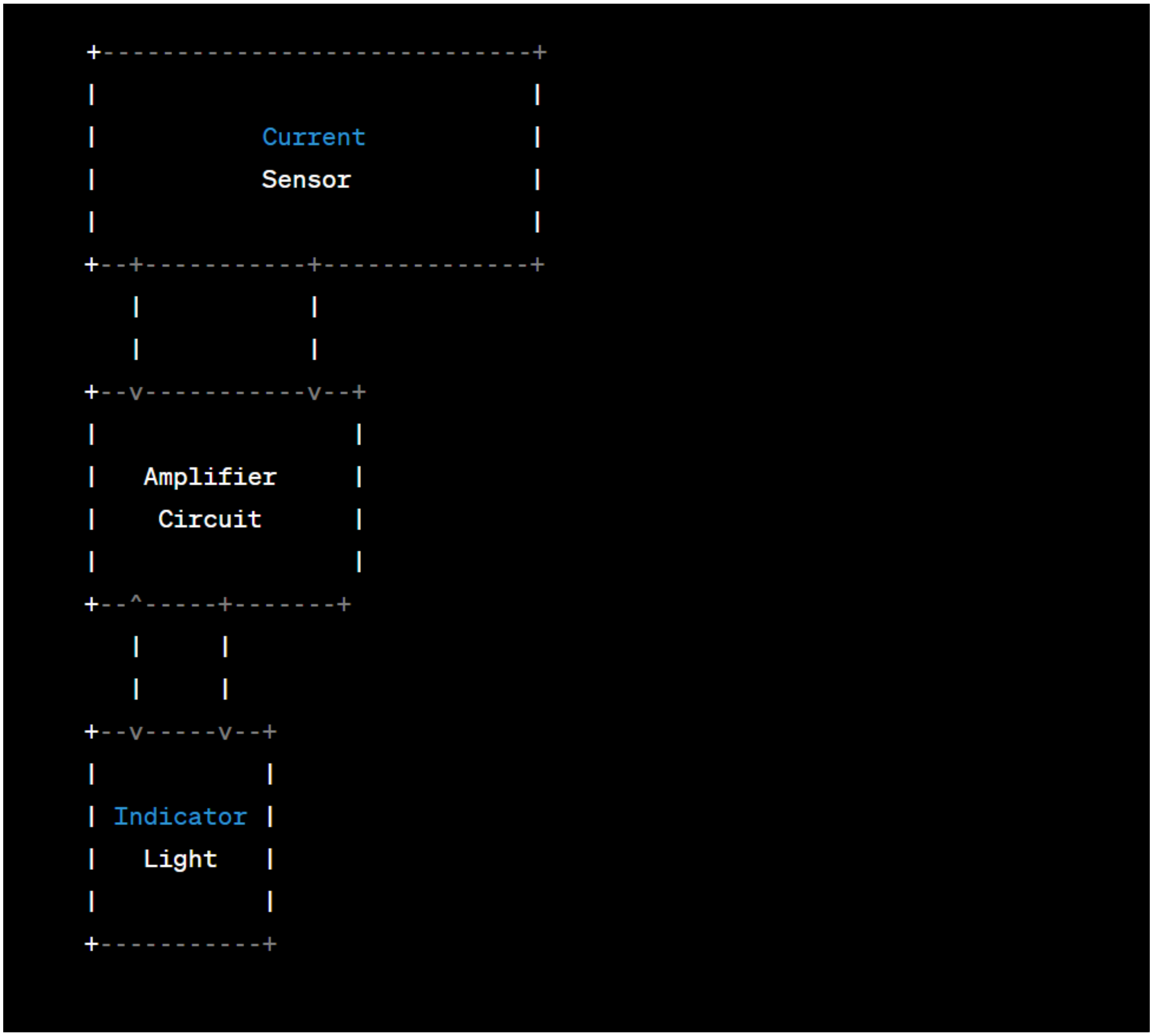
Electrical Analysis

To build a circuit that indicates when a motor's amperage rises, you can use a current sensor, in conjunction with an indicator light, such as an LED.

Current Sensor: Connect the current sensor, such as a current transformer, in series with the motor's power supply. The current sensor should provide an output that is proportional to the motor's amperage.

