Lib 下載及安裝 8
Arduino 範例 9
範例 1：enroll 9
範例 2：identify 12

簡介

BM92S2231-1 是倍創推出的電容式指紋辨識模組，使用 UART 通訊方式。本文檔對 BM92S2231-1 的 Arduino Lib 函式、Arduino Lib 安裝方式進行說明；範例使用 BMA92K223 模組，演示了指紋註冊、指紋辨識功能。

適用型號：

型號	說明
BM92S2231-1	電容式指紋辨識模組
BMA92K223	包括轉接板、連接線、BM92S2231-1

Arduino Library 函式

Arduino Lib 名稱：BM92S2231-1		Lib 版本：V1.0.1
構造函式 & 初始化		
1	BM92S2231_1(uint8_t keyoutPin=2,HardwareSerial *theSerial = &Serial)	
	描述	構造函式，選擇硬體 UART 通訊
	參數	keyoutPin：觸控腳位 (預設為 D2)，連接 BM92S2231-1 的 Keyout 腳位或 BMA92K223 的 I/O (Keyout) 腳位 *theSerial：選擇硬體 UART 介面 (預設為 Serial1 介面)
	回傳值	—
	備註	—
2	BM92S2231_1(uint8_t keyoutPin,uint8_t rxPin, uint8_t txPin)	
	描述	構造函式，選擇軟體 UART 通訊
	參數	keyout：觸控腳位，連接 BM92S2231-1 的 Keyout 腳位或 BMA92K223 的 I/O (Keyout) 腳位 rxPin：RX 腳位，連接 BM92S2231-1 或 BMA92K223 的 TX 腳位 txPin：TX 腳位，連接 BM92S2231-1 或 BMA92K223 的 RX 腳位
	回傳值	—
	備註	—
3	void begin(unsigned long baud = 115200)	
	描述	模組初始化
	參數	baud：通訊鮑率選擇 (預設 115200 bps)
	回傳值	void
	備註	—
功能函式 – 註冊		
4	int8_t InputEnrollID(int16_t ID)	
	描述	輸入註冊的指紋 ID
	參數	ID：指紋 ID，範圍：1~1000
	回傳值	執行情況 0：ID 在範圍內且未被使用 4：ID 已使用 8：ID 超出範圍
	備註	—

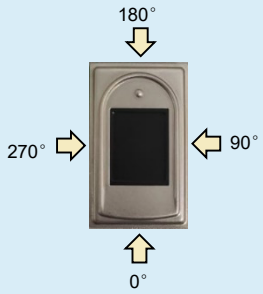
5	uint8_t enroll()	
	描述	註冊指紋
	參數	—
	回傳值	執行情況： 0x01：再次按壓手指 0x02：註冊成功 0x03：註冊失敗 0x05：用手指其他區域重試 0x06：請移動手指
	備註	需在 “if(InputEnrollID ()==0)” 後使用
6	bool stopEnroll()	
	描述	停止註冊
	參數	—
	回傳值	執行情況 true：成功 false：失敗
	備註	—
功能函式 – 辨識指紋		
7	uint8_t getKeyout()	
	描述	獲取 I/O (Keyout) 腳位當前狀態
	參數	—
	回傳值	I/O (Keyout) 腳位準位 0：Low，無觸摸按下 1：High，有觸摸按下
	備註	—
8	int16_t identify()	
	描述	根據放置的手指指紋 (已註冊)，辨識出其 ID
	參數	—
	回傳值	執行情況或 ID： 0：無指紋按壓信息 0xffff：檢測指紋失敗 其他：檢測成功，對應放置的手指指紋的 ID
	備註	—
9	bool stopIdentify()	
	描述	停止檢測
	參數	—
	回傳值	執行情況： true：成功 false：失敗
	備註	—

