BMduino-Shield
Dual-channel Motor Driver

BMP73T102
Arduino Library V1.0.1 Description

Revision: V1.01 Date: September 12, 2023

www.bestmodulescorp.com

Contents

Introduction

The BMP73T102 is a shield board for dual-channel motor driver from Best Modules, which is controlled directly by pins. This document provides the description of the BMP73T102 Arduino Lib functions and how to install the Arduino Lib. The example demonstrates the functions of driving DC motor and stepper motor.

Arduino Lib Functions

Arduino Lib Name: BMP73T102		Lib Version: V1.0.1
Constructors & Initialisation		
1	BMP73T102()	
	Description	DC motor constructor function
	Parameter	—
	Return Value	—
	Note	D9 and D6 pins control DC motor M1, D5 and D3 pins control DC motor M2
2	BMP73T102(uint8_t interface, uint8_t pin1=9, uint8_t pin2=6, uint8_t pin3=5, uint8_t pin4=3)	
	Description	Stepper motor constructor function, configure the drive mode and control interface pin
	Parameter	Interface: drive mode 0x01 (FULL4WIRE): full-step drive 0x02 (HALF4WIRE): half-step drive pin1: connection port, when the shield board is directly plugged in, it defaults to D9 pin control pin2: connection port, when the shield board is directly plugged in, it defaults to D6 pin control pin3: connection port, when the shield board is directly plugged in, it defaults to D5 pin control pin4: connection port, when the shield board is directly plugged in, it defaults to D3 pin control
	Return Value	—
	Note	When the shield board is not directly plugged in, pin1, pin2, pin3 and pin4 can be changed according to requirements
3	void begin()	
	Description	Initialise the shield board, pin is set as an output and to a low level
	Parameter	—
	Return Value	void
	Note	—
DC Motor Performance Functions		
4	void dcMotorRun(uint8_t ch, int8_t rank)	
	Description	Run the DC motor
	Parameter	ch: DC motor 1: DC motor 1 2: DC motor 2 rank: speed rank, the range is [-100, 100], the positive value represents clockwise, the negative value represents counter-clockwise, the absolute value represents speed, the greater the value, the faster the motor speed
	Return Value	void
	Note	—

5	void dcMotorStop(uint8_t ch)	
	Description	Stop the DC motor
	Parameter	ch: DC motor 1: DC motor 1 2: DC motor 2
	Return Value	void
	Note	—
6	void dcMotorBrake(uint8_t ch)	
	Description	Brake the DC motor
	Parameter	ch: DC motor 1: DC motor 1 2: DC motor 2
	Return Value	void
	Note	—
DC Motor Acquisition Functions		
7	int8_t getDcMotorRank(int8_t ch)	
	Description	Get the DC motor speed rank
	Parameter	ch: DC motor 1: DC motor 1 2: DC motor 2
	Return Value	Speed rank, the range is [-100~100]
	Note	Positive and negative represent direction: the positive value represents clockwise, the negative value represents counter-clockwise
Stepper Motor Performance Functions		
8	void stepperMoveTo(int32_t absolute, uint16_t speed)	
	Description	The stepper motor rotates to the target position at the set speed
	Parameter	absolute: target position (step), a position relative to zero position speed: stepper motor speed (step/s)
	Return Value	void
	Note	Stepper motor speed recommended value: No greater than 800 step/s is recommended(1) for full-step No greater than 1600 step/s is recommended(1) for half-step This function will run the “block” program. The Arduino will not continue executing the subsequent program content until the target position is reached.
9	void stepperMoveTo(int32_t absolute)	
	Description	The stepper motor rotates to the target position with acceleration
	Parameter	absolute: target position (step), a position relative to zero position
	Return Value	void
	Note	Note: This function will run the “block” program. The Arduino will not continue executing the subsequent program content until the target position is reached.
10	void stepperMove(int32_t relative, uint16_t speed)	
	Description	The stepper motor rotates the target step number at the set speed
	Parameter	relative: the stepper motor rotates the target step number (step), the positive value represents clockwise, the negative value represents counter-clockwise, the absolute value represents the step number speed: stepper motor speed (step/s)
	Return Value	void
	Note	Stepper motor speed recommended value: No greater than 800 step/s is recommended(1) for full-step No greater than 1600 step/s is recommended(1) for half-step Note: This function will run the “block” program. The Arduino will not continue executing the subsequent program content until the target position is reached.

