

# Lichtmaschinenregler mit Benzinpumpenabschaltautomatik im Saporoshez

## 1. Der Lichtmaschinenregler

Die übliche Funktion der Ladekontrollleuchte im Saporoshez ist wie folgt: Die rote Kontrollleuchte stellt eine Verbindung zwischen dem Zündungs-Plus und dem gleichgerichteten Ausgang der Lichtmaschine her. Wenn die Lichtmaschine keinen Strom liefert, fließt Strom zwischen dem Zündungs-Plus und der Lichtmaschine durch die rote Kontrollleuchte. Die Lichtmaschine verhält sich in diesem Fall wie die Fahrzeugmasse ( Minus ). Wenn die Lichtmaschine im Betrieb Strom liefert, gibt es keine Spannungsdifferenz mehr zwischen Zündung-Plus und der Lichtmaschine. Ohne Spannungsdifferenz leuchtet keine Lampe, die rote Kontrollleuchte ist aus.

Die Kontrollleuchte wird also mit Zündungs-Plus versorgt. Verbindet man das Kabel der Kontrollleuchte im Motorraum mit Masse, leuchtet sie bei eingeschalteter Zündung.

Im Lichtmaschinenregler mit Benzinpumpenabschaltautomatik im Saporoshez wird die Ladekontrollleuchte über den NPN-Transistor BD139 gegen Masse geschaltet. Wenn die Lichtmaschine keinen Strom liefert, wird der NPN-Transistor BD139 gegen Masse leitend geschaltet und die die Ladekontrollleuchte leuchtet. Wenn die elektrische Benzinpumpe abgeschaltet wird, wird die Ladekontrollleuchte dreimal hintereinander gegen Masse geschaltet, sie leuchtet dann als Warnhinweis dreimal hintereinander. Geschaltet wird der NPN-Transistor BD139 über den PIN11 des Arduino Nano.

Der konventionelle Lichtmaschinenregler PP310-Б im Saporoshez funktioniert im Prinzip wie ein Relais, das bei einer Bordspannung von unter 13,8 V bis 14,8 V die Erregerwicklung der Lichtmaschine mit Strom versorgt. Oberhalb einer Bordspannung von 13,8 V bis 14,8 V wird die Erregerwicklung stromlos geschaltet. Da im normalen Betrieb die Bordspannung häufig um diesen Wert schwankt, schaltet dieses Relais häufig an und aus.

Im Lichtmaschinenregler mit Benzinpumpenabschaltautomatik wird die Funktion des Relais elektronisch ersetzt. Der Mosfet FQP47P06 versorgt im Prinzip wie ein PNP Transistor die Erregerwicklung der Lichtmaschine mit Strom. Hier wird im Gegensatz zur Ladekontrollleuchte „Plus“ geschaltet. Da die Digitalpins des Arduino Nano maximal 5V mit 20ma liefern, ist eine so genannte High-Side-Treiber-Schaltung notwendig. Der Mosfet FQP47P06 wird über den Transistor 2N3904 angesteuert, der Transistor 2N3904 ist wiederum mit dem PIN9 des Arduino Nano verbunden.

Wie viele andere Mikrocontroller auch kann der Arduino Nano [PWM Signale](#) abgeben. Obwohl ein Mikrocontroller eigentlich nur seine Ausgänge ( seine PINs ) an- oder ausschalten kann, kann durch schnelles Ein- und Ausschalten mit unterschiedlichen langen Zeiträumen z.B. eine Leuchte gedimmt werden. Auch der Strom der über den Arduino Nano PIN9 angesteuerten Erregerwicklung kann so angepasst werden.

Über einen Spannungsteiler mit zwei Widerständen, 1000  $\Omega$  und 330  $\Omega$  misst der Arduino Nano am Analogpin A1 die Bordspannung. Liegt die Bordspannung unter 13,51 Volt, wird die PWM Frequenz am PIN9 erhöht, die Erregerwicklung erhält praktisch mehr Strom. Liegt die Bordspannung dann über 13,51 Volt, wird die PWM am PIN9 verringert, die Erregerwicklung erhält praktisch weniger Strom.

## 2. Die Benzinpumpenabschaltautomatik

Elektrische Benzinpumpen haben im Gegensatz zu mechanischen, vom Motor angetriebenen Benzinpumpen den Vorteil, dass sie unabhängig von der Motordrehzahl stets die gleiche Menge Benzin fördern. Vor dem Starten des Motors kann damit bereits die Schwimmerkammer des Vergasers gefüllt werden. Auch im Leerlauf ist immer genügend Benzin im Vergaser vorhanden.