功能函式 – 刪除指紋		
10	uint8_t deleteID(uint16_t id)	
	描述	刪除指定 ID 的指紋
	參數	id：指紋 ID，範圍：1~1000
	回傳值	執行情況： 0x00：資料接收有誤 0x01：刪除成功 0x02：刪除失敗 0x03：刪除的 ID 不存在
	備註	—
11	bool deleteAll()	
	描述	刪除全部 ID 的指紋
	參數	—
	回傳值	執行情況 true：成功 false：失敗
	備註	—
功能函式 – 通用		
12	bool StandbyMode()	
	描述	進入待機模式
	參數	—
	回傳值	進入待機模式情況： true：成功 false：失敗
	備註	可以通過手指觸摸指紋檢測區域來喚醒
13	bool changeBaud(unsigned long baud)	
	描述	修改模組的鮑率
	參數	baud：通訊鮑率選擇 9600：9600bps 19200：19200bps 38400：38400bps 57600：57600bps 115200：115200bps (預設) 128000：128000bps 230400：230400bps 256000：256000bps 460800：460800bps
	回傳值	執行情況： true：成功 false：失敗
	備註	模組斷電後重新上電或進入待機模式再次喚醒時鮑率將變為預設鮑率

14	uint8_t getDeviceInformation()	
	描述	回傳獲取的模組設備資訊
	參數	—
	回傳值	執行情況： 0：獲取失敗 1：BM92S2131_1 2：BM92S2231_1 3：BM92S2331_1
	備註	
15	bool getModuleSettingsInformation(uint8_t inforArray[])	
	描述	獲取模組目前配置的分數閾值、檢測角度和模板數量。
	參數	inforArray[]：獲取到的參數資訊 inforArray[0]：分數閾值低位 inforArray[1]：分數閾值高位 inforArray[2]：檢測角度 inforArray[3]：模板數量
	回傳值	執行情況： true：成功 false：失敗
	備註	—
16	bool getImageSettingInformation(uint8_t inforArray[])	
	描述	獲取模組目前配置的圖像閾值和圖像百分比
	參數	inforArray[]：獲取到的參數資訊 inforArray[0]：圖像閾值低位 inforArray[1]：圖像閾值高位 inforArray[2]：圖像百分比低位 inforArray[3]：圖像百分比高位
	回傳值	執行情況： true：成功 false：失敗
	備註	—

17	bool userSet(uint16_t score,uint8_t checkAngle,uint8_t numberTemplates)	
	描述	用戶設置
	參數	score：分數閾值，指紋檢測分數大於分數閾值，則檢測成功 範圍是 0~100，預設 90 checkAngle：檢測角度(註) 0x00：全角度檢測方式 1，手指檢測的按壓方向可以是 0°、90°、180°、270° 0x01：單角度檢測，手指檢測的按壓方向需和註冊的方向一致 (0°) 0x03：3 個角度檢測，手指檢測的按壓方向可以是 0°、90°、270° 0x04(default)：全角度檢測方式 2，手指檢測的按壓方向可以是 0°、90°、180°、270° numberTemplates：模板數量 範圍是 1~3，預設 3
	回傳值	執行情況： true：成功 false：失敗
	備註	① 模板數量和分數閾值的設置共同決定 FAR (誤識率)，FRR (拒識率) 模板數量 1：分數閾值 79 (FAR≤0.001%，FRR≤9%) 分數閾值 73 (FAR≤0.01%，FRR≤7%) 模板數量 2：分數閾值 89 (FAR≤0.001%，FRR≤5%) 分數閾值 78 (FAR≤0.01%，FRR≤3%) 模板數量 3：分數閾值 87 (FAR≤0.001%，FRR≤3%) 分數閾值 80 (FAR≤0.01%，FRR≤2%) ② 全角度檢測方式 1、2 的區別：1 的耗時比 2 少，但是辨識效果比 2 略差。
18	bool imageSet(uint16_t imageThreshold,uint16_t imagePercentage)	
	描述	圖像設置
	參數	imageThreshold：圖像閾值，範圍是 50~1000，預設 128 imagePercentage：圖像百分比，範圍是 0~100，預設 60
	回傳值	執行情況： true：成功 false：失敗
	備註	—

