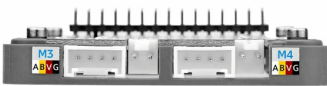
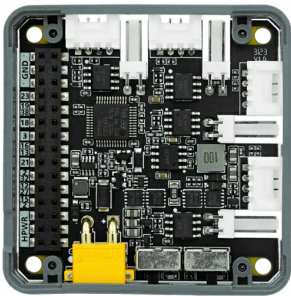
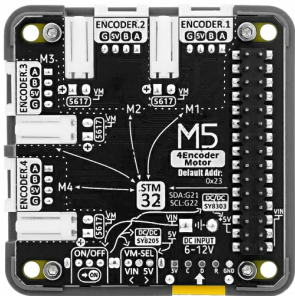
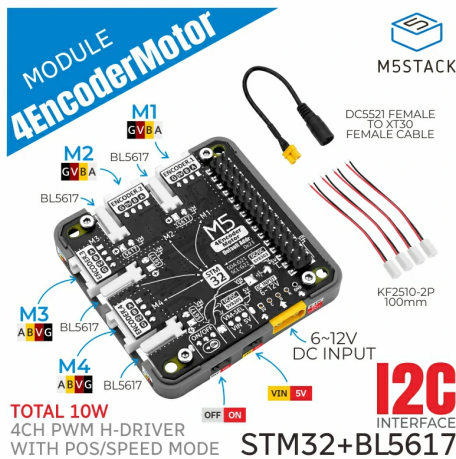


Module 4EncoderMotor

SKU:M138



描述

Module 4EncoderMotor 是一款 4 通道编码电机驱动模块，采用 **STM32 + BL5617 H 桥驱动 IC** 方案。以 I2C 通信方式，支持从机地址修改，提供灵活的控制方式。通过 AB 脉冲编码信号输入，实现精准的电机运动状态和位置检测。支持占空比控制、绝对位置定位和速度调节等模式，实现电机的正转、反转、停止和制动等多种功能。集成 INA199 电源监控，实时监测电压和电流状态。板载电源输入开关，可选择 DC 5V 或外部 DC 6 ~ 12V 电源输入。适用于机器人运动控制、自动化设备、智能车辆、实验室设备和工业自动化系统等领域。

产品特性

- 4 通道编码电机驱动
- AB 脉冲信号输入
- 占空比，绝对位置定位，速度调节控制模式
- I2C 通讯方式
- 电源电流电压监控

包装内容

- 1 x Module 4EncoderMotor
- 1 x DC5521 Female To XT30 Female 线缆
- 4 x KF2510-2P 100mm

