

轮 趣 科 技

双通道直流有刷电机驱动器 D50A 用户手册

推荐关注我们的公众号获取更新资料



版本说明:

版本	日期	内容说明
1.0	2022/12/16	第一次发布

网址: www.wheeltec.net

序言

我们将通过这篇教程与大家一起学习 12A 双通道直流有刷电机驱动器 D50A。
下文用 D50A 直接替代双通道直流有刷电机驱动器。

目录

1. 驱动器介绍	4
1.1 功能特性	4
1.2 应用场合	4
1.3 综合描述	4
1.4 原理框图	5
2. 引脚说明与运行逻辑	6
2.1 引脚说明	6
2.2 运行逻辑	7
3. 驱动参数与应用	8
3.1 工作参数	8
3.2 结构参数	9
3.3 典型应用	9
4. 代码例程	11
4.1 例程的接线	11
4.2 代码	12

1. 驱动器介绍

1.1 功能特性

- 更小的尺寸，50mm×50mm×12.5mm
- 内置过电压、欠电压、过热保护电路
- 全部控制信号电气隔离，充分兼容 3-5V 电压输入
- 高速 PWM 隔离输入，隔离带宽>50MHz
- 支持满占空比输入，可脱离 MCU，实现外部开关直接控制，可串联限位开关
- 驱动器运行电压 6.5-33.8V，支持 7-24V 电压等级直流有刷电动机
- 每通道具有双色指示灯指示电机转向

1.2 应用场合

- 自动化机器人
- 机电一体化
- 设备制造
- 科研、生产
- 电子竞技

1.3 综合描述

D50A 专为驱动低压直流电机设计，每通道具有持续输出 12A 电流能力，可驱动最高 290W 电功率的直流电机。驱动器内器件时序高度优化，允许 PWM 输入最小脉宽低至 2us，充分保证 PWM 的动态调节范围，提高对电机的控制品质。板载保护电路，降低驱动器在异常工作条件下受损的可能，保护状态由指示灯实时输出。全电气隔离输入增强了主控 MCU 电路安全性，更可显著提高系统电磁兼容性能。

