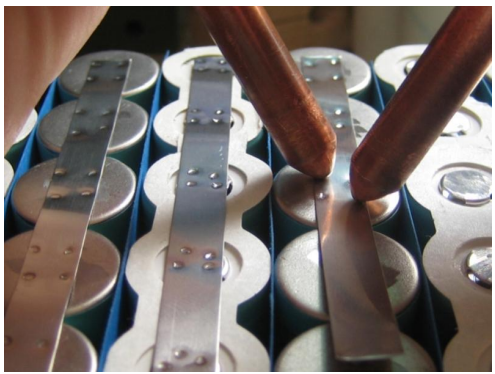
Punktschweißen?

- Hoher Strom -> Materialerwärmung -> Verschmelzen

- Kurzer Impuls
  -> Erwärmung nur an Schweißstelle, Akkuchemie ungeschädigt

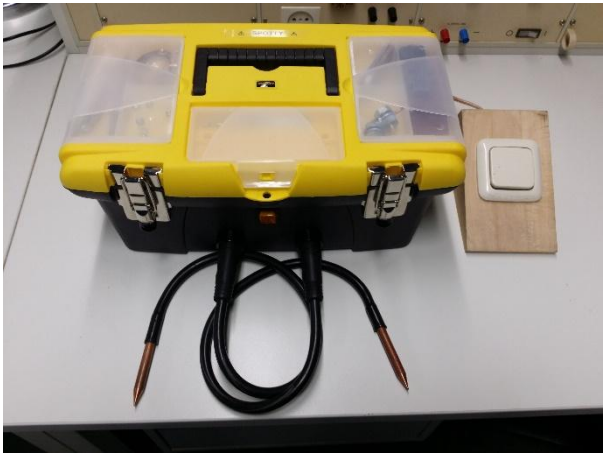
Hoher Strom durch „Kurzschluss“ über Verbinder und Akkugehäuse

Kondensator vs. Transformator
+ Bauteilverfügbarkeit
+ keine Wartezeit (Kondensatorladung)



Funktionen

- Bauart bedingter Schweißstrom
- Intensität einstellbar über Dauer Schweißimpuls
- Steuerung über Arduino mit Touchdisplay
- Reinigung und Pausenzeit
- Auslösung über Fußschalter
- Austauschbare Elektroden



Baugruppen

- Transformator
  - Stellt den hohen Schweißstrom zur Verfügung
- Platine
  - Anbindung aller Bauteile
  - Versorgung von Transformator und Lüfter
  - Spannungsversorung Arduino und Lüfter
- Arduino
  - Auswertung Temperatursensor und Schalter
  - Ansteuerung und Auswertung Display
  - Ansteuerung von Transformator und Lüfter
  - Verwaltung Programmablauf
- Touchdisplay
  - Einstellung Intensität Schweißvorgang
  - Anzeige Temperatur

Sicherheit

- Hardware
  - Echter Netzschalter
  - Überstromsicherung
    (230V für komplettes Gerät, 12V für Arduino und Lüfter)
  - Temperatursicherung (in Sekundärwicklung vergossen)
  - Temperatursensor
  - Lüfter
- Software
  - Watchdog
    (überwacht korrekten Programmablauf)
  - Temperaturgesteuerter Lüfter