11	void stepperMove(int32_t relative)	
	Description	The stepper motor rotates the target step number with acceleration
	Parameter	relative: the stepper motor rotates the target step number (step), the positive value represents clockwise, the negative value represents counter-clockwise, the absolute value represents the step number
	Return Value	void
	Note	Note: This function will run the “block” program. The Arduino will not continue executing the subsequent program content until the target position is reached.
Stepper Motor Parameter Acquisition & Configuration Functions		
12	uint16_t getStepperMaxSpeed()	
	Description	Get the stepper motor maximum speed
	Parameter	—
	Return Value	Stepper motor maximum speed (step/s)
	Note	—
13	uint16_t getStepperAcceleration ()	
	Description	Get the stepper motor acceleration
	Parameter	—
	Return Value	Stepper motor acceleration (step/s^2)
	Note	—
14	int32_t getStepperPosition()	
	Description	Get the stepper motor current position
	Parameter	—
	Return Value	Stepper motor current position (step)
	Note	—
15	void setStepperMaxSpeed(uint16_t maxSpeed)	
	Description	Set the stepper motor maximum speed
	Parameter	maxSpeed: stepper motor maximum speed (step/s)
	Return Value	void
	Note	Stepper motor maximum speed recommended value: No greater than 800 step/s is recommended(1) for full-step No greater than 1600 step/s is recommended(1) for half-step
16	void setStepperAcceleration(uint16_t acceleration)	
	Description	Set the stepper motor acceleration
	Parameter	acceleration: stepper motor acceleration (step/s^2)
	Return Value	void
	Note	—
17	void setStepperCurrentPosition(int32_t position)	
	Description	Set the stepper motor current position
	Parameter	position: position (step)
	Return Value	void
	Note	—

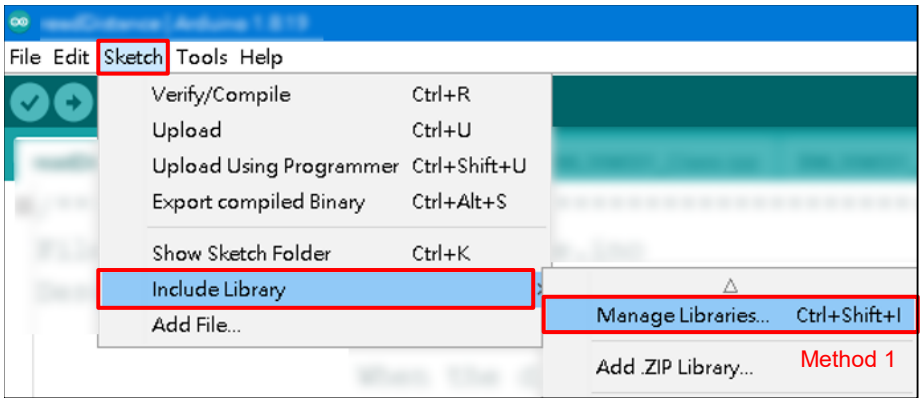
Note 1: Various stepper motors in different voltages can achieve a different maximum speed, so users need to adjust the motor speed according to the power supply to be used. For the recommended value here, the test condition is 12V motor power supply, the test motor is a stepper motor with 1.8° step angle.

Arduino Lib Download and Installation

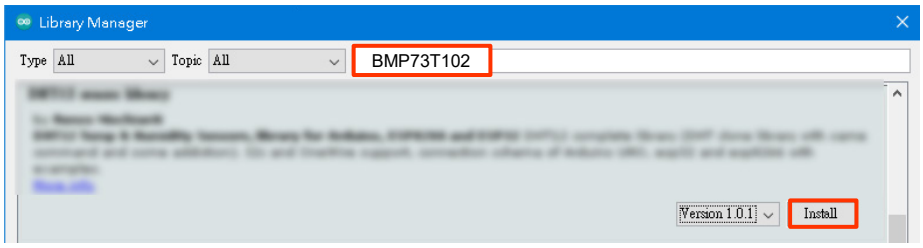
BMP73T102 Library: Refer to the following two methods to install the BMP73T102 Arduino Library.

