



---

**BMduino-Shield**  
四通道馬達驅動擴充板

**BMP73T104**  
**Arduino Library V1.0.2 說明**

版本：V1.03 日期：2023-10-19

[www.bestmodulescorp.com](http://www.bestmodulescorp.com)

## 目錄

簡介 .....	3
Arduino Lib 函式 .....	3
Arduino Lib 下載及安裝 .....	9
Arduino 範例 .....	10
範例 1：dcMotorRun .....	10
範例 2：stepperRun .....	11
範例 3：twoSteppersRun .....	12

## 簡介

BMP73T104 是倍創推出的四通道馬達驅動擴充板，使用 I<sup>2</sup>C 通訊方式。本文檔對 BMP73T104 Arduino Lib 函式、Arduino Lib 安裝方式進行說明；範例演示了驅動直流馬達和步進馬達功能。

## Arduino Lib 函式

Arduino Lib 名稱：BMP73T104		Lib 版本：V1.0.2
<b>構造成函式 &amp; 初始化</b>		
1	<b>BMP73T104(uint8_t intPin, uint8_t object, TwoWire *theWire=&amp;Wire)</b>	
	描述	構造成函式，設定馬達類型、wire 參數
	參數	intPin：INT 腳位 object：馬達類型 0x01 (StepMotor)：步進馬達 0x00 (DCMotor)：直流馬達 *theWire：wire 參數
	返回值	—
	備註	—
2	<b>uint8_t begin(uint8_t motor, uint8_t interface=0, uint8_t i2c_addr=Slave_Addr)</b>	
	描述	對象初始化，選擇輸出通道，設定步進類型、擴充板位址
	參數	motor：馬達號 0x01 (M1)：直流馬達 1 0x02 (M2)：直流馬達 2 0x04 (M3)：直流馬達 3 0x08 (M4)：直流馬達 4 0x03 (STEP1)：步進馬達 1 0x0c (STEP2)：步進馬達 2 interface：步進馬達驅動方式，步進馬達時才設定 ininterface，預設整步驅動 0 (FULL_STEP)：整步驅動 1 (HALF_STEP)：半步驅動 2 (MICRO_STEP)：微步驅動 i2c_addr：I <sup>2</sup> C 通訊位址，預設 0x19
	返回值	執行情況： 0：成功 1：失敗
	備註	使用馬達時必須先初始化對應馬達介面。

直流馬達功能函式		
3	uint8_t setOCP(uint8_t ocp, uint8_t debounce=1, uint8_t status=ENABLE)	
	描述	設定直流馬達過流保護
	參數	ocp：過流電流參數 ocp = ( 過流電流 × 0.05 ÷ 開發板的 IOREF 電壓 ) × 4095 ( 取整數 ) debounce：去抖時間參數 去抖時間 = debounce×5ms status：開關 0x01 (ENABLE)：開啟 0x00 (DISABLE)：關閉
	返回值	執行情況： 0：成功 1：失敗
	備註	—
4	uint8_t clearOverCurrent()	
	描述	清除當前過流狀態
	參數	—
	返回值	執行情況： 0：成功 1：失敗
	備註	—
5	uint8_t dcMotorRun(int8_t rank)	
	描述	直流馬達轉動
	參數	rank：速度檔位，範圍 [-100, 100]，正為順時針，負為逆時針，絕對值表示速度，數值越大馬達速度越快
	返回值	執行情況： 0：成功 1：失敗
	備註	—
6	uint8_t dcMotorStop()	
	描述	直流馬達停止
	參數	—
	返回值	執行情況： 0：成功 1：失敗
	備註	—
7	uint8_t dcMotorBrake()	
	描述	直流馬達制動
	參數	—
	返回值	執行情況： 0：成功 1：失敗
	備註	—