Gehäuse

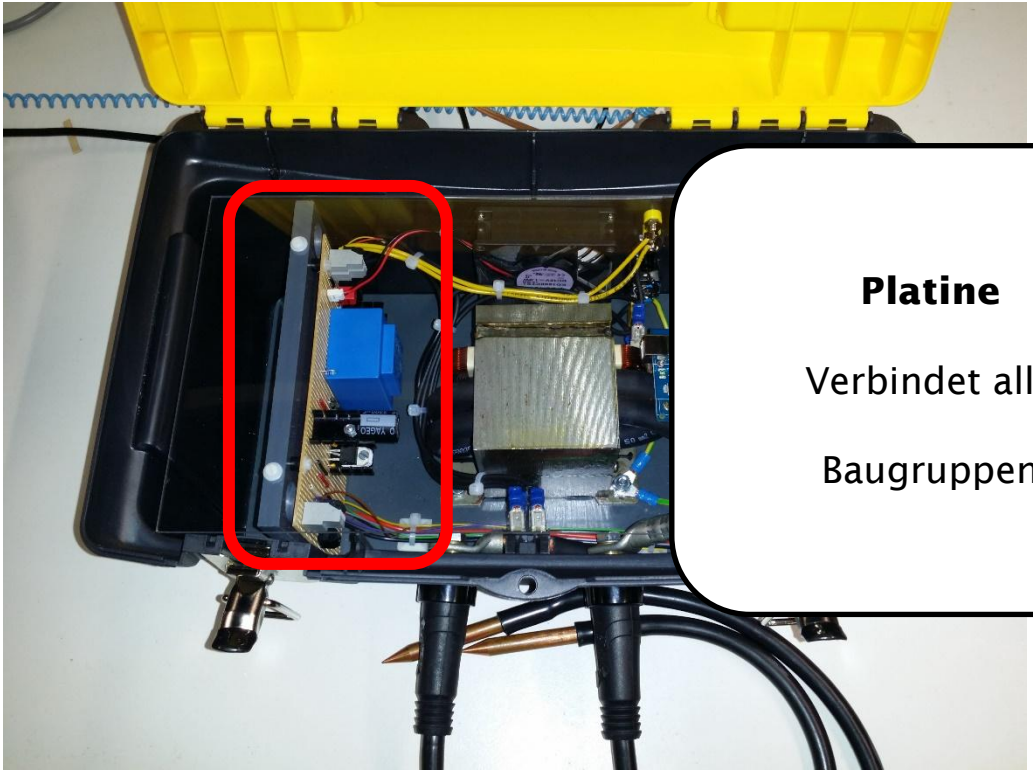
Baumarkt-Werkzeugbox

+ Praktische Verstaumöglichkeiten

+ Tragegriff

+ Display geschützt

Gehäuse

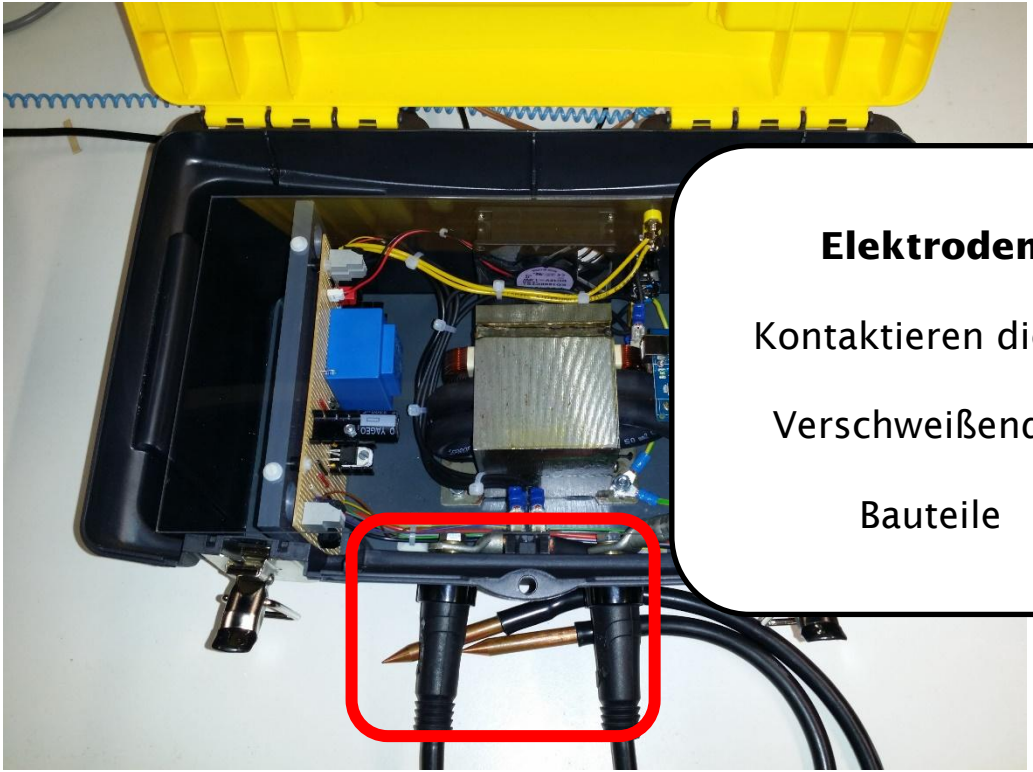


Platine

Verbindet alle

Baugruppen

Gehäuse



