

Vibration Motor

HIGH buzzes, LOW stops. The motor runs from the block's own 3.3 V.

✗

VCC can be 5V, even beside a 3.3 V board

VCC feeds only the block's 3.3 V regulator and never reaches SIGNAL, so 5V is safe on an ESP32, ESP32-S3 or Pico (VBUS on a Pico), and 3V3 works as well. **Keep VCC at 5 V or below:** every volt above 3.3 V is heat in the regulator.

Pins, and which to wire

Parts up, header at the bottom. Wire GND, VCC and SIGNAL; NC is connected to nothing on the board.

Coin motor, Ø10 mm
Rated 3.0 V, for 2.7 to 3.3 V. On its own tape.

SS14 diode
Takes the kick when the motor stops. Nothing to add.

LED
Lit while the MOSFET is on, even if the motor is not.

GND is the square pad
Count from it. The back prints no pin names.

AO3400A MOSFET
SIGNAL is its gate. Your pin gives 0.33 mA, not the motor's.

3.3 V regulator
The only thing VCC reaches. It feeds the motor.

NC
Connected to nothing on the board.

1

2

3

4

Front	GND	VCC	NC	SIGNAL
On / off	GND	VCC	•	SIGNAL
Strength	GND	VCC	•	SIGNAL

SIGNAL only switches

SIGNAL is a MOSFET's gate with 10 kΩ to GND: HIGH buzzes, LOW or unconnected stops. The motor's 54 mA never passes through your pin.

What VCC gives the motor

VCC on 5V

3.3 V on the motor

Measured. The top of its range, about 15,800 rpm, typical. The motor's current comes off USB, not your board's 3.3 V.

VCC on 3V3

3.3 V on the motor

Measured, the same as from 5V: the same speed and shake. The motor's current comes off your board's 3.3 V.

Low levels, from rest

178 of 255 to be sure

A stopped motor is only promised to start from 2.3 V. Below that, write 255 for 50 ms first.

Wire VCC to 5V or 3V3, whichever is handy: the motor gets about 3.3 V from either. Use full on and off for a buzz you can rely on, and PWM for gentler ones, with a kick at 255 to start them.

Motor voltage measured on the motor's + pad of a board, with 5 V and with 3.3 V on VCC. Speed from the motor's typical curve (2.6 to 3.4 V), not measured. Start voltage 2.3 V max, motor 80 mA max running and 120 mA max starting, from the datasheets.

Any strength, with a kick

Arduino IDE · Board: your board · ESP32-S3: USB CDC On Boot: Enabled

```
// SIGNAL: Uno 9. ESP32 4. S3 4. Pico 15.
const int MOTOR_PIN = 9;
const int KICK_MS = 50; // full power first: tune it
bool running = false;

void setLevel(int level) { // 0 to 255
  if (level > 0 && !running) { // from rest: kick
    analogWrite(MOTOR_PIN, 255);
    delay(KICK_MS);
  }
  analogWrite(MOTOR_PIN, level);
  running = level > 0;
}

void setup() {
  Serial.begin(115200);
  pinMode(MOTOR_PIN, OUTPUT);
}

void loop() {
  const int tries[] = {255, 150, 100};
  for (int level : tries) {
    Serial.println(level);
    setLevel(level);
    delay(1500);
    setLevel(0);
    delay(1000);
  }
}
```

255
150
100

Serial Monitor at 115200: the level being tried. The LED dims with the level; the motor shows whether it kept turning.

Why low levels stay still

analogWrite switches SIGNAL on and off hundreds of times a second, and the motor sees the average: 128 of 255 is about 1.66 V. A stopped motor is only promised to start from 2.3 V, about 178 of 255. A turning motor keeps going lower, by an amount the datasheet does not give, so give it 255 for 50 ms and then drop.

When it does not work

Still, and the LED dark

No 3.3 V or no gate: VCC, GND or SIGNAL is off. Count from the square pad.

Buzzes weakly

Loose on a desk, it lifts the board. Fix it to what it should shake.

Your board resets

The start draws up to 120 mA. Take VCC from 5V, not 3V3.

Specifications

Motor	LCM1027A2445F, Ø10 × 2.7 mm	Rated	3.0 V, 2.7 to 3.3 V
Current	80 mA max, 120 mA starting	Motor supply	XC6206P332MR, 3.3 V
Switch	AO3400A, gate 0.7 to 1.4 V	SIGNAL	Gate, 10 kΩ to GND
VCC	5 V or 3.3 V; the motor gets 3.3 V	Protection	SS14 flyback, 100 nF

Wiring

	FRONT	UNO	ESP32	S3	PICO
1	GND	GND	GND	GND	GND
2	VCC	5V	5V	5V	VBUS
3	NC				
4	SIGNAL	D9	GPIO 4	GPIO 4	GP15

Hole 1 is the square pad. VCC is 5V here, or 3V3: the motor gets about 3.3 V from either. The ESP32 boards' 5V and the Pico's VBUS carry 5 V while on USB; on a battery, use 3V3.