Method 1: Search for installation

Search for installation: Arduino IDE→Sketch→Include Library→Manage Libraries...→Search BMP73T102→Install



Search for Installation Step 1

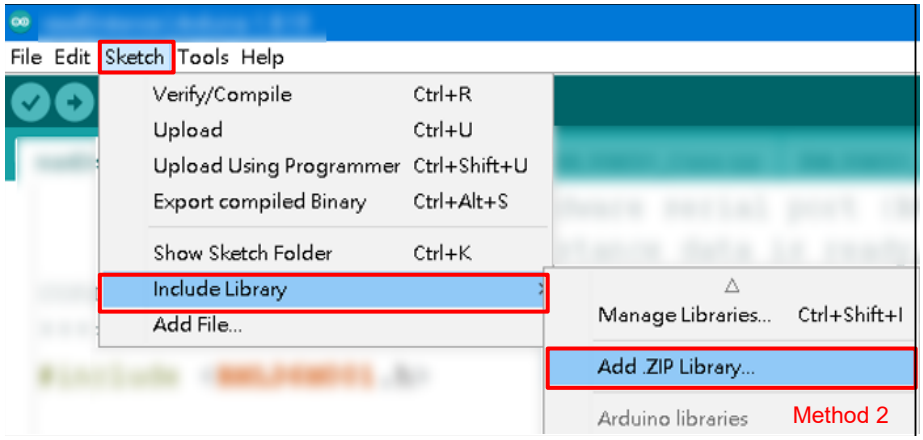


Search for Installation Step 2

Method 2: Download before adding a ZIP library

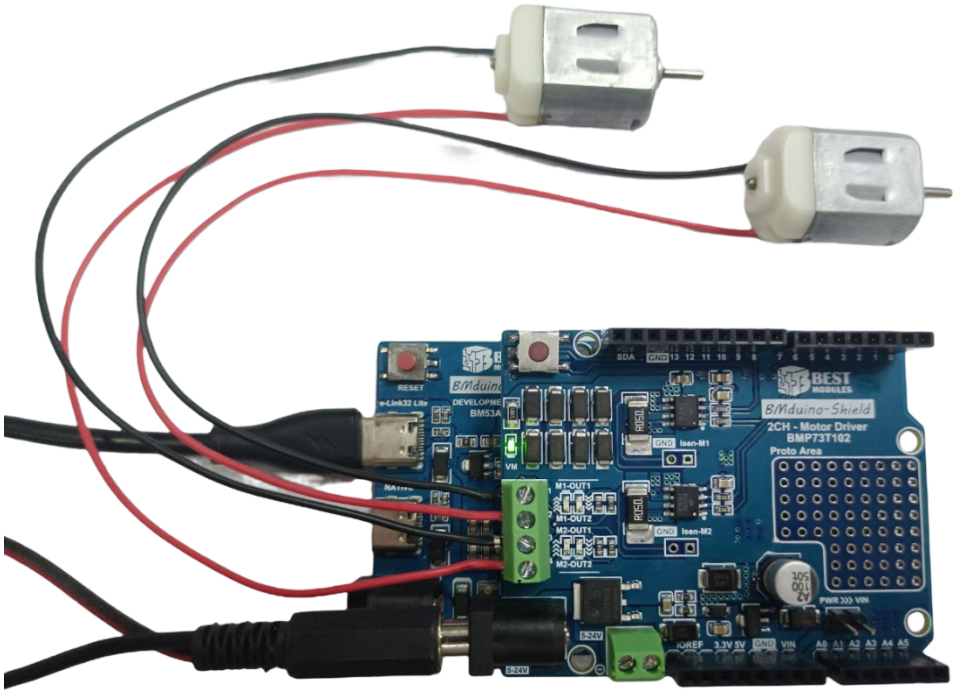
Download method: Open the Best Modules official website (https://www.bestmodulescorp.com/bmp73t102.html) and download the BMP73T102 Library from “Arduino example program” under the “DOCUMENTS” menu.

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



Arduino Examples

Example1: rotatingDcMotor



Physical Connection Diagram

Example function: demonstrate the function of driving the DC motor

1. The DC motor 1 rotates reversely at 50 rank speed and the DC motor 2 rotates forwardly at 80 rank speed, last for 3s.
2. The motor 1 and the motor 2 stop for 1s.

The DC motor repeats action 1 and 2.