Der Nachteil der elektrischen Benzinpumpe ist jedoch, dass auch bei einem plötzlichem Stillstand des Motors das Benzin weiter gefördert wird.

Die Benzinpumpenabschaltautomatik sorgt nun dafür, dass – mit einer Ausnahme – die elektrische Benzinpumpe nur Strom erhält, wenn der Motor läuft. Die elektrische Benzinpumpe hätte man über ein Relais ansteuern können. Statt einem Relais wird genau wie bei der Ansteuerung der Erregerwicklung ebenfalls ein High-Side-Mosfet FQP47P06 verwendet. Dieser zweite Mosfet FQP47P06 wird vom Arduino-Pin 10 angesteuert.

Beim Saporoshez wird mit Kabeln aus der Lichtmaschine nicht nur das gleichgerichtete PLUS zum Laden der Batterie geführt, sondern auch für die Funktion des Anlasserverriegelungsrelais auch zwei Phasen der Drehstromlichtmaschine. An diesen beiden Phasen liegt nur Spannung an, wenn sich die Lichtmaschine dreht. Die Lichtmaschine dreht sich nur, wenn der Motor nicht stillsteht.

Hierdurch kann für den Lichtmaschinenregler mit Benzinpumpenabschaltautomatik gemessen werden, ob sich die Lichtmaschine und damit der Motor dreht. Die beiden Phasen der Lichtmaschine werden über 4 Dioden gleichgerichtet, die Spannung wird über den üblichen Spannungsteiler mit zwei Widerständen 1000  $\Omega$  und 330  $\Omega$  verringert und anschließend über den Analogpin 3 des Arduino Nano gemessen.

Der im Arduin Nano realisierte Lichtmaschinenregler mit Benzinpumpenabschaltautomatik wird mit Zündungs-Plus mit Spannung versorgt, wobei ein Step-Down Wandler zum Herausfiltern von Spannungsspitzen zuerst die Spannung auf etwa 6,3 Volt senkt. Diese 6,3 Volt gehen in den Eingang VIN des Arduino Nano. Jedes Mal, wenn im Saporoshez die Zündung am Zündschloss ein- und ausgeschaltet wird, startet der Lichtmaschinenregler mit Benzinpumpenabschaltautomatik neu.

Zur Messung, ob der Anlasser betätigt wird, ist das Kabel vom Anlasserrelais mit dem üblichen Spannungsteiler mit zwei Widerständen 1000  $\Omega$  und 330  $\Omega$  mit dem Analogpin 3 des Arduino Nano verbunden. Hierdurch kann im Arduino Nano gemessen werden, ob der Anlasser betätigt wird.

Nach jedem neuen Einschalten der Zündung und damit jedem neuen Starten des Lichtmaschinenreglers mit Benzinpumpenabschaltautomatik wird die Benzinpumpe für 15 Sekunden mit Strom versorgt. Und dies unabhängig davon, ob der Motor läuft.

Ebenfalls wird nach jedem ( auch noch so kurzem ) Betätigen des Anlassers die Benzinpumpe für 15 Sekunden mit Strom versorgt.