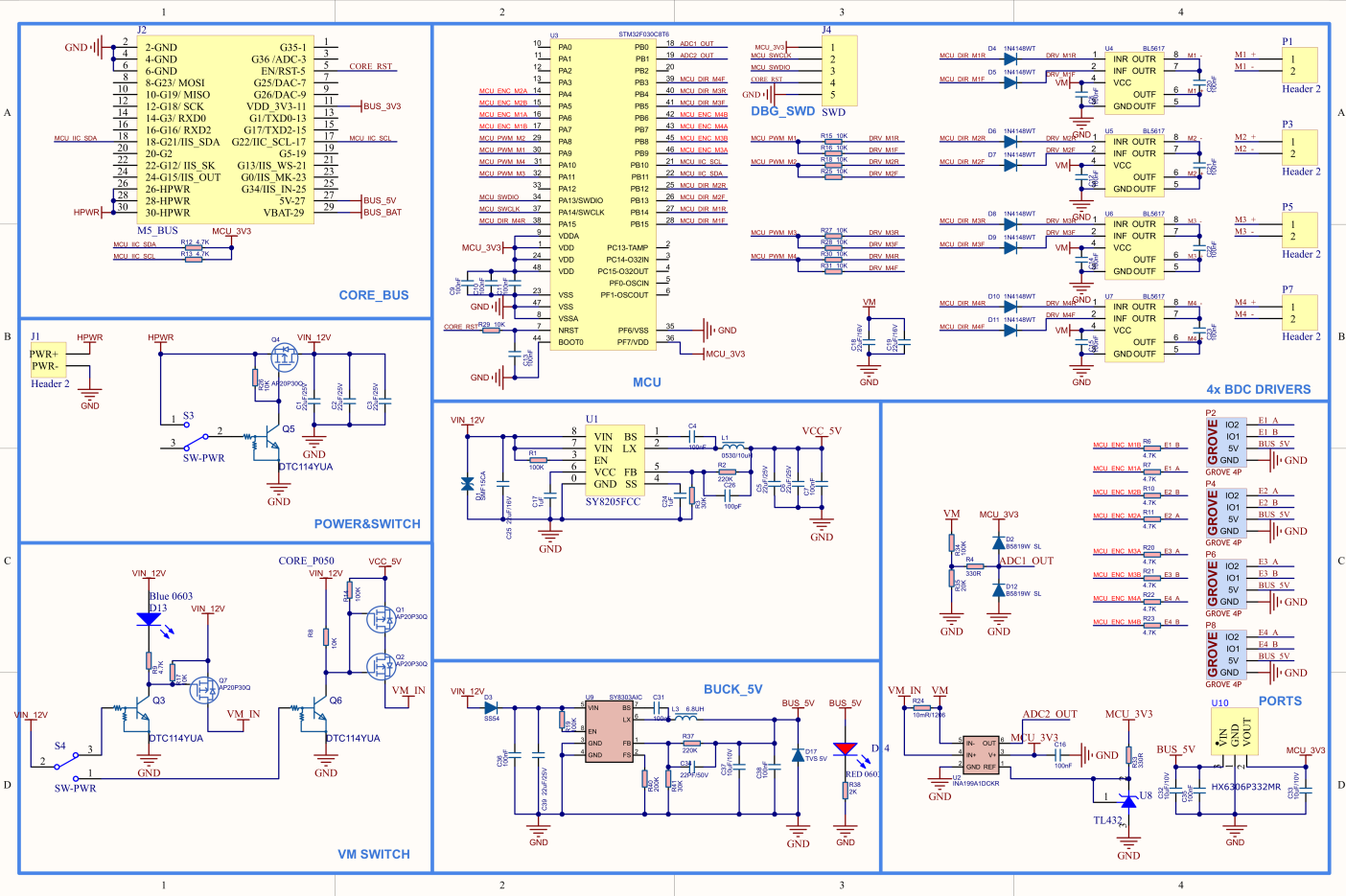
应用场景

- 机器人运动控制
- 自动化设备
- 智能车辆
- 工业自动化系统

规格参数

规格	参数
MCU	STM32F030C8T6
编码电机驱动 IC	BL5617
支持最大电流	3.0A
功率	最大 10W
外接 DC 电源	6 ~ 12V
通信接口	I2C 通信 @ 0x24
工作温度	0°C ~ 40°C
产品尺寸	54.0 x 54.0 x 13.1mm
产品重量	15.8g
包装尺寸	95.0 x 66.0 x 26.0mm
毛重	45.7g