Elektroden

Kontaktieren die zu Verschweißenden Bauteile

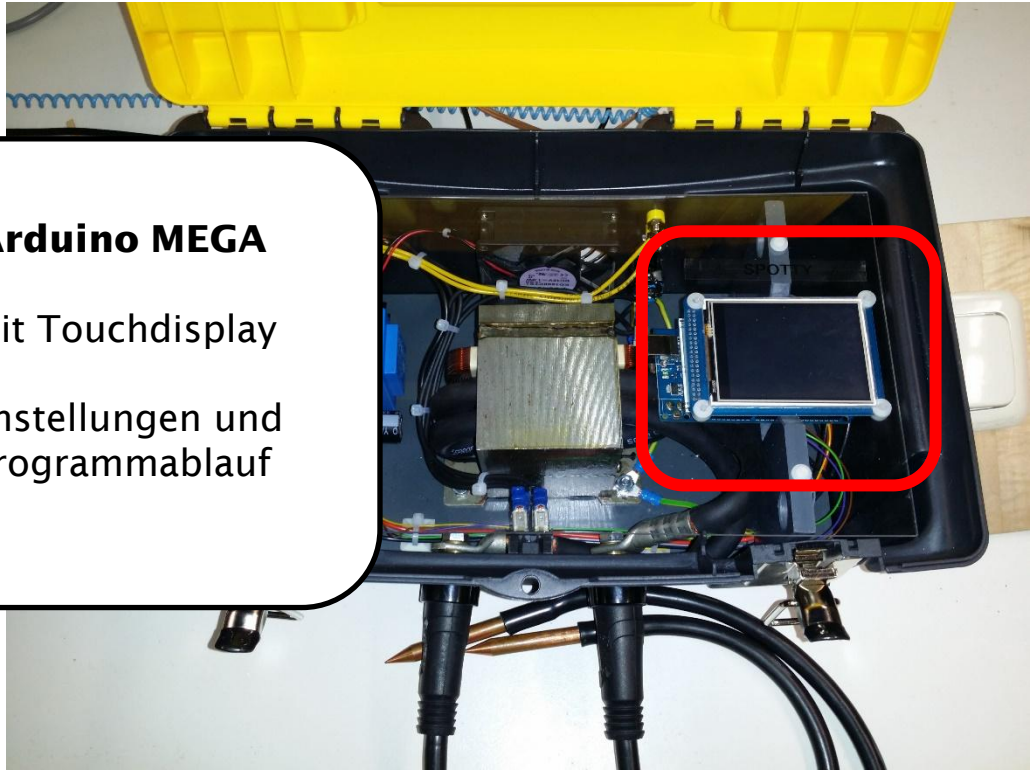
Gehäuse

Mikrowellentrafo

sorgt für den nötigen

Schweißstrom

Gehäuse

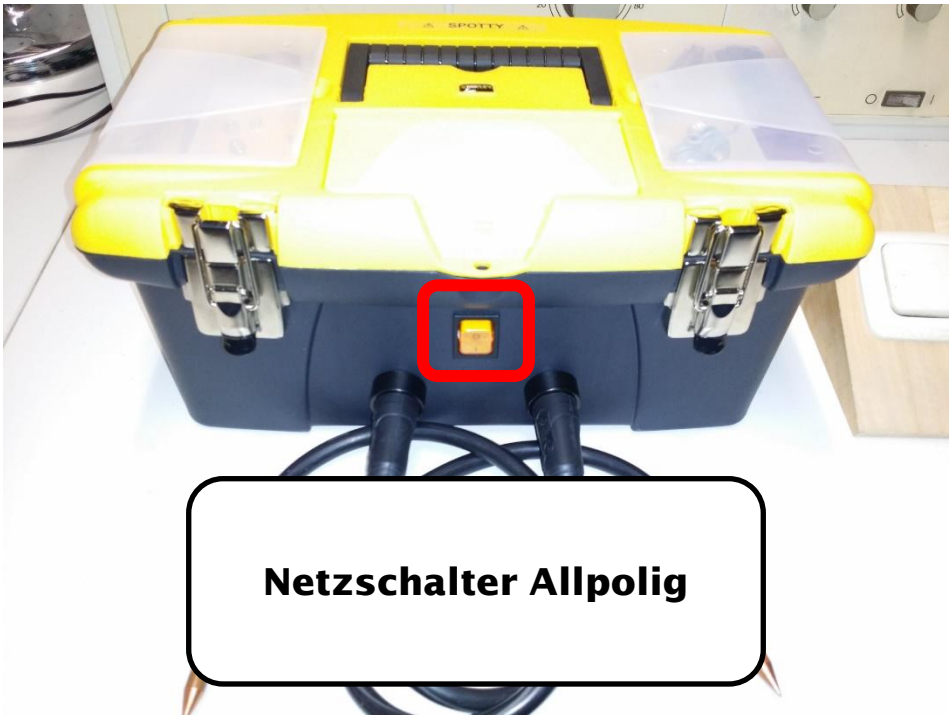


Arduino MEGA