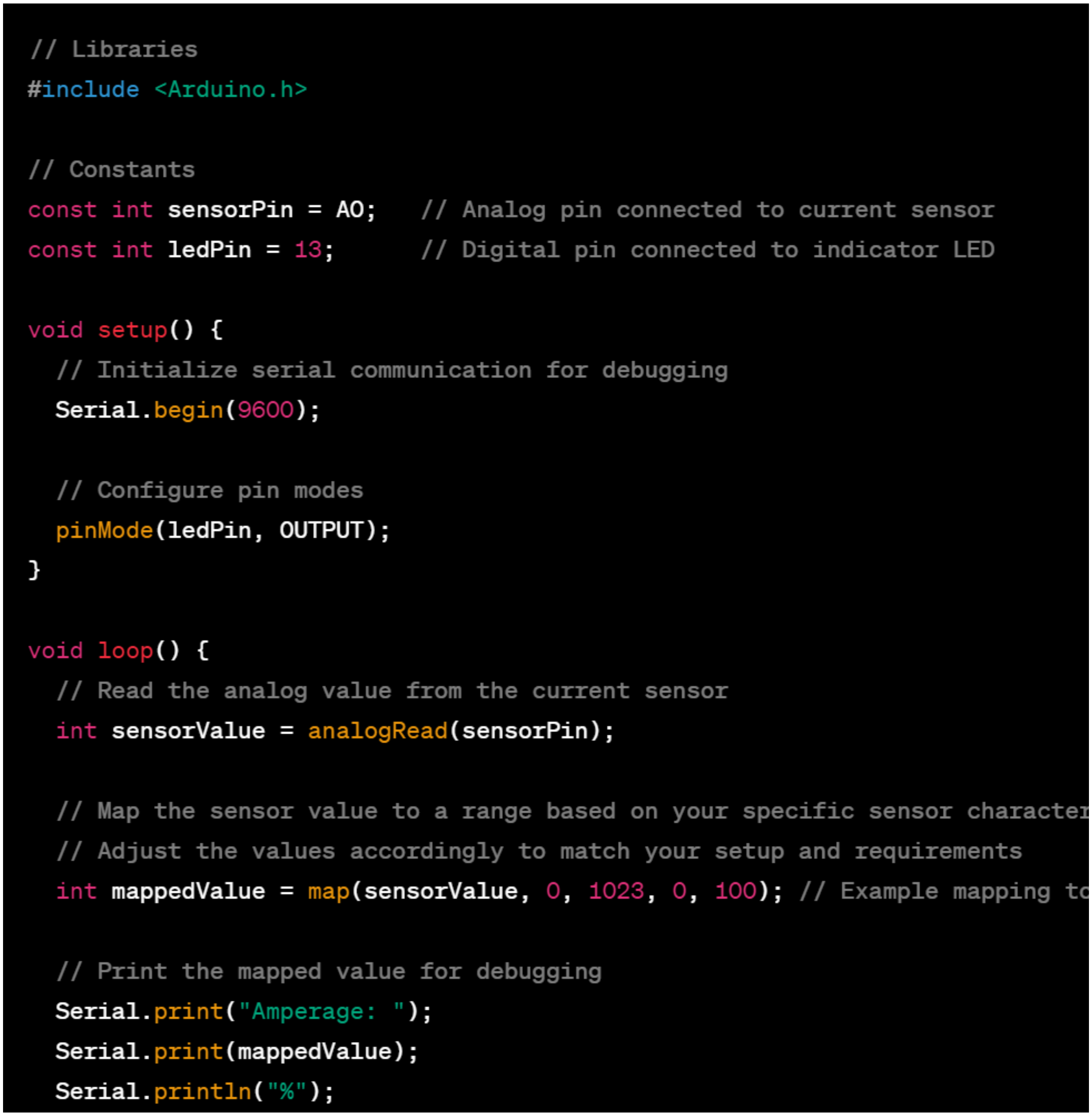
Amplifier Circuit: We will connect the output of the current sensor to an amplifier circuit. The amplifier will boost the sensor's weak output signal to a level suitable for driving the indicator light.

Indicator Light: We will connect an indicator light, such as an LED, in parallel with the output of the amplifier circuit.