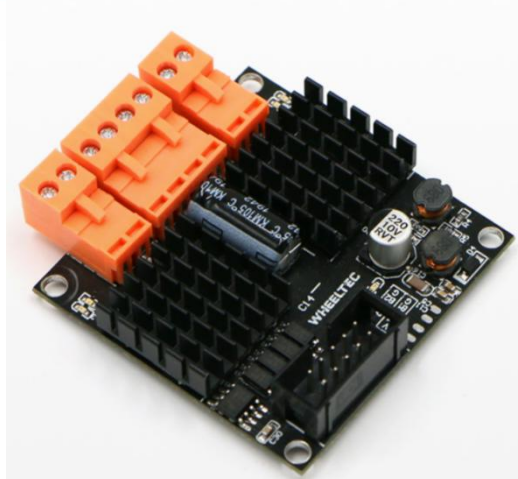


图 1-3-1 实物图

1.4 原理框图

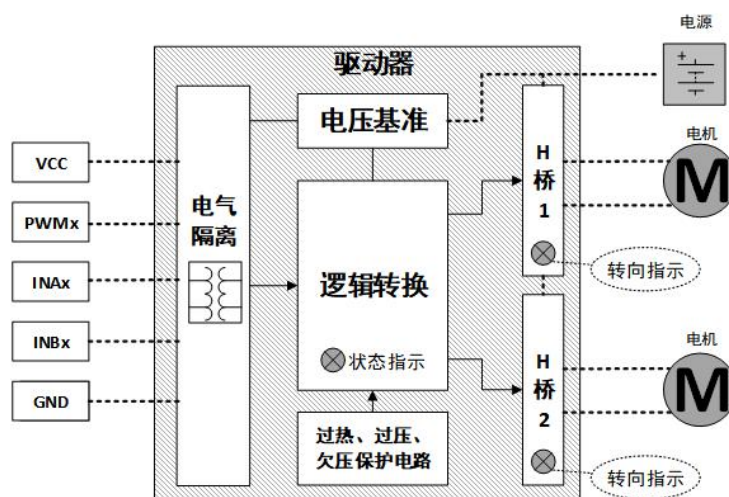


图 1-4-1 原理框图

2. 引脚说明与运行逻辑

2.1 引脚说明

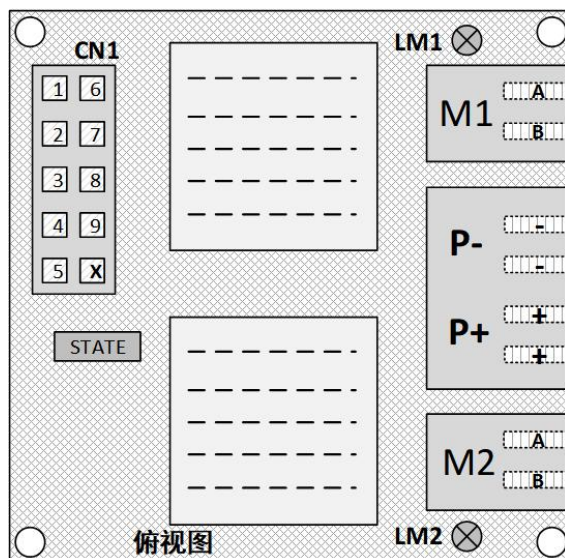


图 2-1-1 驱动器引脚和指示灯详情图

表 2-1-1 输入引脚详情表

CN1	定义	描述
1	VCC	隔离正电源输入，兼容 3.3V、5V 电源
2	PWM1	M1 通道占空比调制输入，高速隔离，带宽 50MHz
3	INA1	M1 通道控制逻辑输入 A
4	INB1	M1 通道控制逻辑输入 B
5	GND	隔离电源地输入
6	VCC	隔离正电源输入，兼容 3.3V、5V 电源
7	PWM2	M2 通道占空比调制输入，高速隔离，带宽 50MHz
8	INA2	M2 通道控制逻辑输入 A
9	INB2	M2 通道控制逻辑输入 B
X	GND	隔离电源地输入

注： 1. 输入信号悬空时为高电平。
2. 控制输入端子 CN1 为 2.54mm 间距 2x5 双排针。

表 2-1-2 功率接口详情表

端子	定义
M1_A	电机通道 1 的输出引脚 A
M1_B	电机通道 1 的输出引脚 B
M2_A	电机通道 2 的输出引脚 A
M2_B	电机通道 2 的输出引脚 B
P-	功率电源负极输入
P+	功率电源正极输入

表 2-1-3 指示灯状态对应表

指示灯	状态说明		
STATE	运行状态指示 灯	常亮	正常运行
		快闪	电源过电压，输出关闭
		慢闪	电源欠电压，输出关闭
		双闪	驱动过热，输出关闭
LM1	电机通道 1 转向指示，蓝色正转，红色反转		
LM2	电机通道 2 转向指示，蓝色正转，红色反转		

2.2 运行逻辑

表 2-2-1 驱动的运行逻辑

信号输入			功率输出		
INAx	INBx	PWMx	Mx_A	Mx_B	电机状态
L	L	X	L	L	制动
L	H	PWM	PWM	L	正转
H	L	PWM	L	PWM	反转
H	H	X	Z	Z	脱机
驱动器发生过欠压保护、过热保护			Z	Z	脱机

注：

- 1) INAx、INBx、PWMx、Mx_A、Mx_B 中 x 为通道号，可为 1 或 2。
- 2) H 为高电平，L 为低电平，X 为电平无关，Z 为高阻态。
- 3) 在进行大能量正反转切换时应先 PWM 逐步调速至 0%，再进行切换，否则可能造成驱动器损坏。