mit Touchdisplay

Einstellungen und
Programmablauf

Gehäuse



Gehäuse

Anschluss Fußtaster

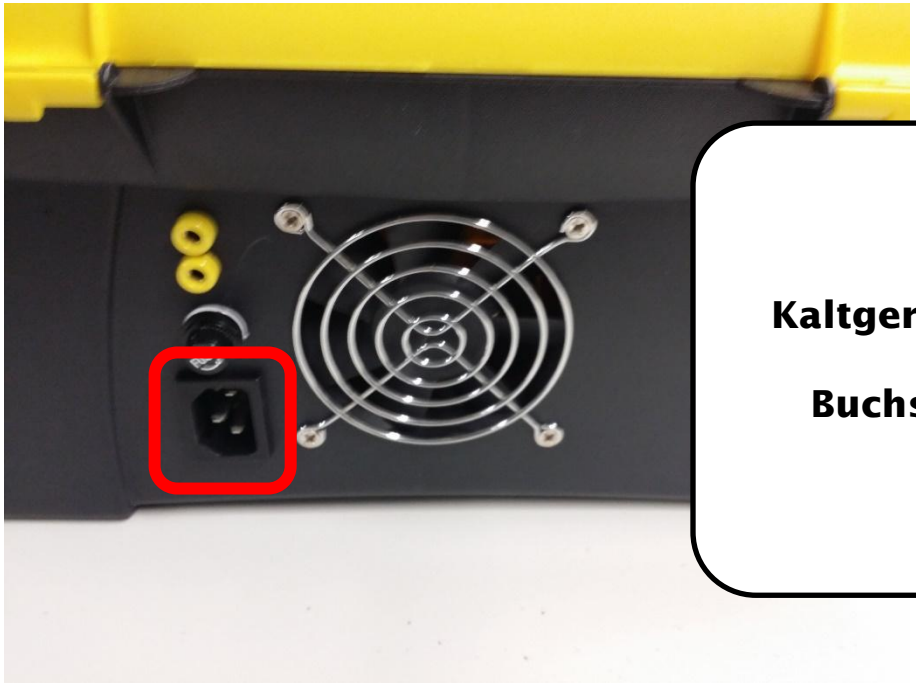
4mm Buchsen

Gehäuse

Geräte-Sicherung

Träge 4A

Gehäuse

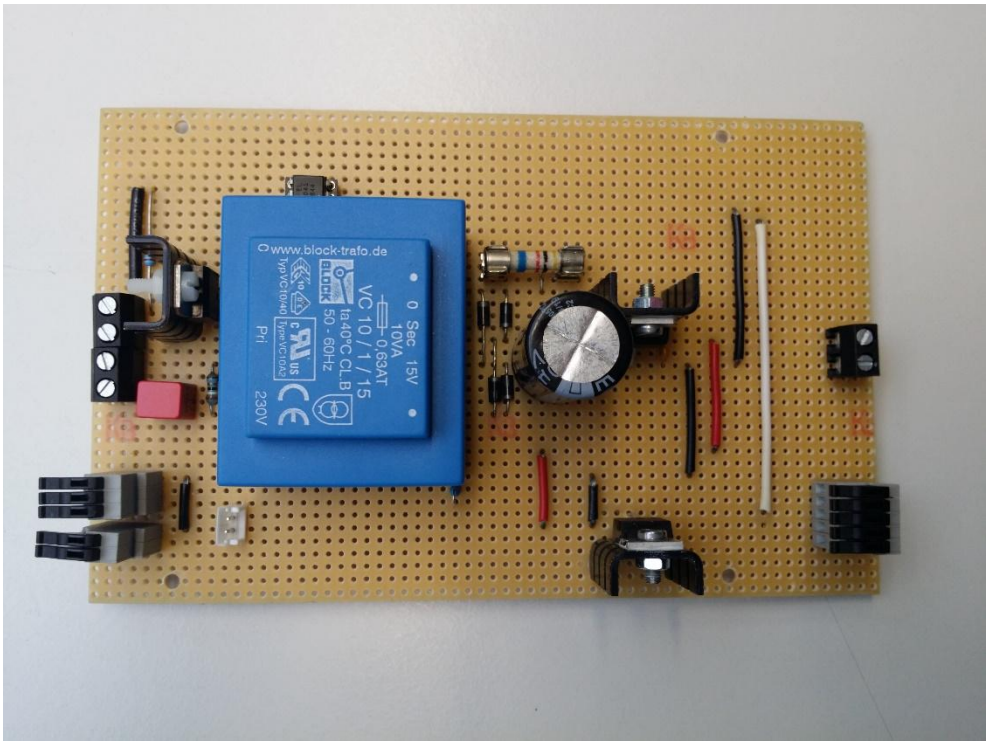


