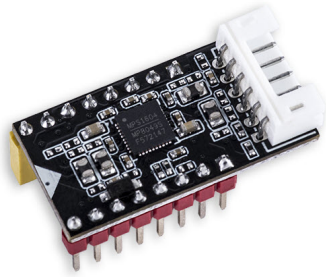


MegaPi Pro Encoder/DC Motor Driver Module



overview

This module is a dedicated module for MegaPi Pro, which can drive 2 DC motors or 1 coded motor. It adopts 2x8Pin plug-in mode and can be easily installed on MegaPi Pro. This module must be installed on the drive motor of the MegaPi Pro main control board. MegaPi Pro can install up to four of this module to drive 4 encoder motors (or 8 DC motors or 8 solenoid valves). The encoding port can be inserted into 25 photoelectric encoder motors, 180 photoelectric coding motors, the MegaPi Pro motor port driven by it can be inserted into 25 DC motors, 37 DC motors, and solenoid valves.

technical specifications

- Motor drive: MP80495
- Motor channel: 2
- Minimum working voltage: 6V
- Maximum working voltage: 12V
- Logic voltage: 5V
- Rated working current of each channel: 3A
- Peak operating current: 5.5A
- Module size: 30mmx15mm (length x width)

Features

- Support motors with a working voltage of 6~12V
- When powered by 12V power supply, the working current can reach 3A
- The module has over-voltage protection, over-current protection, and over-temperature protection to ensure the safety of use in all aspects
- Can drive 1 coded motor (white interface) or 2 DC motors (green interface)
- Colored male and female sockets to prevent wrong insertion
- Encoded motor drive end has echo electromotive force protection (TVS protection)
- The module is compact and easy to replace

programming guide

DC Motor Programming

- Arduino programming

If you use Arduino programming, you need to call the library `Makeblock-Library-master` to control the motor.

function	Function
MeDCMotor (uint8_port)	select interface
run(int16_t speed)	Set speed (-255~+255)
stop()	stop motor

programming example

After the following program is run, the DC motor of interface 1 of the four-way circuit module rotates at full speed for 1 second, stops for 1 second, and moves circularly.

```
#include <Arduino.h>
#include <Wire.h>
#include <SoftwareSerial.h>


#include <MeMegaPiPro.h>
MeDCMotor dcfourmotor_1(1);

void loop(){
    dcfourmotor_1.run(255);
    delay(1);
    dcfourmotor_1.run(0);
    delay(1);
    loop();
}
```

[Download sample program](#)

• mBlock programming

Block description

building blocks	illustrate
	Select interface (M9~M10); set speed (-255~+255).

After the following program runs, the four-way DC motor drive interface 1 motor rotates at full speed for 1 second, stops for 1 second, and repeats.



[Download sample program](#)

• mBlock programming

Block description

building blocks	illustrate
-----------------	------------

直流电机 M9 ▾ 以动力 50 %转动	Select interface (M9~M12); set parameters (-100~+100).
电磁阀 M9 ▾ 设置状态为 断电 ▾	选定接口 (M9~M12) ; 设定参数 (通电/断电) 。

以下程序运行后直流电机 M9 以 50% 动力运行，电磁阀 M10 为通电，1 秒后直流电机 M9 停止转动，电磁阀 M10 断电，重复此过程。

MegaPi Pro 主程序

重复执行

直流电机 M9 ▾ 以动力 50 %转动

电磁阀 M10 ▾ 设置状态为 通电 ▾

等待 1 秒

直流电机 M9 ▾ 以动力 0 %转动

电磁阀 M10 ▾ 设置状态为 断电 ▾

等待 1 秒

下载示例程序

● Python3 编程

- 1、主控板 MegaPi Pro 和树莓派连接。
- 2、树莓派安装最新的 makeblock 库 `pip3 install makeblock --upgrade`。
- 3、新建 Python 文件，后缀为 .py。
- 4、在 Python 文件里写入程序。
- 5、运行 Python文件，如“python123.py”。

语句说明

函数	功能
DCMotor(port)	创建直流电机对象。 port: MegaPiPro.M9 ~ MegaPiPro.M12
run(speed)	以百分比速度旋转。 speed: 速度百分比，范围：-100~+100

以下程序运行后 M9 直流电机将以 50% 的动力运行 2 秒，停止运动 1秒，以 -50% 的动力运行 2 秒，停止 1 秒，循环运动。

```
from time import sleep
from makeblock import MegaPiPro
board = MegaPiPro.create()
motor = board.DCMotor(MegaPiPro.M9)
while True:
    motor.run(50)
    sleep(2)
    motor.run(0)
    sleep(1)
    motor.run(-50)
    sleep(2)
    motor.run(0)
    sleep(1)
```

[下载示例程序](#)

