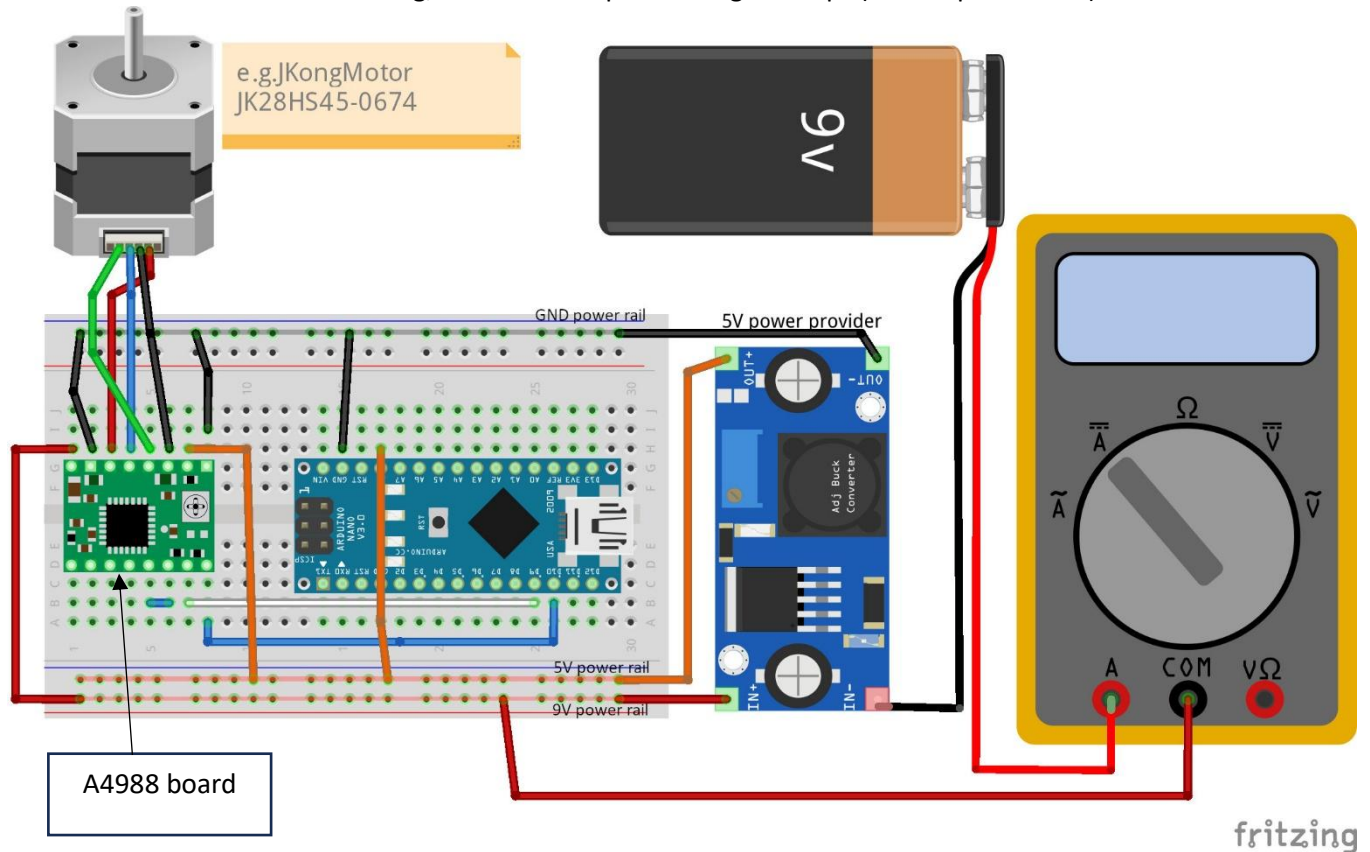


## Mod-8 Individual Component Test

Circuit and sketch for motor testing, and film transport testing at 18 fps (frames per second).



### Notes

Authors often recommend a capacitor between the 9V Power Rail and GND. Not seen here because the “buck” power provider board includes 220 uF capacitors.

“JKongMotor” stepper motors have published voltage ratings in the range 3.7V to 6.7V. In my tests they are giving good reliable high speed operation at a board supply of 7V to 12V leaving the current limiting control (potentiometer) on the A4988 in its default middle position as delivered. The target cameras for modification have 9V battery power. That is a good place to start for motor power supply.