注：输入信号悬空时为高电平

3. 驱动参数与应用

3.1 工作参数

表 3-1-1 驱动的工作参数详情表

参数项	条件	最小值	典型值	最大值	单位
功率电源电压 VP		6.5		32	V
过压保护值			33.8		V
欠压保护值			5.6		V
功率电源电流 IP	双端子接线			24	A
单通道持续输出电流 I _{mc}	VP=24V 阻性负载			12	A
单通道峰值输出电流 I _{mp}	VP=24V t = 100ms			70	A
单通道峰值输出电流 I _{mp}	VP=24V t = 10us			200	A
逻辑输入电压 V _{in}	与逻辑信号幅值相同	3		5.5	V
逻辑输入电流 I _{in}	VCC=5V		15		mA
PWM 输入频率			18	60	kHz
PWM 最小脉宽	正、负向脉冲		2		us
过热保护	传感器所在位置	85		95	°C
工作温度		-25		85	°C

注：

1. 除特殊说明外，本驱动器测试均在 20℃、开放环境下进行。
2. 双端子接线为 P+、P- 端子各两条电源线连接（因受限于端子容量）。
3. PWM 输入频率，建议高于 10kHz，低于该值时可能会有明显电流噪声，静音应用设定在 17-18kHz 左右为宜。