编码电机编程

• Arduino编程

如果使用Arduino编程，需要调用库 `Makeblock-Library-master` 来控制电机。

以下程序运行后编码电机 1 和 2 将以 100 速度运行 1 秒，停止 1 秒，重复此过程。

```
#include <Arduino.h>
#include <Wire.h>
#include <SoftwareSerial.h>

#include <MeMegaPiPro.h>

double angle_rad = PI/180.0;
double angle_deg = 180.0/PI;
MeEncoderOnBoard Encoder_1(SLOT1);
MeEncoderOnBoard Encoder_2(SLOT2);
void isr_process_encoder1(void)
{
    if(digitalRead(Encoder_1.getPortB()) == 0){
        Encoder_1.pulsePosMinus();
    }else{
        Encoder_1.pulsePosPlus();
    }
}
void isr_process_encoder2(void)
{
    if(digitalRead(Encoder_2.getPortB()) == 0){
        Encoder_2.pulsePosMinus();
    }else{
        Encoder_2.pulsePosPlus();
    }
}

void setup(){
    TCCR1A = _BV(WGM10); //PIN12
    TCCR1B = _BV(CS11) | _BV(CS10) | _BV(WGM12);
    TCCR2A = _BV(WGM21) | _BV(WGM20); //PIN8
    TCCR2B = _BV(CS22);
    attachInterrupt(Encoder_1.getIntNum(), isr_process_encoder1, RISING);
    attachInterrupt(Encoder_2.getIntNum(), isr_process_encoder2, RISING);
}

void loop(){
    Encoder_1.setTarPWM(100);
    Encoder_2.setTarPWM(100);
    _delay(1);
    Encoder_1.setTarPWM(0);
    Encoder_2.setTarPWM(0);
    _delay(1);
    _loop();
}
```

下载示例程序

● mBlock编程

积木块说明

积木块	说明
	选择接口；设置动力（-255~+255）。
	选择接口；设置速度（最大速度由电机型号决定）。
	选择接口；设置旋转角度（非负整数）；设置速度（最大速度由电机型号决定）。

编码电机 接口1▼ 速度/圈每分钟	获取当前电机速度。
编码电机 接口1▼ 位置/角度	获取电机当前相对于零点的角度位置。

以下程序运行后编码电机 1 和 2 将以 100 速度运行 1 秒，停止 1 秒，重复此过程。



[下载示例程序](#)

• 慧编程编程

积木块说明（以MegaPi Pro主控为例）

积木块	说明
编码电机 接口1▼ 以动力 50 % 转动	选择接口；设置动力（-255~-255）。
编码电机 接口1▼ 以速度 30 转每分钟 转动	选择接口；设置速度（最大速度由电机型号决定）。
编码电机 接口1▼ 转动 360 度, 以速度 30 转每分钟	选择接口；设置旋转角度（非负整数）；设置速度（最大速度由电机型号决定）。
编码电机 接口1▼ 转速 (转每分钟)	获取当前电机速度。
编码电机 接口1▼ 角度位置 (度)	获取电机当前相对于零点的角度位置。

以下程序运行后编码电机 1 和 2 将以 100 速度运行 1 秒，停止 1 秒，重复此过程。



● Python3 编程

- 1、主控板 MegaPi Pro 和树莓派连接，180 光电编码电机和直流/编码驱动模块相连接。
- 2、树莓派安装最新的 makeblock 库 `pip3 install makeblock --upgrade`。
- 3、新建 Python 文件，后缀为 .py。
- 4、在 Python 文件里写入程序。
- 5、运行 Python文件，如 “python123.py”。

语句说明

函数	功能
EncoderMotor(port)	创建直流编码电机对象。 port: MegaPiPro.PORT1~MegaPiPro.PORT4
run(speed)	以指定速度旋转。 speed: 转速（范围为-180~+180）
move_to(position,speed,callback)	以指定速度旋转到指定位置。 position: 目标位置； speed: 转速（范围为-180~+180）； callback: 达到目标位置时触发回调
set_home()	设置当前位置为原点。

程序示例 1

After the following program runs, the encoder motor connected to MegaPi Pro DC/encoder motor drive interface 1 rotates at 50 speed for 2 seconds, stops for 1 second, rotates at -50 speed for 2 seconds, stops for 1 second, and repeats this process.

```
from time import sleep
from makeblock import MegaPiPro
board = MegaPiPro.create()
encoder = board.EncoderMotor(MegaPiPro.PORT1)
while True:
    encoder.run(50)
    sleep(2)
    encoder.run(0)
    sleep(1)
    encoder.run(-50)
    sleep(2)
    encoder.run(0)
    sleep(1)
```

[Download sample program](#)

Program example 2

After the following program runs, the encoder motor connected to MegaPi Pro DC/encoder motor drive interface 1 rotates to the target position.

```
from time import sleep
from makeblock import MegaPiPro
board = MegaPiPro.create()
encoder = board.EncoderMotor(MegaPiPro.PORT1)
position = 0
def on_finished(value):
    position = 5000 - position
    encoder.move_to(position,100,on_finished)

on_finished(position)
```

[Download sample program](#)

connection method