註：感測器的方向

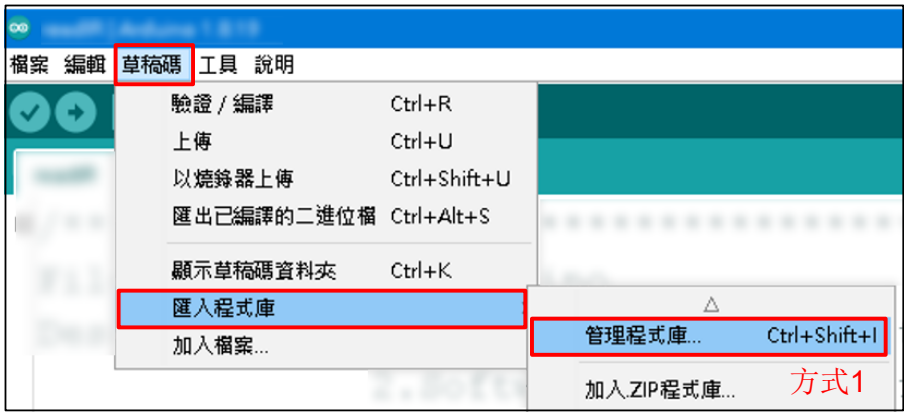


Arduino Lib 下載及安裝

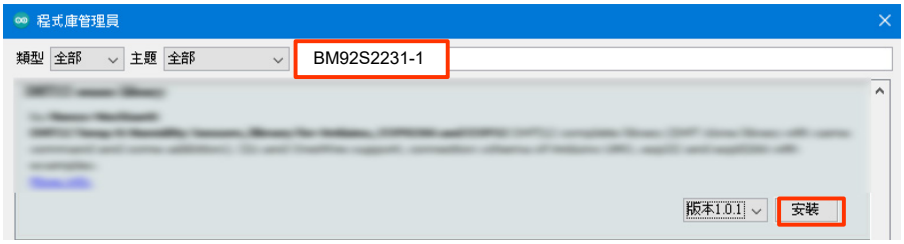
BM92S2231-1 A Library：可參考下面兩種方法安裝 BM92S2231-1 的 Arduino Library

方式 1：搜索安裝

搜索安裝：Arduino IDE → 草稿碼 → 匯入程式庫 → 管理程式庫 → 搜索 BM92S2231-1 → 安裝



搜索安裝流程 1

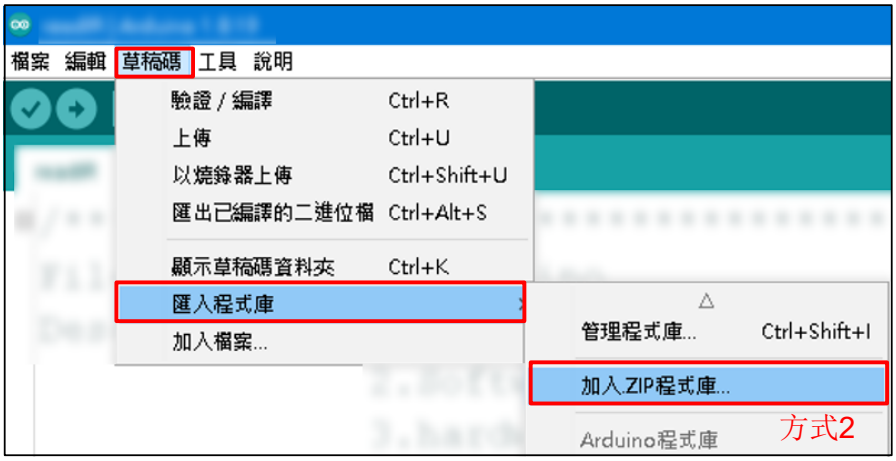


搜索安裝流程 2

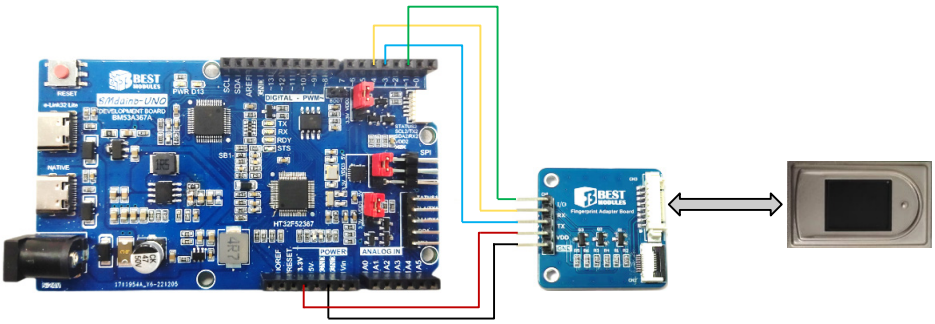
方式 2：加入 .ZIP 程式庫，需提前下載 .ZIP 程式庫

下載方法：打開倍創官方網站 (https://www.bestmodulescorp.com/bm92s2231-1.html) 文件目錄下的 Arduino 範例程式 (BM92S2231-1.zip)。

加入 .ZIP 程式庫：Arduino IDE → 草稿碼 → 匯入程式庫 → 加入 .ZIP 程式庫 ...