直流馬達參數獲取函式		
8	uint8_t getINT()	
	描述	獲取 INT 腳位準位
	參數	—
	返回值	INT 腳位準位狀態： 0：低準位 1：高準位
備註	過流保護未開啟：恒為高準位 過流保護開啟：有直流馬達過流為低準位；否則為高準位	
9	int8_t getDcMotorRank()	
	描述	獲取直流馬達速度檔位
	參數	—
	返回值	直流馬達速度檔位·範圍 [-100, 100]
備註	正負代表方向：正為順時針，負為逆時針	
步進馬達功能函式		
10	uint8_t stepperMoveTo(int32_t absolute, uint16_t speed)	
	描述	步進馬達勻速轉動到目標位置
	參數	absolute：目標位置。 speed：轉動速度(步/s)
	返回值	執行情況： 0：成功 1：失敗
備註	轉動速度建議值： 全步進下建議 <sup>(1)</sup> 不超過 800 步/s 半步進下建議 <sup>(1)</sup> 不超過 1600 步/s 微步進下建議 <sup>(1)</sup> 不超過 3200 步/s	
11	uint8_t stepperMoveTo(int32_t absolute)	
	描述	步進馬達具備加減速轉動到目標位置
	參數	absolute：目標位置
	返回值	執行情況： 0：成功 1：失敗
備註	—	
12	uint8_t stepperMove(int32_t relative, uint16_t speed)	
	描述	步進馬達勻速轉動目標步數
	參數	relative：目標步數·正為順時針·負為逆時針·絕對值表示步數。 speed：轉動速度(步/s)
	返回值	執行情況： 0：成功 1：失敗
備註	轉動速度建議值： 全步進下建議 <sup>(1)</sup> 不超過 800 步/s 半步進下建議 <sup>(1)</sup> 不超過 1600 步/s 微步進下建議 <sup>(1)</sup> 不超過 3200 步/s	

13	uint8_t stepperMove(int32_t relative)	
	描述	步進馬達具備加減速轉動目標步數
	參數	relative：目標步數，正為順時針，負為逆時針，絕對值表示步數
	返回值	執行情況： 0：成功 1：失敗
	備註	—
14	uint8_t stepperRun(uint8_t dir, uint16_t speed)	
	描述	步進馬達勻速持續轉動
	參數	dir：步進馬達轉動方向 0x00 (CW)：順時針 0x01 (CCW)：逆時針 speed：轉動速度 (步 / s)
	返回值	執行情況： 0：成功 1：失敗
	備註	轉動速度建議值： 全步進下建議 <sup>(1)</sup> 不超過 800 步 / s 半步進下建議 <sup>(1)</sup> 不超過 1600 步 / s 微步進下建議 <sup>(1)</sup> 不超過 3200 步 / s
15	uint8_t stepperStop()	
	描述	步進馬達停止轉動
	參數	—
	返回值	執行情況： 0：成功 1：失敗
	備註	—
16	uint8_t stepperRelease()	
	描述	釋放保持轉矩功能，如果馬達在轉動，則馬達停下
	參數	—
	返回值	執行情況： 0：成功 1：失敗
	備註	—
17	uint8_t isRunning()	
	描述	步進馬達是否正在向目標位置轉動
	參數	—
	返回值	轉動情況： 0：未在向目標轉動 1：正在向目標轉動
	備註	—

步進馬達參數獲取 & 設定函式		
18	uint16_t getStepperMaxSpeed()	
	描述	獲取步進馬達最大速度
	參數	—
	返回值	步進馬達最大速度 · 單位步 / s
	備註	—
19	uint16_t getStepperSpeed()	
	描述	獲取步進馬達當前速度
	參數	—
	返回值	步進馬達當前速度 · 單位步 / s
	備註	—
20	uint16_t getStepperAcceleration()	
	描述	獲取步進馬達加速度
	參數	—
	返回值	步進馬達加速度 · 單位步 / s <sup>2</sup>
	備註	—
21	int32_t getStepperTargetPosition()	
	描述	獲取步進馬達目標位置
	參數	—
	返回值	步進馬達目標位置 (步)
	備註	—
22	int32_t getStepperDistanceToGo()	
	描述	獲取步進馬達距離目標位置
	參數	—
	返回值	步進馬達距離目標位置 (步)
	備註	—
23	int32_t getStepperPosition()	
	描述	獲取步進馬達當前位置
	參數	—
	返回值	步進馬達當前位置 (步)
	備註	—
24	uint8_t setStepperMaxSpeed(uint16_t maxSpeed)	
	描述	設定步進馬達最大速度
	參數	maxSpeed : 步進馬達最大速度 (步 / s)
	返回值	執行情況： 0 : 成功 1 : 失敗
	備註	步進馬達最大速度建議值： 全步進下建議不超過 800 步 / s 半步進下建議 <sup>(1)</sup> 不超過 1600 步 / s 微步進下建議 <sup>(1)</sup> 不超過 3200 步 / s