3.2 结构参数

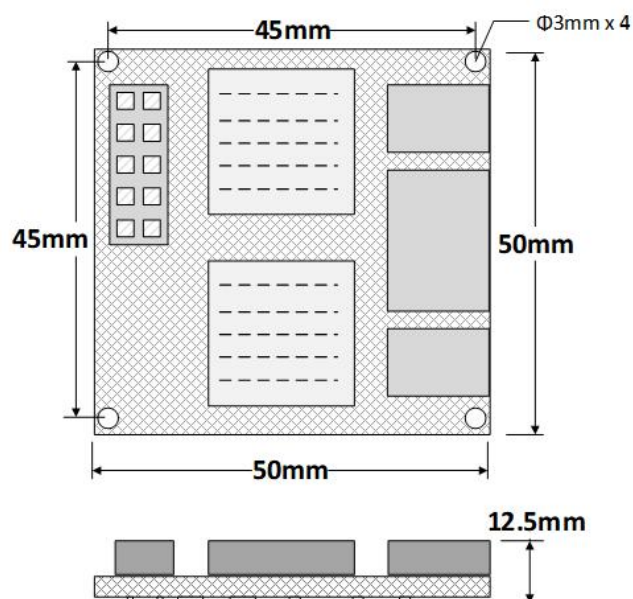


图 3-2-1 结构参数图

注：高度 12.5mm 为计入功率端子座总高，如考虑功率端子插头，则总高度为 15mm。

3.3 典型应用

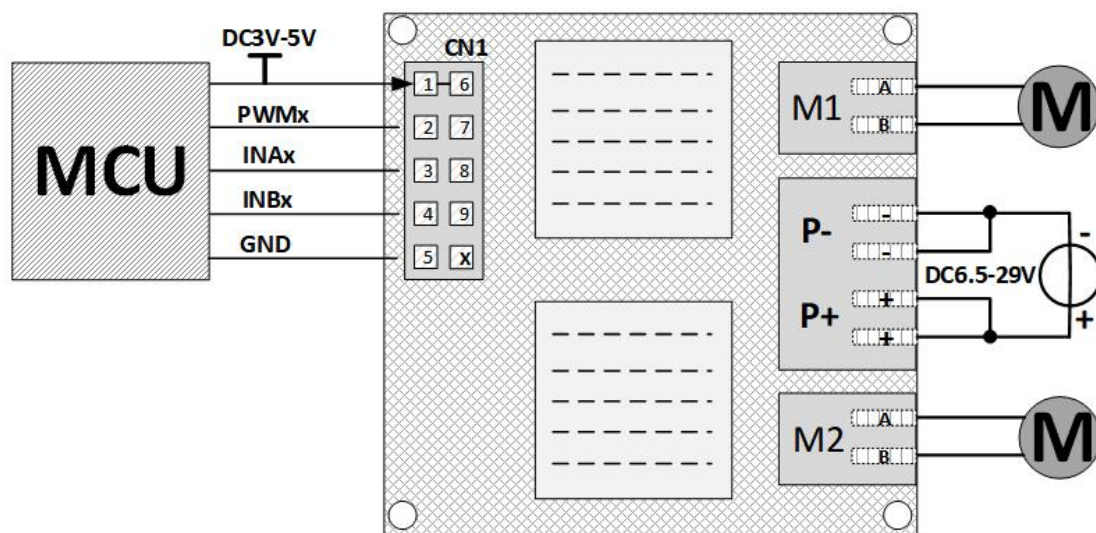


图 3-3-1MCU 直接控制电机运转

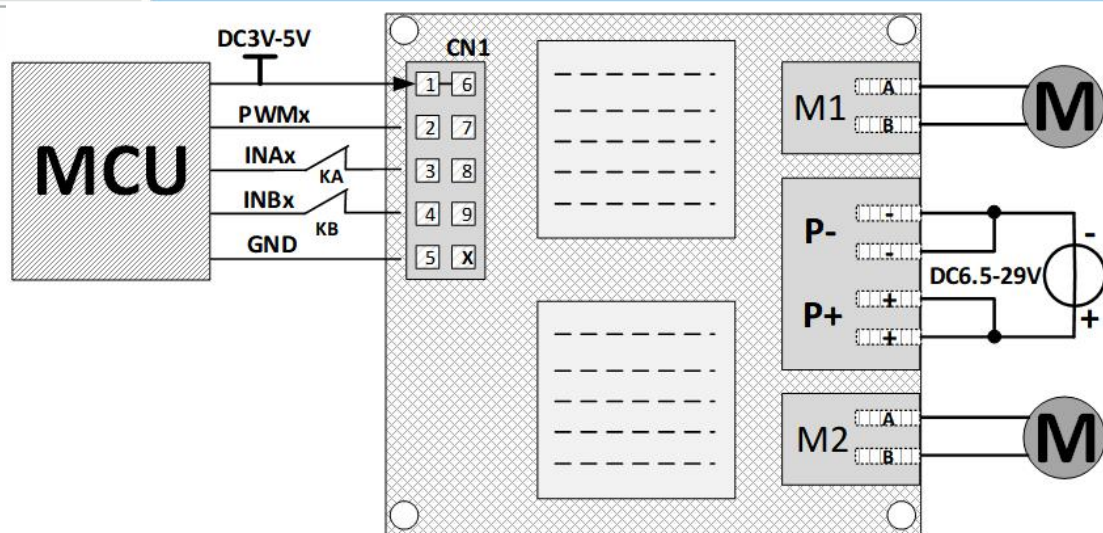


图 3-3-2 MCU 控制电机运转，加入限位开关功能（KA、KB 为限位常闭接点）

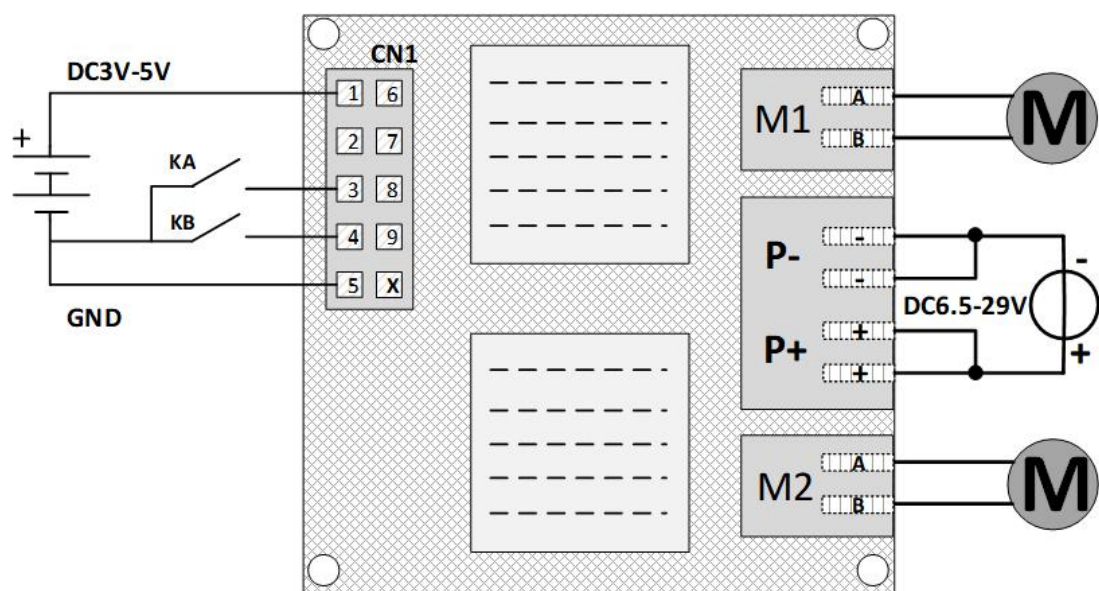


图 3-3-3 按键（KA、KB）控制电机正反转，无需 MCU 参与

- 注：
1. 功率电源输入大于 12A 时，电源端子必须使用双端接线。
 2. 为避免短路冲击电流的破坏，功率输入输出回路应增加熔断器保护。

4. PWM 调速控制教程

PWM 输入引脚如果占空比为 100% 时电机输出等于电源输入的电压，调整 PWM 占空比大小可以调整电机的输出。提供的接线说明都是基于本公司产品的代码例程，我们提供了 STM32F103C8T6 核心板的代码例程一份。如果使用的不是本公司的核心板，那就需要自行摸索一下，但也只需要改动引脚，使用和本例程的相同功能的引脚即可。

4.1 例程的接线