Arduino 範例



實物連接示意圖

注意：模組 touch 靈敏度高，不檢測指紋時，手指請勿觸碰到模組

範例 1：enroll

範例 1 實現功能：通過一個手指進行註冊指紋，並在序列埠監視器上顯示。

1. 範例打開方式：Arduino IDE → 檔案 → 範例 → Lib 選擇 (BM92S2231-1) → 選擇範例 (enroll)
2. 範例說明：
   a. 構建對象 & 模組初始化及配置

```
#include "BM92S2231-1.h"
BM92S2231_1 fps(2,4,5); // 使用軟體 UART 通訊。Keyout 腳位連接 D2，TX 腳位連
                        // 接 D4，RX 腳位連接 D5
uint16_t ID=6; // 創建變數，用於存儲註冊 ID
uint8_t idCheck = 1,enrollEnable=0; // 創建變數，idCheck 用於記錄是否開啟
                                    // ID 檢測（開啟檢測 idCheck=1；關閉
                                    // 檢測：idCheck=0 ），enrollEnable
                                    // 用於記錄是否開啟註冊（開啟註冊
                                    // enrollEnable=1；關閉註冊：
                                    // enrollEnable=0 ）
uint16_t enrollState; // 創建變數，用於記錄註冊的執行情況
uint8_t idState=0xff;  // 創建變數，用於記錄 ID 使用情況
uint16_t timeCnt=0;    // 創建變數，用於記錄未檢測到手指的時間
void setup() {
  Serial.begin(115200); // 序列埠監視器初始化
  delay(100);  // 等待模組穩定
  fps.begin(); // 模組初始化
}
```