25	uint8_t setStepperAcceleration(uint16_t acceleration)	
	描述	設定步進馬達加速度
	參數	acceleration：步進馬達加速度 (步 / s <sup>2</sup> )
	返回值	執行情況： 0：成功 1：失敗
	備註	—
26	uint8_t setStepperCurrentPosition(int32_t position)	
	描述	設定步進馬達當前位置
	參數	position：位置
	返回值	執行情況： 0：成功 1：失敗
	備註	—
27	uint8_t setStepperHoldTorque(uint8_t status=DISABLE)	
	描述	設定步進馬達保持轉矩
	參數	status：保持轉矩開關 0 (ENABLE)：打開 1 (DISABLE)：關閉，預設關閉
	返回值	執行情況： 0：成功 1：失敗
	備註	—
<b>其他功能函式</b>		
28	uint8_t setIICSlaveAddr(uint8_t i2c_addr)	
	描述	設定擴充板 I <sup>2</sup> C 位址
	參數	i2c_addr：I <sup>2</sup> C 位址，0~127
	返回值	執行情況： 0：成功 1：失敗
	備註	—
29	void reset()	
	描述	重置擴充板
	參數	—
	返回值	void
	備註	任意對象調用
30	uint16_t getFWVer()	
	描述	獲取當前版本號
	參數	—
	返回值	版本號
	備註	任意對象調用

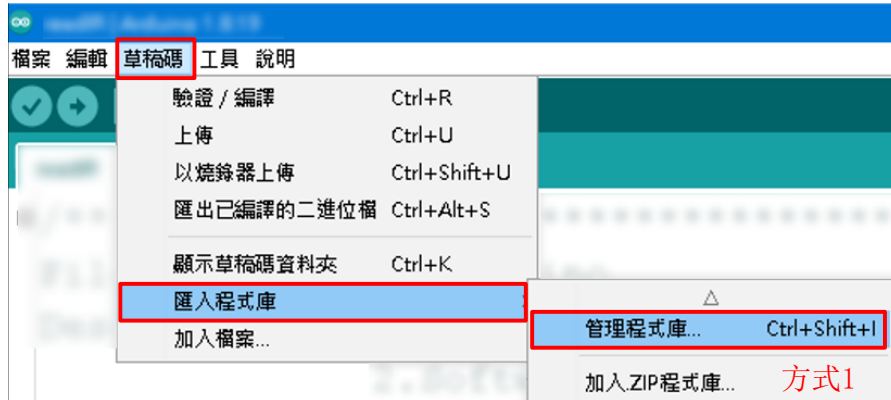
註 1：不同型號的步進馬達在不同電壓下，能到達的最大速度有所不同，因此需要根據自己使用的電源對馬達速度進行調整。此處建議值測試條件馬達電源 12V，測試馬達為 1.8° 步距角的步進馬達。

## Arduino Lib 下載及安裝

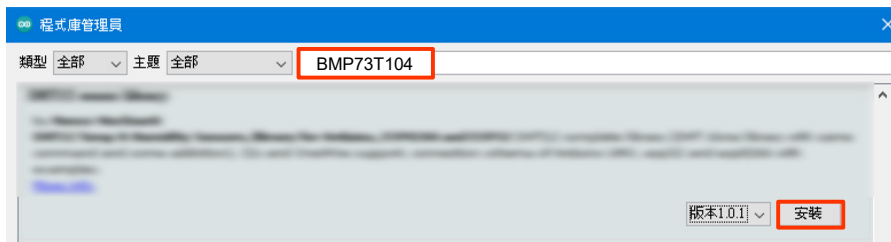
BMP73T104 Library：可參考下面方法安裝 BMP73T104 的 Arduino Library