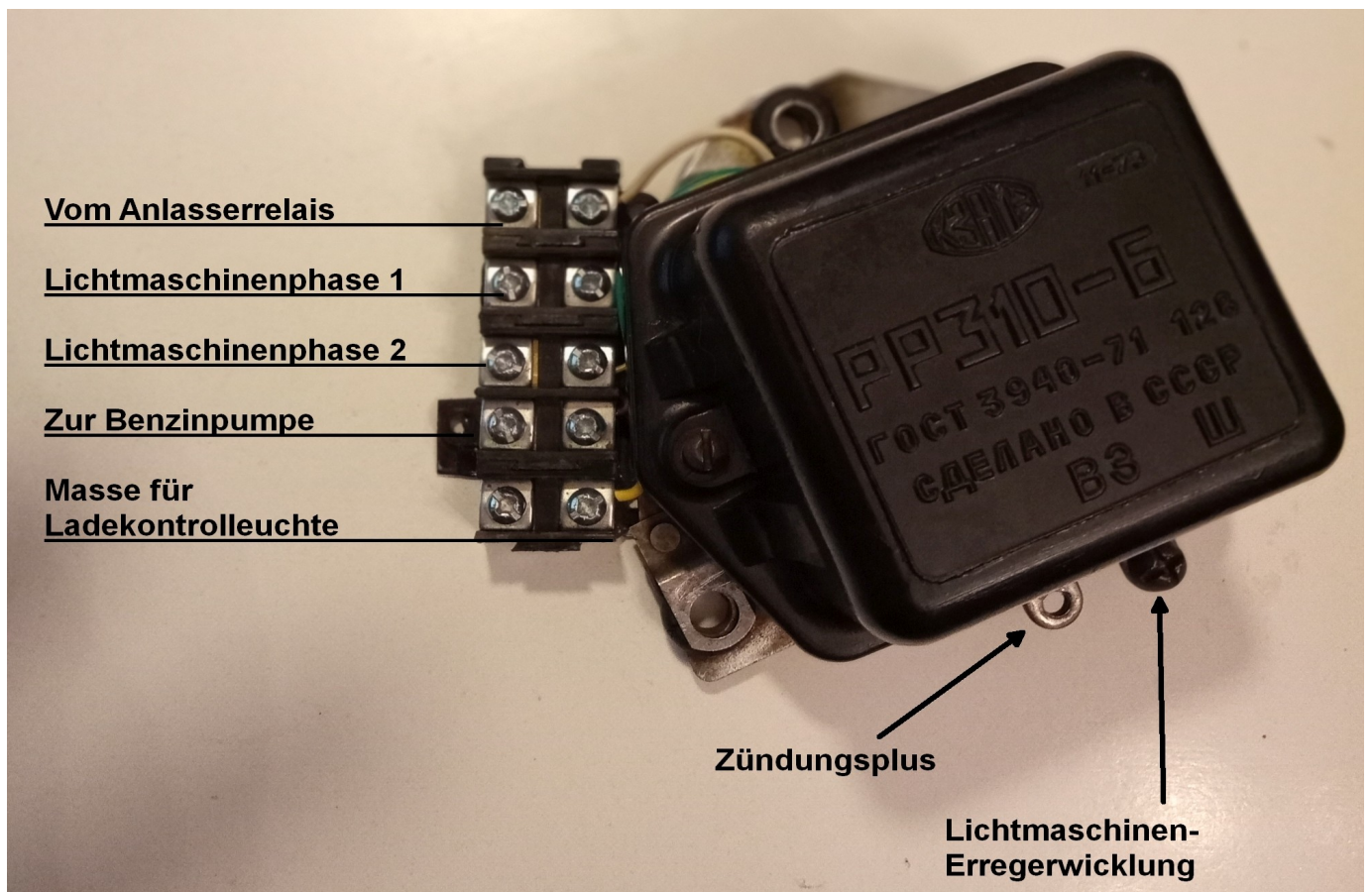
Nach den jeweils 15 Sekunden zu Anfang wird die Benzinpumpe stromlos geschaltet, wenn der Motor nicht läuft. Als Warnung, dass die Benzinpumpe nun stromlos ist, blinkt die Ladekontrollleuchte 3 Mal hintereinander.

Wenn man den Motor „abwürgt“, d.h. er bei eingeschalteter Zündung stehen bleibt, wird ebenfalls die Benzinpumpe stromlos geschaltet und es blinkt die Ladekontrollleuchte 3 Mal hintereinander. Für den Fall, dass der Saporoshez anschließend angeschoben wird, läuft der Motor ja wieder und die Benzinpumpe wird mit Strom versorgt.