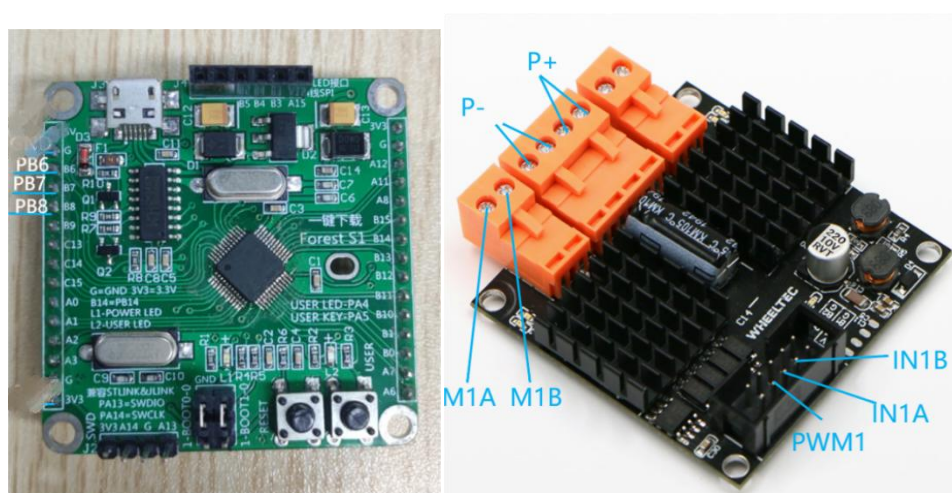


图 4-1-1 各模块引脚图

表 4-1-1 例程接线详情表

STM32 主控（主控由外部供电）	驱动	电机	
PB6	PWM1		
PB7	IN1A		
PB8	IN1B		
	M1A	电机线+	
	M1B	电机线-	
	P+		外部电源+（7V~24V）
	P-		外部电源-

4.2 代码

```
void Gpio_Init(void)
{
    GPIO_InitTypeDef GPIO_InitStructure;           //定义结构体GPIO_InitStructure

    RCC_APB2PeriphClockCmd( RCC_APB2Periph_GPIOB, ENABLE); // 使能PB端口时钟
    GPIO_InitStructure.GPIO_Pin = GPIO_Pin_7| GPIO_Pin_8; //PB7 PB8
    GPIO_InitStructure.GPIO_Mode = GPIO_Mode_Out_PP;      //推挽，增大电流输出能力
    GPIO_InitStructure.GPIO_Speed = GPIO_Speed_50MHz;     //IO口速度
    GPIO_Init(GPIOB, &GPIO_InitStructure);             //GPIOB初始化
}
```