原理图



STM32	PA9	PA8	PA11	PA10
BL5617 (PWM)	MCU_PWM_M1	MCU_PWM_M2	MCU_PWM_M3	MCU_PWM_M4

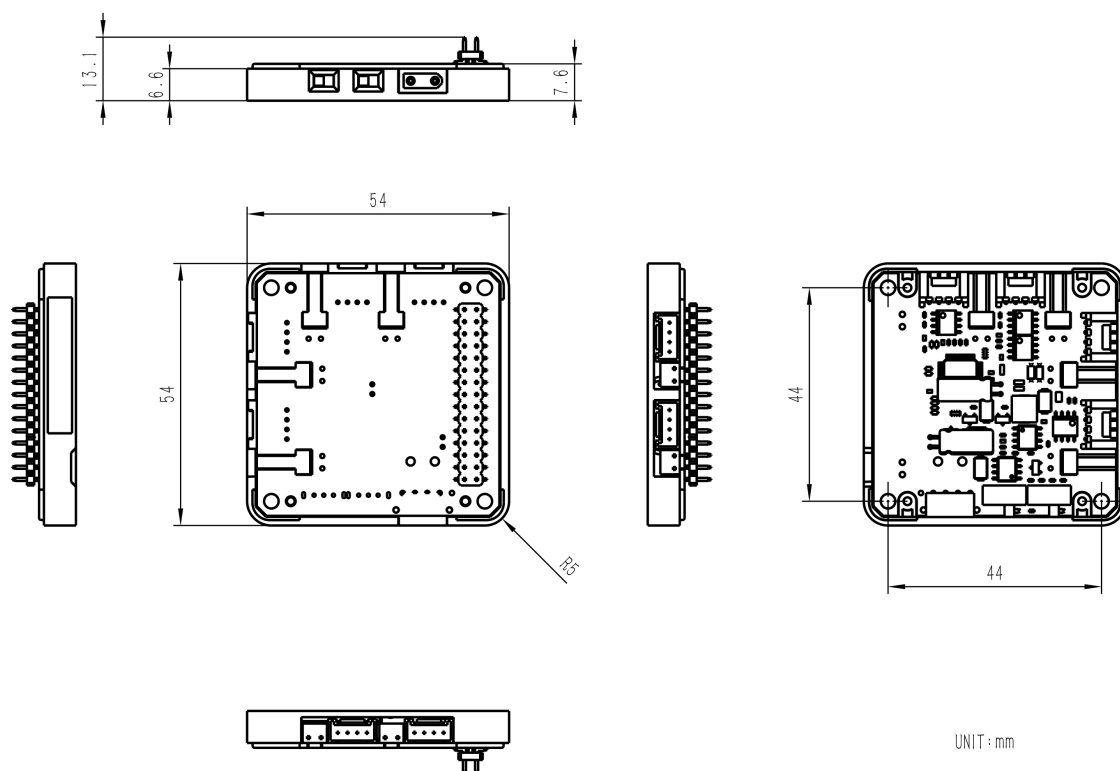
A/B 信号检测引脚

STM32	PA6/PA7	PA4/PA5	PB9/PB8	PB7/PB6
Encoding motors	E1_A/E1_B	E2_A/E2_B	E3_A/E3_B	E4_A/E4_B

M5-Bus

PIN	LEFT	RIGHT	PIN
GND	1	2	
GND	3	4	
GND	5	6	RST
	7	8	
	9	10	
	11	12	
	13	14	
	15	16	
SDA	17	18	SCL
	19	20	
	21	22	
	23	24	
HPWR	25	26	
HPWR	27	28	5V
HPWR	29	30	BAT

尺寸图



数据手册

- [BL5617 Datasheet](#)

软件开发

Arduino

- [Module 4EncoderMotor Arduino 驱动库](#)
- [Module 4EncoderMotor Example with Basic](#)
- [Module 4EncoderMotor Example with Core2](#)
- [Module 4EncoderMotor Example with CoreS3](#)

UiFlow1

- [Module 4EncoderMotor UiFlow1 文档](#)

UiFlow2

- [Module 4EncoderMotor UiFlow2 文档](#)

Internal Firmware

- [Module 4EncoderMotor 内置固件](#)