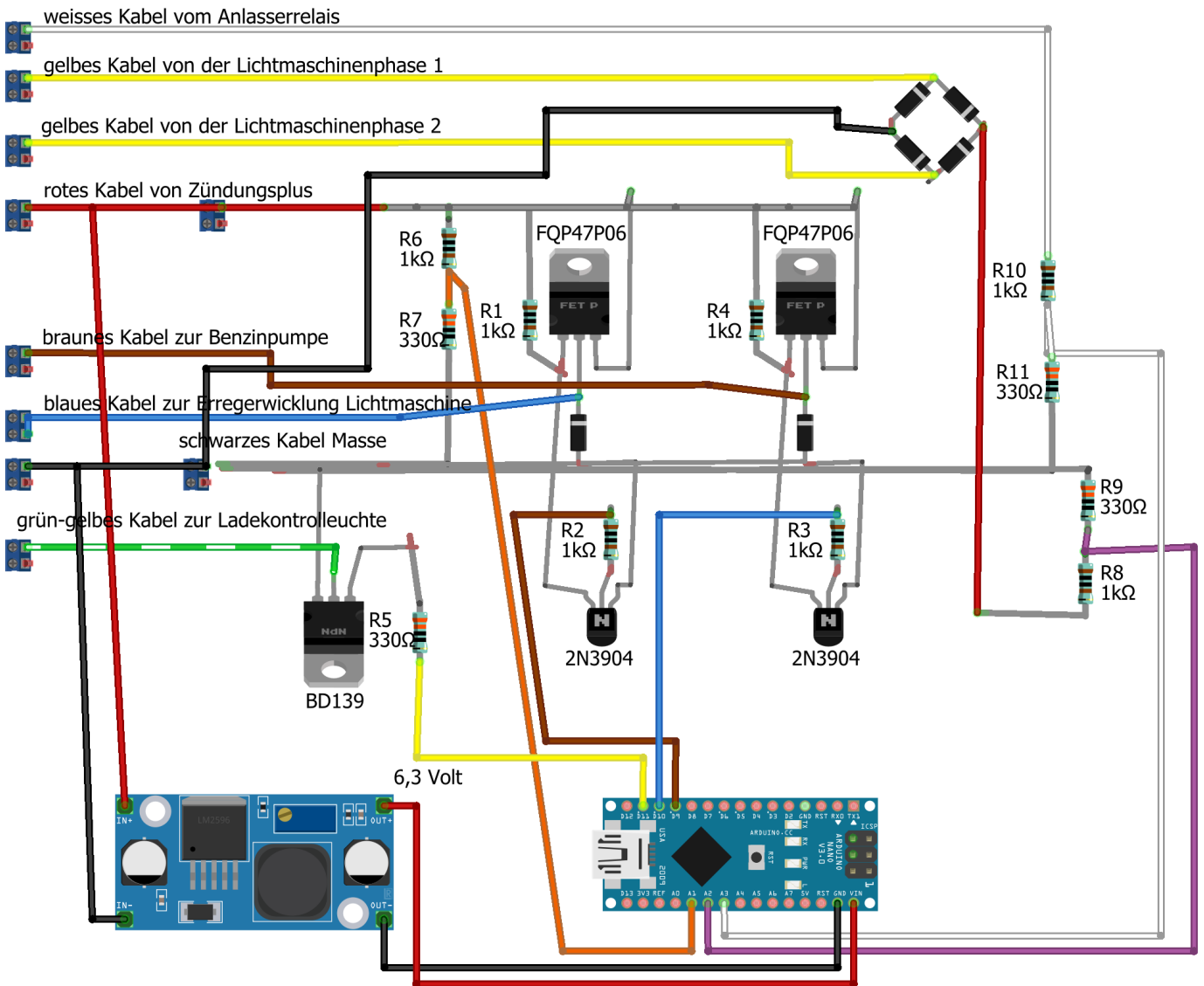
### 3. Notschaltung bei Defekt des Lichtmaschinenreglers mit Benzinpumpenabschaltautomatik

Die Lichtmaschine im Saporoshez liefert einen maximalen Strom von 30 A. Bei eingeschalteter Zündung, Abblendlicht, Scheibenwischer und Heizung reicht dies gerade einmal aus, um knapp die Bordspannung von 12 V zu halten. Bei einem Defekt des Lichtmaschinenreglers kann gefahrlos eine längere Strecke mit vollem Strom zur Erregerwicklung gefahren werden. Hierbei sollte dann auf jeden Fall mit Licht gefahren werden, um etwas Strom zu verbrauchen. Als Notschaltung wird dafür das Kabel der Lichtmaschinen-Erregerwicklung direkt an die links daneben liegende Klemme für Zündungs-Plus angeschlossen.

Die elektrische Benzinpumpe kann bei einem Defekt der Benzinpumpenabschaltautomatik einfach an die Klemme für Zündungs-Plus angeschlossen werden. Dann ist die Benzinpumpe an, wenn die Zündung an ist.

