

Wächter für Laser-Wasserkühlung

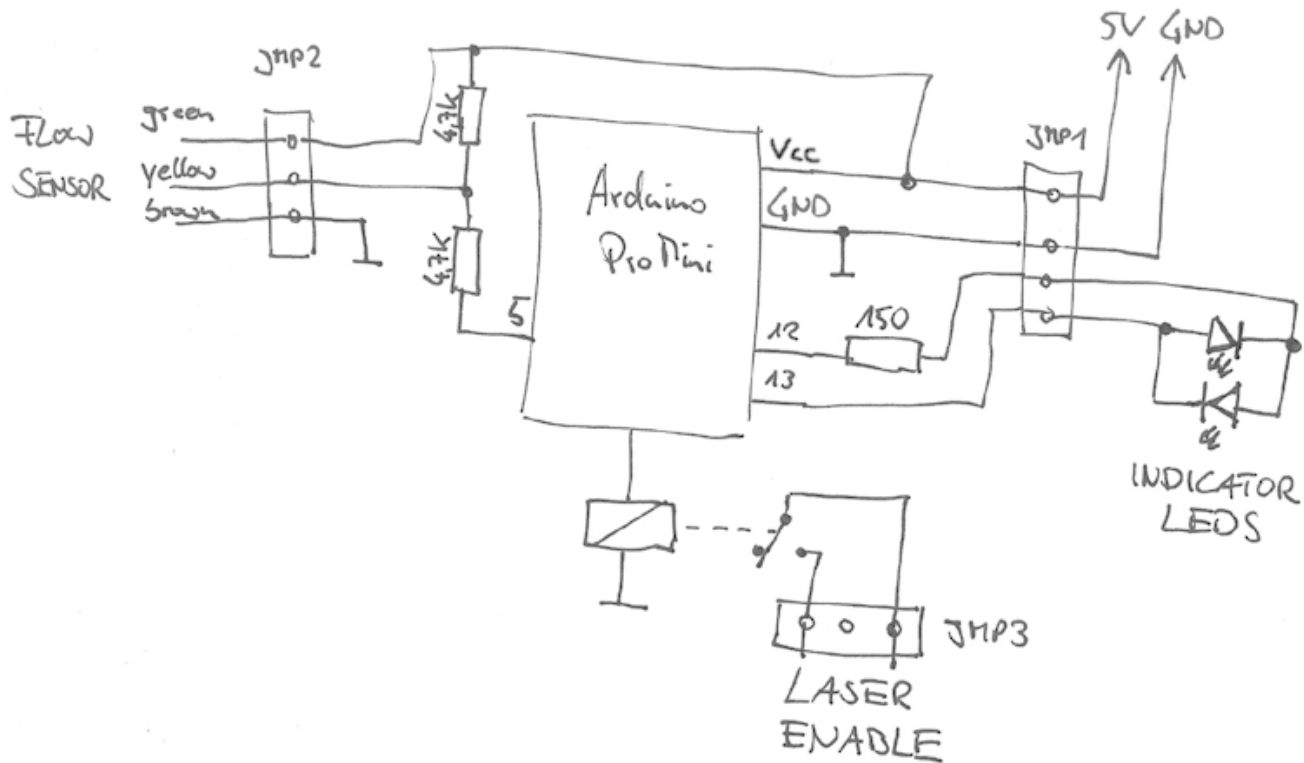
Für unseren LAOS-Laser wäre eine Überwachung der Wasserkühlung gut. Fällt die Wasserkühlung aus, sollte der Laser sofort ausgeschaltet werden.

Zutaten:

- Flow-Sensor aus Kaffemaschine (SAECO Cafe Prima / SUP 018C)
- Arduino Mini Pro 5V/16MHz
- Relais

Der Flowsensor wird in den Rücklauf der Wasserkühlung eingebaut. Erkennt der Arduino, dass kein Wasser mehr fließt, schaltet das Relais (welches in Reihe zum Magnetschalter im Deckel des Lasers geschaltet ist) den Laser aus.

Schaltplan



Code

WaterCoolerGuard.ino

```
/*
  Water cooler guard
  */

int flowSensorPin = 5;
int led1 = 12;
int led2 = 13;
int relayPin = 10;

int minTickCount = 10; // alarm if less ticks per 500 millisecs

int blinkState = 0;

void setup()
{
```

```

pinMode(flowSensorPin, INPUT);
pinMode(led1, OUTPUT);
pinMode(led2, OUTPUT);
pinMode(relayPin, OUTPUT);

// start serial port at 9600 bps:
Serial.begin(9600);
Serial.println("Water cooler guard - INIT");
digitalWrite(relayPin, LOW);

// initial blinking
digitalWrite(led1, HIGH);
digitalWrite(led2, LOW);
delay(200);
digitalWrite(led1, LOW);
digitalWrite(led2, HIGH);
delay(200);
digitalWrite(led1, HIGH);
digitalWrite(led2, LOW);
delay(200);
digitalWrite(led1, LOW);
digitalWrite(led2, HIGH);
delay(200);
digitalWrite(led1, LOW);
digitalWrite(led2, LOW);
}

void loop()
{
  int ticks = getTickCount(500);

  Serial.print("Tick count: ");
  Serial.print(ticks);
  if (ticks >= minTickCount)
  {
    // all good, show green light
    digitalWrite(relayPin, HIGH);
    digitalWrite(led1, LOW);
    digitalWrite(led2, HIGH);
    blinkState = 0;
  }
  else
  {
    // alert! red light
    Serial.print(" Low flow alert!");
    digitalWrite(relayPin, LOW);
    digitalWrite(led1, blinkState ? LOW : HIGH);
    digitalWrite(led2, LOW);

    blinkState = !blinkState;
  }
  Serial.println();
}

// count the pulses from the flow sensor
// for the given time in millisecs
int getTickCount(int milliSecs)
{
  unsigned long startTime = millis();
  unsigned long endTime = startTime+milliSecs;

  while (endTime < startTime)
  {
    // timer overrun (will probably never happen, but just in case)
    // simply wait a bit and retry
    delay(50);
    startTime = millis();
    endTime = startTime+milliSecs;
  }

  int count = 0;

```

```
while (millis() < endTime)
{
  // wait until low
  while(HIGH == digitalRead(flowSensorPin))
  {
    if (millis() >= endTime)
    {
      return count; // reached end time, jump out
    }
  }

  // count
  count++;

  // wait until high again
  while(LOW == digitalRead(flowSensorPin))
  {
    if (millis() >= endTime)
    {
      return count; // reached end time, jump out
    }
  }
}
return count;
}
```