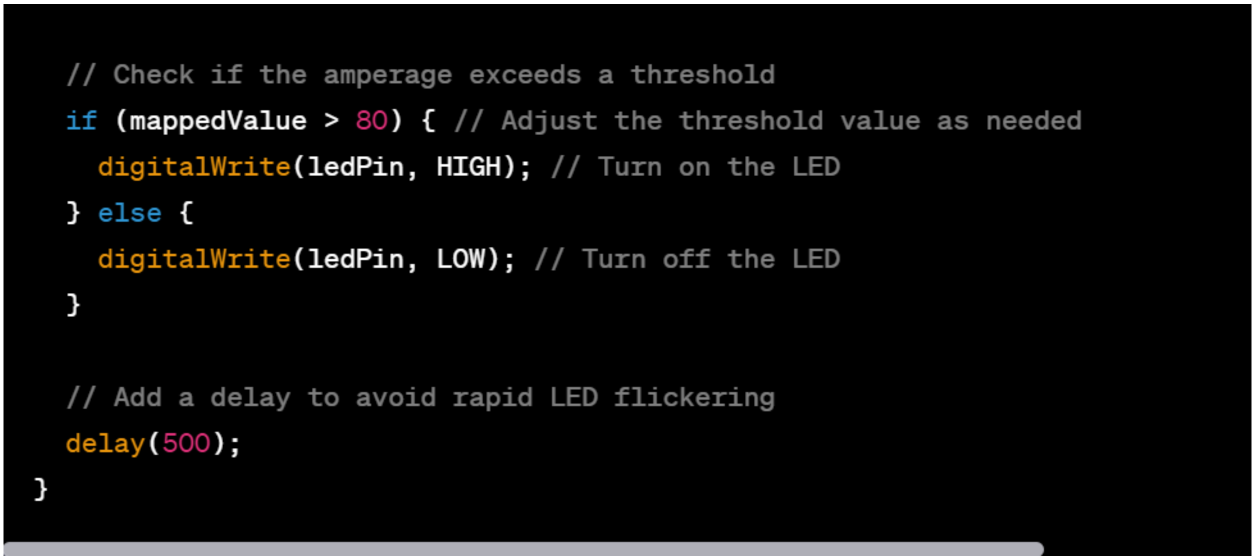


```
+-----------------------------+
|                             |
|           Current           |
|           Sensor            |
|                             |
+--+-----------+--------------+
   |           |
   |           |
+--v-----------v--+
|                 |
|   Amplifier     |
|    Circuit      |
|                 |
+--^-----+-------+
   |     |
   |     |
+--v-----v--+
|           |
| Indicator |
|   Light   |
|           |
+-----------+
```

Below is a sample Arduino Code:



```
// Libraries
#include <Arduino.h>

// Constants
const int sensorPin = A0;   // Analog pin connected to current sensor
const int ledPin = 13;      // Digital pin connected to indicator LED

void setup() {
  // Initialize serial communication for debugging
  Serial.begin(9600);

  // Configure pin modes
  pinMode(ledPin, OUTPUT);
}

void loop() {
  // Read the analog value from the current sensor
  int sensorValue = analogRead(sensorPin);

  // Map the sensor value to a range based on your specific sensor character
  // Adjust the values accordingly to match your setup and requirements
  int mappedValue = map(sensorValue, 0, 1023, 0, 100); // Example mapping to

  // Print the mapped value for debugging
  Serial.print("Amperage: ");
  Serial.print(mappedValue);
  Serial.println("%");
```



```
  // Check if the amperage exceeds a threshold
  if (mappedValue > 80) { // Adjust the threshold value as needed
    digitalWrite(ledPin, HIGH); // Turn on the LED
  } else {
    digitalWrite(ledPin, LOW); // Turn off the LED
  }

  // Add a delay to avoid rapid LED flickering
  delay(500);
}
```

Using an Arduino board with an analog pin (A0) connected to the output of the current sensor and a digital pin (13) connected to the indicator LED. Adjust the pin numbers and mapping values according to our specific setup.

The code reads the analog value from the current sensor and maps it to a range (0-100 in this example). It then checks if the mapped value exceeds a threshold (80% in this example) and turns on the LED if it does.