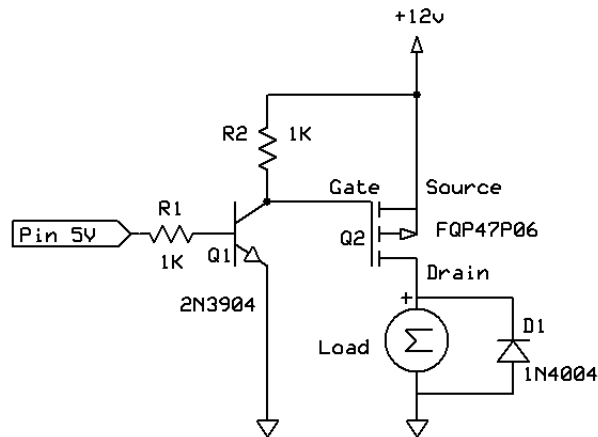


# Lichtmaschinenregler mit Benzinpumpenabschaltautomatik



fritzing

High-side driver (source current)  
 Note: motor is on when pin is high.



R1 limits current  
 R2 pulls Q2 gate high when Q1 is off  
 D1 is flyback protection  
 Q1 is NPN BJT low-level transistor  
 Q2 is P-channel MOSFET

```

// lichtmaschinenregler_benzinpumpensteuerung_im_sapo
// seit dem 11.05.2023 im Saporoshez eingebaut
//*****
// Dies ist die Version mit der Platine
//*****
// 1 // 2 // 3 // 4 // 5 //
//*****//
// Lichtmaschinenregler //
//*****//
const int analogPin1 = A1;
// oranges Kabel vom Spannungsteiler Zündungsplus (PLUS)
const int analogPin2 = A2;
// dünnes lila Kabel vom Spannungsteiler Lichtmaschinenphase, dickes lila/rotes Kabel
// dazwischen noch eine GleichrichterbrückeS
// Schraubkontakt 3 gelbes Kabel von Lichtmaschinenphase 1
// Schraubkontakt 4 gelbes Kabel von Lichtmaschinenphase 2
const int analogPin3 = A3; // dünnes weißes Kabel vom Spannungsteiler Anlasserrelais
// Schraubkontakt 5 dickes weißes Kabel vom Spannungsteiler Anlasserrelais
const int PWMPin1 = 9; // dünnes braunes Kabel zum P-Mosfet 1
// P-Mosfet 1: dickes blaues Kabel zum Anschluss Erregerwicklung
const int DigitalPin1 = 10; // dünnes blaues Kabel zum P-Mosfet 2
// Schraubkontakt 2 P-Mosfet 2: dickes braunes Kabel zum Anschluss Benzinpumpe
const int DigitalPin2 = 11; // dünnes gelbes Kabel zum NPN Leistungs-Transistor
// Schraubkontakt 1 NPN Leistungs-Transistor: grün-gelb um Ladekontrolle auf Masse zu schalten

int val1 = 0; // speichert den gelesenen Wert 1 Zündungsplus (PLUS)
int val2 = 0; // speichert den gelesenen Wert 2 Lichtmaschinenphase
int val3 = 0; // speichert den gelesenen Wert 3 Anlasserrelais
int pwm = 0; // pwm zum Senden
int sollvolt = 691; // 691 entsprechen vermutlich 13,51 Volt
// sollvolt muss bei einem Nachbau durch Probieren ermittelt werden
long pumpendauer = 15000; // erste Einschaltdauer der Benzinpumpe = 15 Sekunden
unsigned long zuendungdauer = 0; // Dauer der Zündung, d.h. seit dem Einschalten Arduino
unsigned long zuendungdauer2 = 0; // Dauer der Zündung seit Anlasser, d.h. seit dem nächsten Startversuch
long startTime = 0; // millis-Wert beim Start der Zündung, d.h. beim Einschalten Arduino
long startTime2 = 0; // millis-Wert beim Betätigen des Anlassers
int benzinpumpenvorlauf = LOW; // Vorlauf der Benzinpumpe ohne Motorlauf
int b1 = 1; // Wert 1 für Berechnung Benzinpumpe, erster Wert für Zeitablauf
int b2 = 0; // Wert 2 für Berechnung Benzinpumpe, Wert für erneuten Zeitablauf
int b3 = 0; // Wert 3 für Berechnung Benzinpumpe, Wert für Ermittlung über Lichtmaschinenphase val2

const unsigned long ON_TIME = 200; // für dreimalblinken
const unsigned long OFF_TIME = 200; // für dreimalblinken
const unsigned long PAUSE_TIME = 600; // für dreimalblinken
unsigned long previousMillis = 0; // für dreimalblinken
int ladekontrolleuchte = LOW; // für dreimalblinken
unsigned long blinkCount = 0; // für dreimalblinken

void setup() {
    startTime = millis();

    pinMode(PWMPin1, OUTPUT);
    pinMode(DigitalPin1, OUTPUT);
    pinMode(DigitalPin2, OUTPUT);
}
void auslesen()
{
    val1 = analogRead(analogPin1);// Lesen des Analogpins 1 Zündungsplus (PLUS)
    val2 = analogRead(analogPin2);// Lesen des Analogpins 2 Lichtmaschinenphasen
    val3 = analogRead(analogPin3);// Lesen des Analogpins 3 Anlasserrelais
}
void pwmberechnen()
{
    if ( val1 < sollvolt ) ++pwm;
    if ( pwm > 255 ) pwm = 255;
    if ( val1 > sollvolt ) pwm = pwm - 1;
    if ( pwm < 1 ) pwm = 0;
}

