

V1 to V2

Upgrading from V1 to V2

- [Drum Motor](#)

Drum Motor

How to connect and code for a nano or ESP32 to control the motor.

Motor support **Reversed running** and has more noise when reversed running than normal running, this is a normal phenomenon, please know before buying.

Note: Please Connect wire STRICTLY according to below method, Or motor will be burn and can't work. We can't take any responsible for wrong connection.

Connect red to the positive, connect black and blue to the negative the motor will run.

Connect red to the positive, connect black and blue and white to the negative. the motor will change the direction.

Black line: Connect the power supply negative

Red line: Connect the power supply positive

White line: CW / CCW conversion

Yellow line: FG signal line.

Blue line: PWM pulse width control line can be connected to 0-5V pulse width adjustment

Motor Wire	Arduino
Blue (PWM)	D6
White (DIR)	D7
Red (V+)	12-24V
Black (GND)	Shared GND with Arduino

1. Add These Pin Definitions at the Top:

```
const int drumPwmPin = 6; // PWM pin (connect to Blue wire)
const int drumDirPin = 7; // Direction pin (connect to White wire)
```

2. Add These to `setup()`: Make sure these lines are inside your `setup()` function

```
pinMode(drumPwmPin, OUTPUT);
pinMode(drumDirPin, OUTPUT);
```

3. Replace Your `handleDRUM()` With This Version:

```
void handleDRUM(uint8_t value) {
  if (value > 100) value = 100; // Clamp value to 100
  int pwmValue = map(value, 0, 100, 0, 255); // Scale 0-100 to 0-255

  if (value != 0) {
    digitalWrite(drumDirPin, HIGH); // Set motor direction (HIGH = CW)
    analogWrite(drumPwmPin, pwmValue); // Apply PWM speed
    setValue(&sendBuffer[drumByte], value); // Update protocol buffer
  } else {
    analogWrite(drumPwmPin, 0); // Stop motor
    setValue(&sendBuffer[drumByte], 0);
  }

  lastEventTime = micros();
}
```