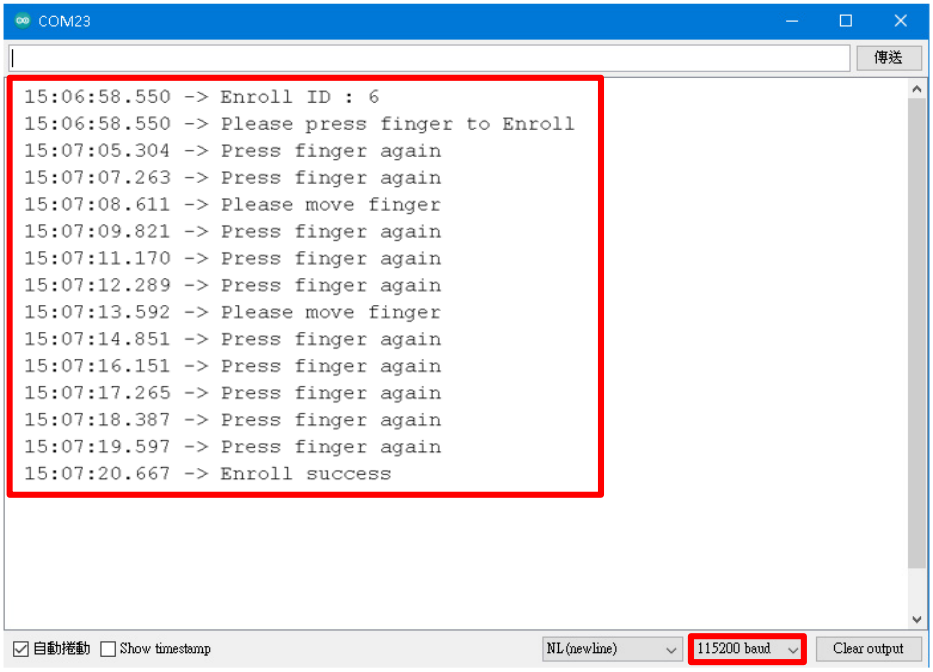
b. 注册指纹并显示在序列埠監視器上

```
void loop() {

     if(idCheck==1)
    {
      idState=fps.InputEnrollID(ID); // 檢測輸入的 ID，記錄當前 ID 使用情況
      if(idState==0) // ID 在範圍內且未被使用
      {
        idCheck=0; // 退出 ID 檢測
        enrollEnable=1; // 開啟註冊
        timeCnt=0; // 手指檢測時間清 0
        Serial.print(F("Enroll ID : ")); // 列印要錄入指紋的 ID
        Serial.println(ID);
        Serial.println(F("Please press finger to Enroll"));
        // 列印開始註冊的提示
      }
      else if(idState==4) // ID 已被使用，加 1 再進行 ID 檢測
      {
       ID ++;
       //Serial.println(F("ID already used"));
      }
      else if(idState==8) // ID 超出範圍
      {
        idCheck=0;
       Serial.println(F("ID out of range"));
      }
      else // 通訊有誤
      {
        idCheck=0;
       Serial.println(F("Communication error"));
      }
    }
    if(enrollEnable==1) // 開始註冊
    {

        enrollState=fps.enroll(); // 開始指紋註冊
        if(enrollState!=0) // enrollState 不為 0，時間計數清 0
          timeCnt=0;

        if(enrollState==1) // enrollState=1，提示繼續按壓手指
         Serial.println(F("Press finger again"));

        else if(enrollState==2) // enrollState=2，註冊成功
        {
         enrollEnable=0; // 關閉註冊
         Serial.println(F("Enroll success")); // 提示註冊成功

         if(fps.stopEnroll()==false) // 停止註冊
         Serial.println(F("Communication error"));
         else
         idCheck=1; // 開啟 ID 檢測
         Serial.println(F(""));
        }
```

```
        else if(enrollState==3) // enrollState=3，註冊失敗
        {
          enrollEnable=0; // 關閉註冊
          Serial.println(F("Enroll fail")); // 提示註冊失敗

          if(fps.stopEnroll()==false) // 停止註冊
          Serial.println(F("Communication error"));
          else
          idCheck=1; // 開啟 ID 檢測

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

        else if(enrollState==5) // enrollState=5，提示用手指表面其他區域
                                // 重試
        Serial.println(F("Try again with other areas on the surface
of your fingers"));

        else if(enrollState==6) // enrollState=6，提示移動手指
         Serial.println(F("Please move finger"));

        else if(enrollState==0) // enrollState=0，手指沒有按壓
        {
          timeCnt++; // 時間計數加 1
          if(timeCnt>60) // 時間計數超過 60
          {
            Serial.println(F("No fingerprint detected")); // 提示沒有
                                                          // 手指按下
            enrollEnable=0; // 關閉註冊

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

    }

}         else if(enrollState==5) // 需要更多的指紋資訊
        Serial.println(F("Try again with other areas on the surface
of your fingers"));

        else if(enrollState==6||enrollState==7) // 需要移動手指
         Serial.println(F("Please move finger"));

        else if(enrollState==0)
        {
          timeCnt++;
          if(timeCnt>60) // 超時檢測，30 秒無操作
          {
            Serial.println(F("No fingerprint detected"));
            enrollEnable=0;
```