### 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 BMP73T104 → 安裝



搜索安裝流程 1

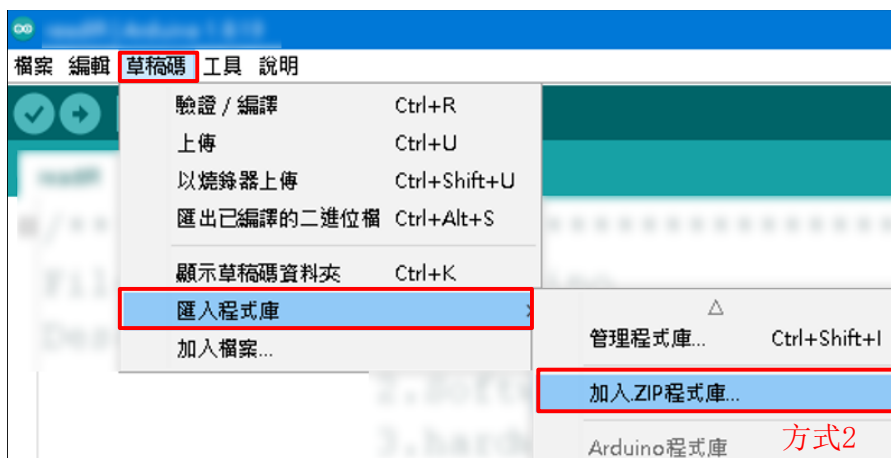


搜索安裝流程 2

### 方式 2：加入 .ZIP 程式庫，需提前下載 .ZIP 程式庫

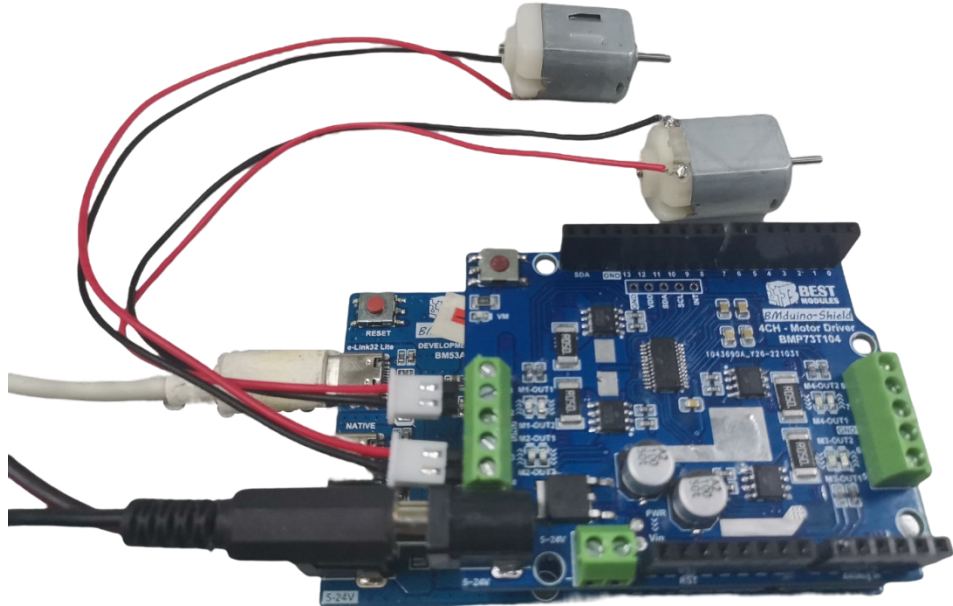
下載方法：打開倍創官方網站 (<https://www.bestmodulescorp.com/bmp73t104.html>) 文件目錄下的 Arduino 範例程式 (BMP73T104 Library)。

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



## Arduino 範例

### 範例 1 : dcMotorRun



實物連接示意圖

範例實現功能：演示驅動直流馬達功能

1. 馬達 1 以 80 檔速度順時針轉動，馬達 2 以 60 檔速度順時針轉動，持續 5 秒。
  2. 馬達 1 以 60 檔速度逆時針轉動，馬達 2 以 80 檔速度逆時針轉動，持續 5 秒。
- 馬達在 1、2 動作循環。