Video of film transport test:

[https://www.youtube.com/watch?v=b9\\_gR04Lhn4](https://www.youtube.com/watch?v=b9_gR04Lhn4)

Arduino Sketch: “Camera\_NEMA\_Test01\_self-advance.ino”

```
/*  
Camera NEMA Test01 John Calder JPC 2026-01-02  
Advance in less than 0.02 sec with pause between  
This code tests the stepper motor on its own,  
The plan for later versions is to trigger the advance from the shutter.
```

```
Mod-8mm Project code applies the Open Source "MIT License"  
copyright (C) 2026 John Calder some rights reserved  
https://choosealicense.com/licenses/mit/
```

```
*/
```

```

const byte STEP = 9; //control pin D9
const byte DIR = 10; //control pin D10
const int SPEED = 18;
const int INTERVAL = 1000/SPEED;
const int STEP_PULSE = 300;
const int STEP_PULSE_A = STEP_PULSE * 3;
const int STEP_PULSE_B = STEP_PULSE_A/2;

//2025-11-24 Step Angle is 1.8 for NEMA - make it a constant
bool ledToggle = false;
// start with an estimate of time to advance the film
int advanceTime = 18;
int holdTime = INTERVAL - advanceTime;
int moveState = 0;
long count = 0;

// the setup function runs once when you press reset or power the board
void setup() {
  // initialize digital pin LED_BUILTIN as an output.
  double stepSpeedAssess = 0;
  pinMode(STEP, OUTPUT); //Step control
  pinMode(DIR, OUTPUT); //Direction configuration
  pinMode(LED_BUILTIN, OUTPUT);
  Serial.begin(9600);
  //initialise ports
  digitalWrite(STEP, LOW);
  digitalWrite(DIR, HIGH);

  //2025-11-24 JPC give a distinctive startup display
  delay(3000);

  digitalWrite(LED_BUILTIN, HIGH);
  delay(3000);
  //digitalWrite(STEP, LOW);
  digitalWrite(LED_BUILTIN, LOW);

  //Arduino Nano clones ATMEGA328 processor with CH340 USB driver
  //are returning 13 as the LED port

```

```

Serial.println("LED_BUILTIN = " + (String)LED_BUILTIN);
Serial.println("STEP_PULSE = " + (String) STEP_PULSE + " microseconds");
Serial.println("INTERVAL = " + (String) INTERVAL + " ms");

//2025-11-19 JPC start with led on for first cycle
digitalWrite(LED_BUILTIN, HIGH);
//ledToggle = false; initialised above
}

void loop() {
//2026-01-02 JPC
// pause between film advance for testing
// planned that shutter sends a pulse that governs this
if(moveState == 0) {
    delay(holdTime);
    moveState = 1;
    return;
}

//advance
//time advance in milliseconds
unsigned long snapshotTime = millis();
//gradual acceleration into advance
digitalWrite(STEP, HIGH);
delayMicroseconds(STEP_PULSE_A);
digitalWrite(STEP, LOW);
delayMicroseconds(STEP_PULSE_A);
digitalWrite(STEP, HIGH);
delayMicroseconds(STEP_PULSE_B);
digitalWrite(STEP, LOW);
delayMicroseconds(STEP_PULSE_B);

//main advance
for(int i = 3; i <= 25; i++){
    digitalWrite(STEP, HIGH);
    delayMicroseconds(STEP_PULSE);
    digitalWrite(STEP, LOW);
    delayMicroseconds(STEP_PULSE);
}

```

```

//gradual deceleration out of advance
digitalWrite(STEP, HIGH);
delayMicroseconds(STEP_PULSE_B);
digitalWrite(STEP, LOW);
delayMicroseconds(STEP_PULSE_B);
digitalWrite(STEP, HIGH);
delayMicroseconds(STEP_PULSE_A);
digitalWrite(STEP, LOW);
delayMicroseconds(STEP_PULSE_A);

moveState = 0;
count += 1;
//Provide monitoring data via Serial every 10 seconds,
//visible on a connected laptop
//Also flashes the led 10 seconds on, 10 seconds off as
//a quick check that this code is running
advanceTime = (int)(millis() - snapshotTime);

if(count%(SPEED * 10) == 0){
    Serial.println((String)count + " frames; advanceTime = " +
(String)advanceTime + " ms");
    if(ledToggle) {
        digitalWrite(LED_BUILTIN, HIGH);
        ledToggle = false;
    } else {
        digitalWrite(LED_BUILTIN, LOW);
        ledToggle = true;
    }
}
}
}

```