1. Open the example:

   File→Examples→Select Lib (BMP73T102)→Select example (rotatingDcMotor)

2. Example description:

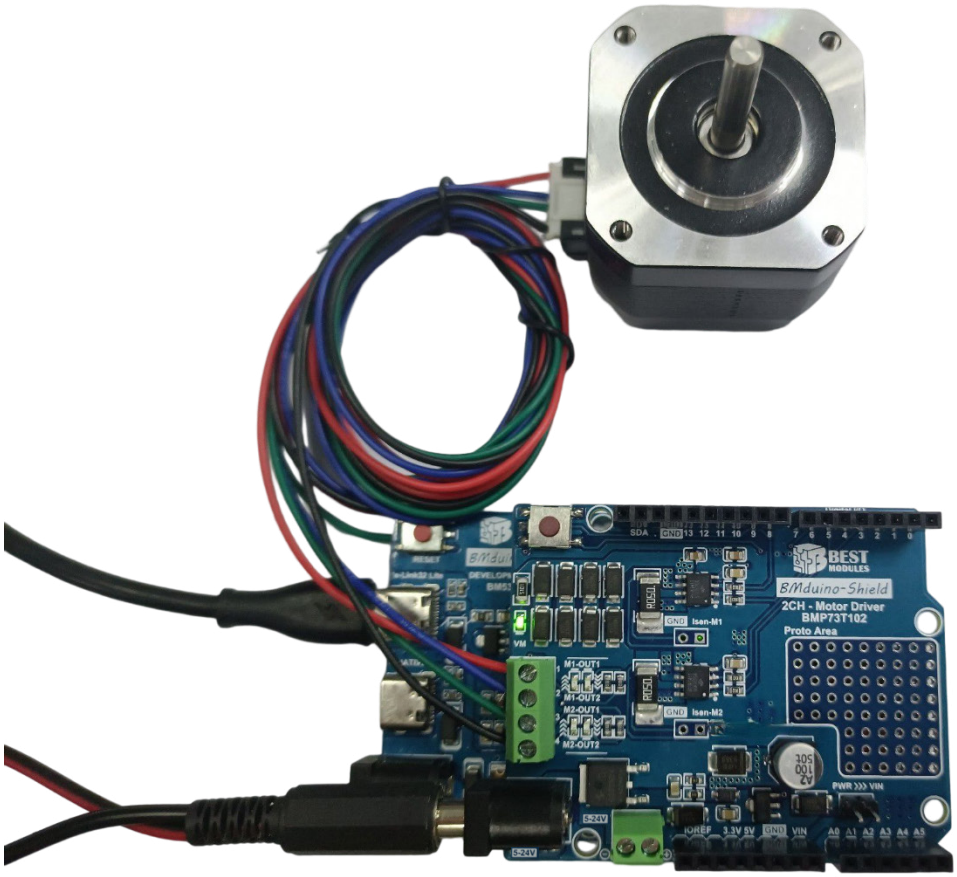
   a. Create object & initialise object, set parameter

```
#include <BMP73T102.h>
BMP73T102 dcmotor; // Create the DC motor object
void setup()
{
  dcmotor.begin(); // Initialise the object
}
```

b. Rotate the DC motor

```
void loop()
{
  dcmotor.dcMotorRun(1,-50);// The DC motor 1 rotates reversely at 50
                            // rank speed
  dcmotor.dcMotorRun(2,80); // The DC motor 2 rotates forwardly at 80
                            // rank speed
  delay(3000);              // The DC motor rotates for 3s
  dcmotor.dcMotorStop(1);   // The DC motor 1 stops
  dcmotor.dcMotorStop(2);   // The DC motor 2 stops
  delay(1000);              // The DC motor stops for 1s
}
```

Example2: rotatingStepperMotor



Physical Connection Diagram

Example function: demonstrate the function of driving the stepper motor

1. Set the stepper motor interface acceleration and maximum speed.
2. The stepper motor rotates 800 steps forwardly at the speed of 800 step/s.
3. The stepper motor rotates to the zero position with acceleration/deceleration.

Thr stepper motor repeats action 2 and 3.

1. Open the example:

   File→Examples→Select Lib (BMP73T102)→Select example (rotatingStepperMotor)

2. Example description:

   a. Create object & initialise object, set parameter

```
#include "BMP73T102.h"
BMP73T102 MyStepper(2);    // Create the object, set to operate in the
                           // half-step drive mode
void setup()
{
  MyStepper.begin();       // Initialise the object
  MyStepper.setStepperMaxSpeed(1000); // Set the stepper motor maximum
                                      // speed to be 1000 step/s
  MyStepper.setStepperAcceleration(400);// Set the stepper motor
                                        // acceleration to be 400
                                        // step/s²
}
```

   b. Rotate the stepper motor

```
void loop()
{
  MyStepper.stepperMove(800, 800);// The stepper motor runs 800 steps
                                  // at the speed of 800 step/s
  delay (1000);                   // Wait for 1s
  MyStepper.stepperMoveTo(0);// The stepper motor runs to the zero
                             // position at the acceleration of 400
                             // step/s²
  delay (1000);              // Wait for 1s
}
```