图 4-2-1 电机接口初始化

电机接口的初始化，也就是对 IN1A、IN1B 这两个接口初始化，使它输出高低电平，从而控制电机正反转。

```
void pwm_int(void)
{
    GPIO_InitTypeDef GPIO_InitStructure;           //定义结构体GPIO_InitStructure
    TIM_TimeBaseInitTypeDef TIM_TimeBaseStructure; //定义结构TIM_TimeBaseStructure
    TIM_OCInitTypeDef TIM_OCInitStructure;         //定义结构TIM_OCInitStructure

    RCC_APB1PeriphClockCmd(RCC_APB1Periph_TIM4, ENABLE); //使能定时器4时钟
    RCC_APB2PeriphClockCmd(RCC_APB2Periph_GPIOB, ENABLE); //使能PB端口时钟

    GPIO_InitStructure.GPIO_Pin = GPIO_Pin_6;      //PB6
    GPIO_InitStructure.GPIO_Speed = GPIO_Speed_50MHz; //IO口速度
    GPIO_InitStructure.GPIO_Mode = GPIO_Mode_AF_PP; //复用模式输出
    GPIO_Init(GPIOB, &GPIO_InitStructure);         //GPIOB初始化

    TIM_TimeBaseStructure.TIM_Period = 7199;       //设置在下一个更新事件装入活动的自动重装载寄存器周期的值
    TIM_TimeBaseStructure.TIM_Prescaler = 0;       //设置用来作为TIMx时钟频率除数的预分频值 不分频
    TIM_TimeBaseStructure.TIM_ClockDivision = 0;   //设置时钟分割:TDTS = Tck_tim
    TIM_TimeBaseStructure.TIM_CounterMode = TIM_CounterMode_Up; //TIM向上计数模式
    TIM_TimeBaseInit(TIM4, &TIM_TimeBaseStructure); //根据TIM_TimeBaseInitStruct中指定的参数初始化TIMx的时间基数单位

    TIM_OCInitStructure.TIM_OCMode = TIM_OCMode_PWM1; //PWM1脉冲宽度调制模式2
    TIM_OCInitStructure.TIM_Pulse = 0;             //输入脉冲捕获值
    TIM_OCInitStructure.TIM_OCPolarity = TIM_OCPolarity_High; //设置TIM输出比较极性为高
    TIM_OCInitStructure.TIM_OutputState = TIM_OutputState_Enable; //比较输出使能
    TIM_OCInit(TIM4, &TIM_OCInitStructure);        //根据TIM_OCInitStructure中指定的参数初始化外设TIM4

    TIM_CtrlPWMOutputs(TIM4, ENABLE);              //主输出使能

    TIM_OC1PreloadConfig(TIM4, TIM_OCPreload_Enable); //使能预装载寄存器

    TIM_ARRPreloadConfig(TIM4, ENABLE);            //使能自动装载允许位

    TIM_Cmd(TIM4, ENABLE); //启动定时器TIM4
}
```

图 4-2-2 PWM 初始化

图 4-2-2 是对 PWM 外设的初始化，用的是定时器 4。定时器周期建议使用 10KZ 以上，调速效果会更平滑

```
void moto(int mode)
{
    if(mode==1)
    {
        GPIO_SetBits(GPIOB, GPIO_Pin_7);    // 高电平
        GPIO_ResetBits(GPIOB, GPIO_Pin_8);  // 低电平

        TIM_SetCompare1(TIM4, 3000);        //设置TIM4通道1的占空比 3000/7200
    }
    if(mode==0)
    {
        GPIO_SetBits(GPIOB, GPIO_Pin_8);    // 高电平
        GPIO_ResetBits(GPIOB, GPIO_Pin_7);  // 低电平

        TIM_SetCompare1(TIM4, 4000);        //设置TIM4通道1的占空比 4000/7200
    }
}
```

图 4-2-3 控制电机转向控制

这里的转向控制的代码和我们上面的逻辑表是一样的，必须一个高电平一个低电平并且需要 PWM 的输入，电机才能转动起来。图 5.1-3 中的 IN1A 为 0 时，IN1B 为 1 时为正转，IN1A 为 1、IN1B 为 0 时为正转，PWM 是调速的，如图 4-2-3 所示，我们设置的 PWM 占空比为 7200，当占空比为 7200 时，转速为满转。