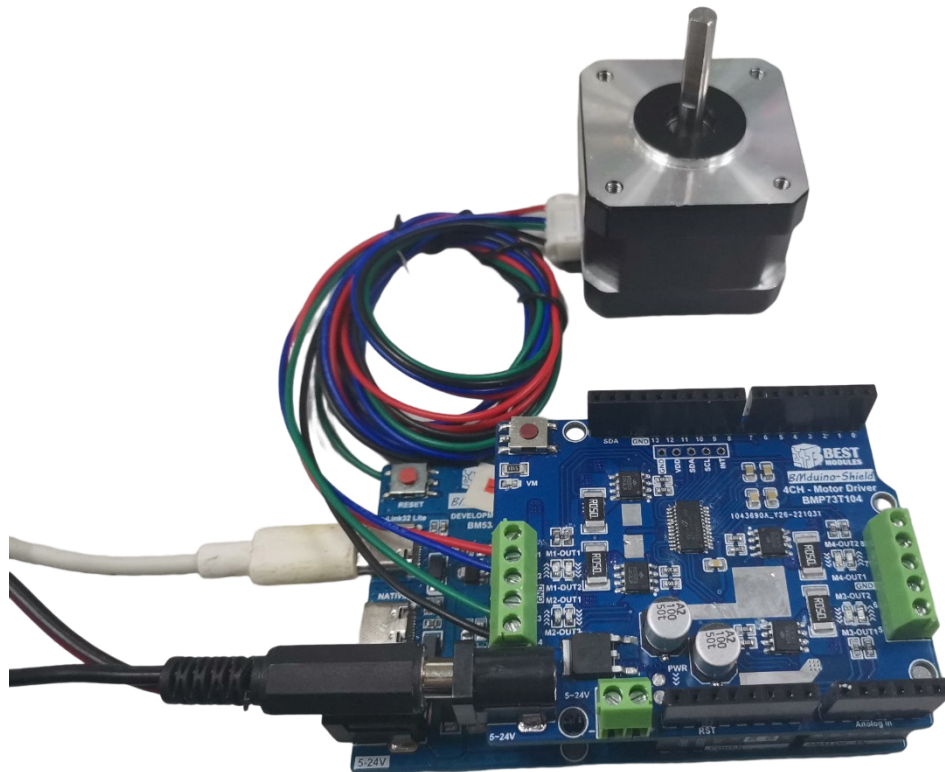
1. 範例打開：檔案 → 範例 → Lib 選擇 (BMP73T104) → 選擇範例 (dcMotorRun)
2. 範例說明：
  - a. 構建 & 初始化對象

```
#include <BMP73T104.h>
BMP73T104 myM1(2,DCMotor); // 建立直流馬達對象
BMP73T104 myM2(2,DCMotor); // 建立直流馬達對象
void setup()
{
  myM1.begin(M1);           // 直流馬達 1 初始化
  // M1 初始化為直流馬達 1
  myM2.begin(M2);           // 直流馬達 2 初始化
  // M2 初始化為直流馬達 2
}
```

b. 直流馬達轉動

```
void loop()
{
  myM1.dcMotorRun(80); // 直流馬達 1 以 80 檔速度正轉
  myM2.dcMotorRun(60); // 直流馬達 2 以 60 檔速度正轉
  delay(5000);          // 轉動 5s
  myM1.dcMotorRun(-60); // 直流馬達 1 以 60 檔速度反轉
  myM2.dcMotorRun(-80); // 直流馬達 2 以 80 檔速度反轉
  delay(5000);          // 轉動 5s
}
```

## 範例 2 : stepperRun



實物連接示意圖

範例實現功能：演示驅動步進馬達功能

1. 設定步進馬達介面 1 加速度和最大速度。
  2. 步進馬達 1 以 1000 步 / s 的速度順時針轉動 1000 步。
  3. 步進馬達加減速轉動到位置 0。
- 步進馬達在 2、3 動作循環。

1. 範例打開：檔案 → 範例 → Lib 選擇 (BMP73T104) → 選擇範例 (stepperRun)

## 2. 範例說明：

### a. 構建 & 初始化對象

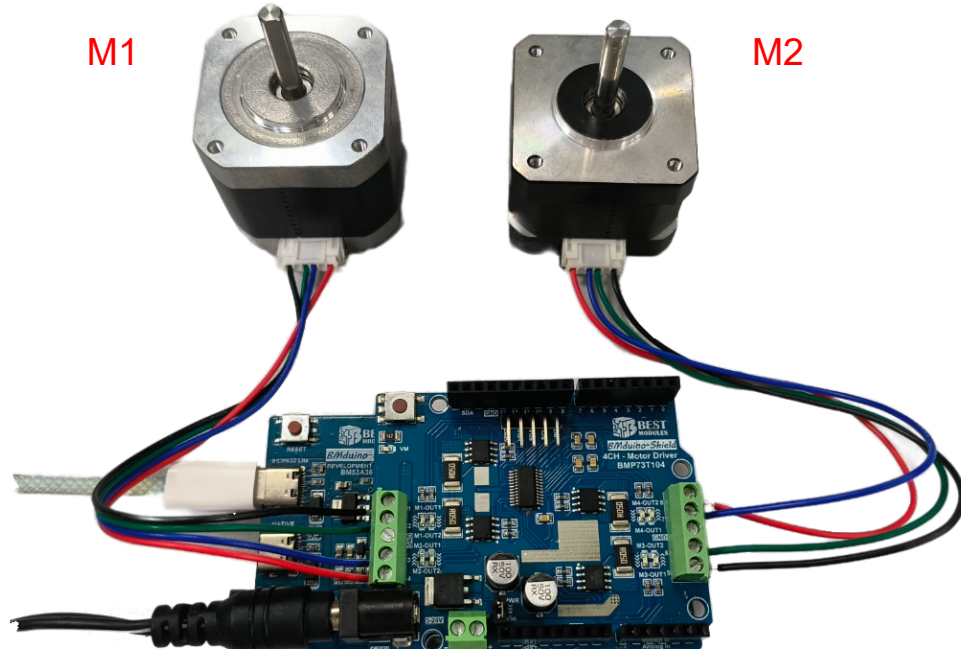
```
#include <BMP73T104.h>
BMP73T104 myStepper1(2, StepMotor); // 建立對象
void setup()
{
  myStepper1.begin(STEP1, HALF_STEP); // 初始化
  myStepper1.setStepperMaxSpeed(1200); // 步進馬達 1 設定步進馬達最大速度
  myStepper1.setStepperAcceleration(800); // 步進馬達 1 設定步進馬達加速度
}
```

