ATOM PWM

SKU:K065

Description

The **ATOM PWM** is a single-channel PWM regulate DC Driver with built-in MOSFET load capacity up to 12V@100W, suitable for high-power DC motor PWM speed control and industrial heating wire control applications. Using ATOM LITE as the core controller (built-in ESP32), it has a wide PWM dynamic adjustment range (e.g., frequency of 5 kHz, duty cycle adjustment range 0-100%, resolution up to 13bit), and combined with the built-in WIFI & Bluetooth function, it can also easily achieve remote control. Supports UIFlow graphical programming for easy configuration of signal output and function expansion.

Product Features

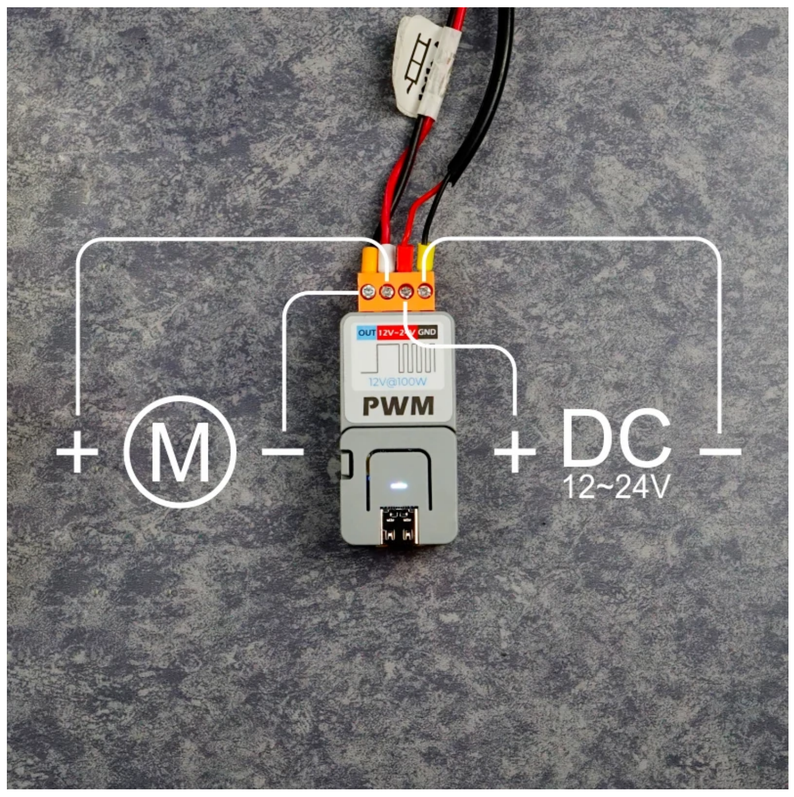
- Single channel low latency PWM signal output
- High power MOSFET, output capacity 12V@100W
- Reserved 1xGROVE expansion interface
- Built-in DC-DC (12V->5V) conversion circuit
- Easy installation, simple operation
- All-in-one design with protective cover
- Development platform: Arduino/UIFlow

Included

- 1x ATOM PWM
- 1x ATOM LITE
- 1x M2 Hex Wrench
- 1x M2*8 cup head Machine screw
- 1x 3.96-4P terminal
- 1x TYPE-C USB data cable (20cm)

Applications

- DC motor control
- Light control
- Heater control
- DC Load



Specifications

Specification	Parameter
Driver Chip	EG27324
MOSFET	FDD8447L
Maximum Output Power	100W
Input Voltage Range	DC 12V-24V
Number of Drive Channels	1
Net Weight	28.9g
Gross Weight	37.3g
Product Dimensions	24 * 48 * 18mm
Package Size	54 * 54 * 20mm

Common Frequency and Resolution

LEDC Clock Source	LEDC Output (PWM) Frequency	Highest Resolution
APB_CLK (80 MHz)	1 kHz	1/65536 (16 bit)
APB_CLK (80 MHz)	5 kHz	1/8192 (13 bit)
APB_CLK (80 MHz)	10 kHz	1/4096 (12 bit)
RTC8M_CLK (8 MHz)	1 kHz	1/4096 (12 bit)
RTC8M_CLK (8 MHz)	8 kHz	1/512 (9 bit)
REF_TICK (1 MHz)	1 kHz	1/512 (9 bit)

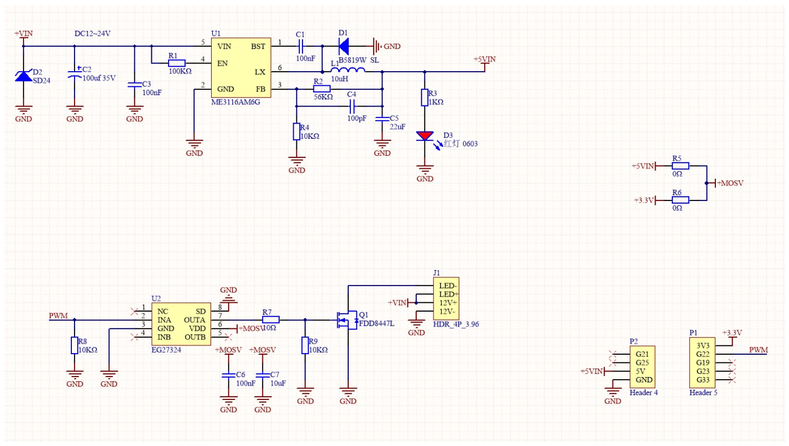
Related Links

- **Datasheet**
  - FDD8447L
  - EG27324
  - ME3116AM6G

Pin Mapping

ATOM	G22
EG27324	INA

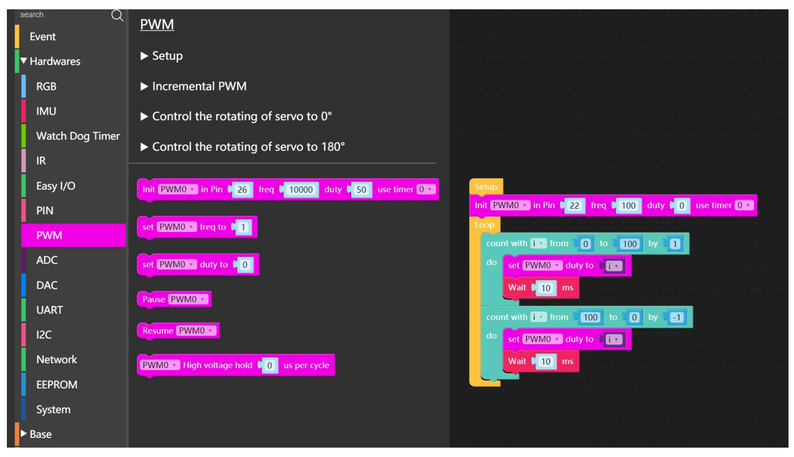
Schematic



Example

UIFlow

- View UIFlow-PWM control documentation



Arduino

```
#include <Arduino.h>

#define SIGNAL 22

int freq = 10000;
int ledChannel1 = 0;
int resolution = 10;
```

```
void setup() {
  ledcSetup(ledChannel1, freq, resolution);
  ledcAttachPin(SIGNAL, ledChannel1);
}

void loop() {

    for(int i=0; i < 500; i++){
      ledcWrite(ledChannel1, i);
      delay(2);
    }

    for(int i=500; i > 0; i--){
      ledcWrite(ledChannel1, i);
      delay(2);
    }
}
```

Video