```

```

void pwmsenden()
{
analogWrite(PWMPin1,pwm); // Senden pwm
}
void dreimalblinken()
{
  unsigned long currentMillis = millis();

  if (ladekontrolleuchte == LOW && blinkCount == 0 && currentMillis - previousMillis >= PAUSE_TIME) {
    ladekontrolleuchte = HIGH;
    previousMillis = currentMillis;
    blinkCount++;
  }
  else if (ladekontrolleuchte == HIGH && currentMillis - previousMillis >= ON_TIME) {
    ladekontrolleuchte = LOW;
    previousMillis = currentMillis;
    blinkCount++;
  }
  else if (ladekontrolleuchte == LOW && blinkCount < 3 && currentMillis - previousMillis >= OFF_TIME) {
    ladekontrolleuchte = HIGH;
    previousMillis = currentMillis;
  }
  else if (ladekontrolleuchte == HIGH && blinkCount < 3 && currentMillis - previousMillis >= ON_TIME) {
    ladekontrolleuchte = LOW;
    previousMillis = currentMillis;
  }
  else if (ladekontrolleuchte == LOW && blinkCount == 3 && currentMillis - previousMillis >= PAUSE_TIME) {
    ladekontrolleuchte = HIGH;
    previousMillis = currentMillis;
    blinkCount = 0;
  }
}

void benzinpumpeberechnen()
{
  zuendungdauer = millis() - startTime;
  if (zuendungdauer > pumpendauer) // wenn die Zündung länger als der Grenzwert läuft
  {
    b1 = 0;          // dann wird Wert 1 auf 0 gesetzt
  }
  if ( val3 > 100 ) // Wenn Spannung am Anlasserrelais anliegt
  {
    b2 = 1; // bei diesem Neustart zu Beginn Wert 2 auf 8
    startTime2 = millis();
  }
  zuendungdauer2 = millis() - startTime2;
  if (zuendungdauer2 > pumpendauer) // wenn die Zündung länger als der Grenzwert läuft
  {
    b2 = 0;          // dann wird Wert 2 auf 0 gesetzt
  }

  if ( val2 > 50) // Wenn Spannung an der Lichtmaschinenphase anliegt
  {
    b3 = 1;          // dann wird Wert 3 auf 1 gesetzt
  }
  if ( val2 < 51 ) // Wenn keine Spannung an der Lichtmaschinenphase anliegt
  {
    b3 = 0;          // dann wird Wert 3 auf 0 gesetzt
  }
  if ( ( b1 + b2 + b3 ) > 0 )
  {
    benzinpumpenvorlauf = HIGH;
    previousMillis = 0;
    blinkCount = 0;
  }
}

```

```
if ( ( b1 + b2 + b3 ) < 1 )
{
  benzinpumpenvorlauf = LOW;
  dreimalblinken();
}
digitalWrite(DigitalPin1, benzinpumpenvorlauf);

}
void ladeleuchteberechnen()
{
  if ( ( b1 > 0 || b2 > 0 ) && b3 == 0 )
  {
    ladekontrolleuchte = HIGH;
    unsigned long previousMillis = 0;
  }
  if ( b3 > 0 )
  {
    ladekontrolleuchte = LOW;
  }

  if ( benzinpumpenvorlauf == LOW )
  {
    dreimalblinken();
  }
  digitalWrite(DigitalPin2, ladekontrolleuchte);
}
void loop()
{
  auslesen();
  pwmberechnen();
  pwmsenden();
  benzinpumpeberechnen();
  ladeleuchteberechnen();

  digitalWrite(DigitalPin2, ladekontrolleuchte);
}
```