通信协议

Module 4EncoderMotor I2C 通信协议

M5Stack Module 4EncodeMotor V1.1 I2C Protocol																		V3 (FW Version)	
REG MAP (Addr:0x24)		0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F	note	
Motor PWM Duty INT8	0x20 R/W	Motor1_PWM_Duty	Motor2_PWM_Duty	Motor3_PWM_Duty	Motor4_PWM_Duty													-127 ~ 127	
Motor Encoder INT32	0x30 R/W	Motor1_Encoder-byte0	Motor1_Encoder-byte1	Motor1_Encoder-byte2	Motor1_Encoder-byte3	Motor2_Encoder-byte0	Motor2_Encoder-byte1	Motor2_Encoder-byte2	Motor2_Encoder-byte3	Motor3_Encoder-byte0	Motor3_Encoder-byte1	Motor3_Encoder-byte2	Motor3_Encoder-byte3	Motor4_Encoder-byte0	Motor4_Encoder-byte1	Motor4_Encoder-byte2	Motor4_Encoder-byte3	Motor Encoder = Motor Encoder-byte0 + Motor Encoder-byte1 * 256 + Motor Encoder-byte2 * 65536 + Motor Encoder-byte3 * 16777216	
Motor Speed INT8	0x40 R	Motor1_Speed	Motor2_Speed	Motor3_Speed	Motor4_Speed													-127 ~ 127 Motor encoder increments every 20 ms	
Motor1 Mode ^[1]	0x50 R/W	Motor1_Mode	Motor1_Position_P	Motor1_Position_I	Motor1_Position_D	Motor1_Position_Point-Byte0	Motor1_Position_Point-Byte1	Motor1_Position_Point-Byte2	Motor1_Position_Point-Byte3	Motor1_Position_MAX-Speed	Motor1_Speed_P	Motor1_Speed_I	Motor1_Speed_D	Motor1_Speed_Point					
Motor2 Mode ^[1]	0x60 R/W	Motor2_Mode	Motor2_Position_P	Motor2_Position_I	Motor2_Position_D	Motor2_Position_Point-Byte0	Motor2_Position_Point-Byte1	Motor2_Position_Point-Byte2	Motor2_Position_Point-Byte3	Motor2_Position_MAX-Speed	Motor2_Speed_P	Motor2_Speed_I	Motor2_Speed_D	Motor2_Speed_Point					
Motor3 Mode ^[1]	0x70 R/W	Motor3_Mode	Motor3_Position_P	Motor3_Position_I	Motor3_Position_D	Motor3_Position_Point-Byte0	Motor3_Position_Point-Byte1	Motor3_Position_Point-Byte2	Motor3_Position_Point-Byte3	Motor3_Position_MAX-Speed	Motor3_Speed_P	Motor3_Speed_I	Motor3_Speed_D	Motor3_Speed_Point					
Motor4 Mode ^[1]	0x80 R/W	Motor4_Mode	Motor4_Position_P	Motor4_Position_I	Motor4_Position_D	Motor4_Position_Point-Byte0	Motor4_Position_Point-Byte1	Motor4_Position_Point-Byte2	Motor4_Position_Point-Byte3	Motor4_Position_MAX-Speed	Motor4_Speed_P	Motor4_Speed_I	Motor4_Speed_D	Motor4_Speed_Point					
VIN Current Float (A)	0x90 R	current-byte0	current-byte1	current-byte2	current-byte3													float	
VIN Current X100 Int (A)	0xC0 R	VIN Current X100-byte0	VIN Current X100-byte1	VIN Current X100-byte2	VIN Current X100-byte3													VIN Current X100 Int = VIN Current X100-byte0 + VIN Current X100-byte1 * 256 + VIN Current X100-byte2 * 65536 + VIN Current X100-byte3 * 16777216	
VIN ADC 8bits ^[2]	0xA0 R	ADC Value 8bits																Vaule: 0~255	
VIN ADC 12bits ^[3]	0xB0 R	ADC Value 12bits-L	ADC Value 12bits-H															Vaule: 0~4095	
Encoder AB or BA	0xD0 R/W	Encoder AB or BA																Vaule: 0~1 0: AB(Default) 1: BA * Need to restart module to affect * Writing to this register will save the value to flash. Please do not write to this register frequently to prevent flash damage	
Soft start and stop	0xD0 R/W		Soft start and stop															Soft start and stop(0:disable, 1enable): bit0: Motor1 bit1: Motor2 bit2: Motor3 bit3: Motor4	
Firmware Version	0xF0 R															Version	Version: firmware version number		
I2C Address	0xF0 R/W															Address	Address: 1~127 Writing to this register will save the value to flash. Please do not write to this register frequently to prevent flash damage		

[1] (1)Mode:
0:Normal(Open loop)
1: Position Lock
2: Speed Lock
(2)P/I/D: 0~255
(3)Motor_Position_Point = Motor_Position_Point-byte0 + Motor_Position_Point-byte1 * 256 + Motor_Position_Point-byte2 * 65536 + Motor_Position_Point-byte3 * 16777216
(4)Motor_Position_MAX-Speed: -127 ~ 127
(5)Motor_Speed_Point: -127 ~ 127

[2] Voltage = ADC Value 8bits/255*3.3/0.16

[3] Voltage = ((ADC Value 12bits-L) * (ADC Value 12bits-H+1)*256)/255*3.3/0.16

Easyloader

Module 4EncoderMotor 内置固件升级程序 Easyloader。

Easyloader	下载链接	备注
Module 4EncoderMotor Firmware Upgrade Easyloader	download	/

| 相关视频

- Module 4EncoderMotor 功能介绍

[M138 4EncoderMotor Module 视频.mp4](#)