### b. 步進馬達轉動

```
void loop() {
  myStepper1.stepperMove(2000, 1000); // 步進馬達 1 以 1000 步 / s 的速度轉
  // 動到 2000 的位置

  delay(5000); // 等待 5s
  myStepper1.stepperMoveTo(0); // 步進馬達 1 以有加速度的轉動到 0 的位置
  delay(5000); // 等待 5s
}
```

## 範例 3：twoSteppersRun



實物連接示意圖

範例實現功能：演示驅動兩步進馬達功能

1. 步進馬達 1、步進馬達 2 介面選擇，並進行初始化，然後設置步進馬達 1、步進馬達 2 的加速度和最大速度。
2. 步進馬達 1 以 200 步/s 的速度順時針轉動 3 秒；同時步進馬達 2 以 800 步/s 的速度逆時針轉動 3 秒。
3. 步進馬達 1、步進馬達 2 停止轉動 1 秒。

4. 步進馬達 1 以 200 步 /s 的速度逆時針轉動 3 秒；同時步進馬達 2 以 800 步 /s 的速度順時針轉動 3 秒。

5. 步進馬達 1、步進馬達 2 停止轉動 1 秒。

循環 2、3、4、5 動作。

1. 範例打開：檔案 → 範例 → Lib 選擇 (BMP73T104) → 選擇範例 (twoSteppersRun)

2. 範例說明：

a. 構建 & 初始化對象

```
#include <BMP73T104.h>
BMP73T104 myStepper1(2,StepMotor); // 建立對象 1
BMP73T104 myStepper2(2,StepMotor); // 建立對象 2
void setup() {
  myStepper1.begin(STEP1,HALF_STEP); // 步進馬達 1 初始化
  myStepper1.setStepperMaxSpeed(1200); // 步進馬達 1 設定步進馬達最大速度
  myStepper1.setStepperAcceleration(800); // 步進馬達 1 設定步進馬達加速度

  myStepper2.begin(STEP2,HALF_STEP); // 步進馬達 2 初始化
  myStepper2.setStepperMaxSpeed(1200); // 步進馬達 2 設定步進馬達最大速度
  myStepper2.setStepperAcceleration(800); // 步進馬達 2 設定步進馬達加速度
}
```

b. 步進馬達轉動

```
void loop()
{
  myStepper1.stepperRun(CW,200); // 步進馬達 1 轉動：順時針·速度 200 步 /s
  myStepper2.stepperRun(CCW,800); // 步進馬達 2 轉動：逆時針·速度 800 步 /s
  delay(3000);
  myStepper1.stepperStop(); // 步進馬達 1 停止
  myStepper2.stepperStop(); // 步進馬達 2 停止
  delay(1000);

  myStepper1.stepperRun(CCW,200); // 步進馬達 1 轉動：逆時針·速度 200 步 /s
  myStepper2.stepperRun(CW,800); // 步進馬達 2 轉動：順時針·速度 800 步 /s
  delay(3000);

  myStepper1.stepperStop(); // 步進馬達 1 停止
  myStepper2.stepperStop(); // 步進馬達 2 停止
  delay(1000);
}
```

Copyright© 2023 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。