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

    }

}
```

3. 打開序列埠監視器，鮑率選擇 115200。序列埠監視器顯示如下：



範例 2：identify

範例 2 實現功能：辨識指紋是否為註冊過的指紋，並在序列埠監視器上顯示註冊的指紋 ID。

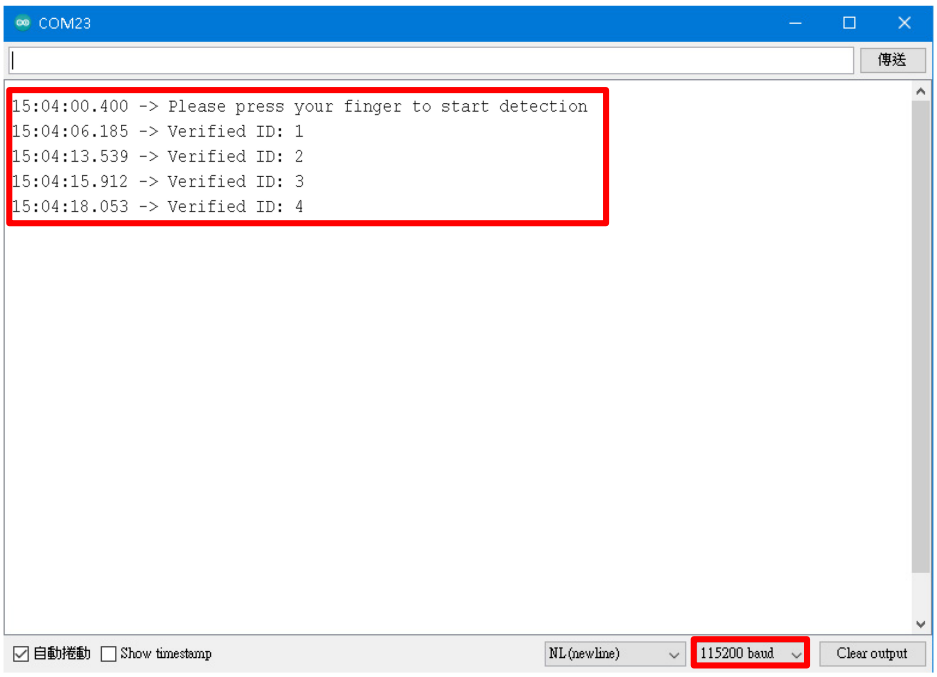
1. 範例打開方式：Arduino IDE → 檔案 → 範例 → Lib 選擇 (BM92S2231-1) → 選擇範例 (identify)
2. 示例說明：
   a. 構建對象 & 模組初始化及配置

```
#include "BM92S2231-1.h" // 包含頭文件

BM92S2231_1 fps(2,4,5);  // 使用軟體 UART 通訊。Keyout 腳位連接 D2，TX 腳位
                         // 連接 D4，RX 腳位連接 D5
uint16_t result;
void setup() {
  Serial.begin(115200); // 初始化序列埠
  delay(100);  // 上電等待初始化完成
  fps.begin(); // 初始化模組
  Serial.println(F("Please press your finger to start detection"));
}
```

b. 辨識指紋是否為註冊過的指紋並顯示在序列埠監視器上

```
void loop() {
  if(fps.getKeyout()!=BM92S2231_1_NO_KEY) // 如果檢測到觸控按下
  {
    delay(100); // 深度休眠喚醒時間
    result=fps.identify(); // 辨識指紋
    if(result>0) // 辨識完成
    {
      if(result==0xffff) // 辨識失敗
      {
      Serial.println(F("Detection failed"));
      }
      else // 辨識成功
      {
      Serial.print(F("Verified ID: "));
      Serial.println(result);
      }
    }
  }
```

3. 打開序列埠監視器，鮑率選擇 115200。序列埠監視器顯示如下：



Copyright© 2024 by BEST MODULES CORP. All Rights Reserved.

本文件出版時倍創已針對所載資訊為合理注意，但不保證資訊準確無誤。文中提到的資訊僅是提供作為參考，且可能被更新取代。倍創不擔保任何明示、默示或法定的，包括但不限於適合商品化、令人滿意的品質、規格、特性、功能與特定用途、不侵害第三人權利等保證責任。倍創就文中提到的資訊及該資訊之應用，不承擔任何法律責任。此外，倍創並不推薦將倍創的產品使用在會因故障或其他原因而可能會對人身安全造成危害的地方。倍創特此聲明，不授權將產品使用於救生、維生或安全關鍵零組件。在救生 / 維生或安全應用中使用倍創產品的風險完全由買方承擔，如因該等使用導致倍創遭受損害、索賠、訴訟或產生費用，買方同意出面進行辯護、賠償並使倍創免受損害。倍創 (及其授權方，如適用) 擁有本文件所提供資訊 (包括但不限於內容、資料、示例、材料、圖形、商標) 的智慧財產權，且該資訊受著作權法和其他智慧財產權法的保護。倍創在此並未明示或暗示授予任何智慧財產權。倍創擁有不事先通知而修改本文件所載資訊的權利。如欲取得最新的資訊，請與我們聯繫。