Gehäuse

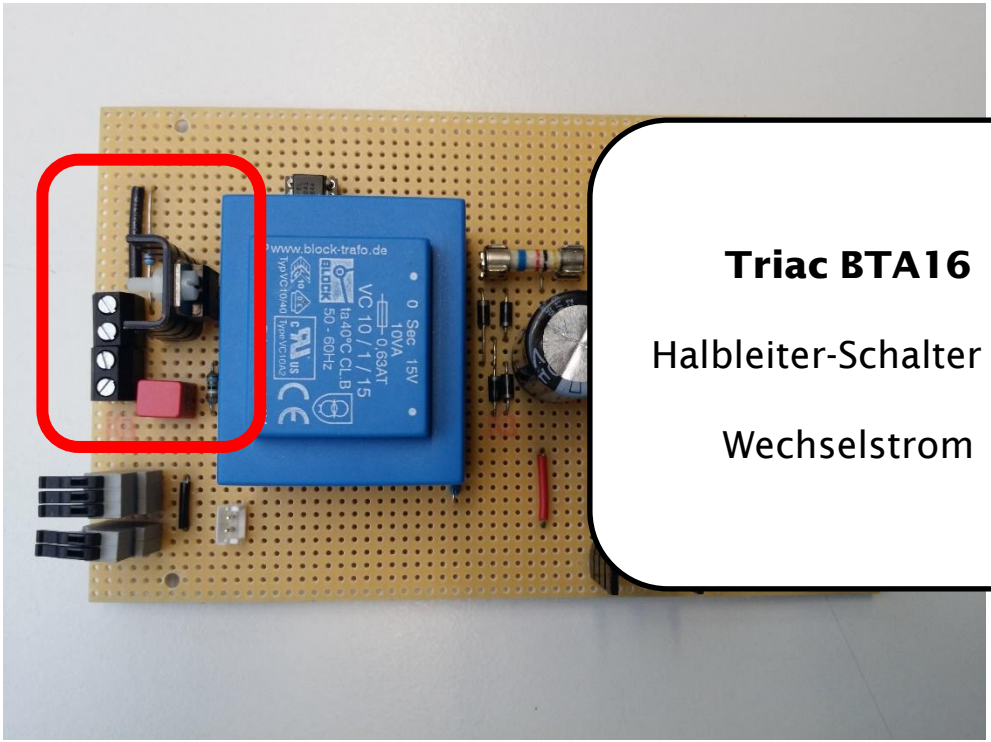
Ventilator

80mm
12V
0,25A

Platine


www.block-trafo.de
0 Sec 15V
10VA
0.63AT
VC 10 / 1 / 15
ta 40°C CL B
50 - 60Hz
Pri
230V

Platine



Triac BTA16

Halbleiter-Schalter für

Wechselstrom

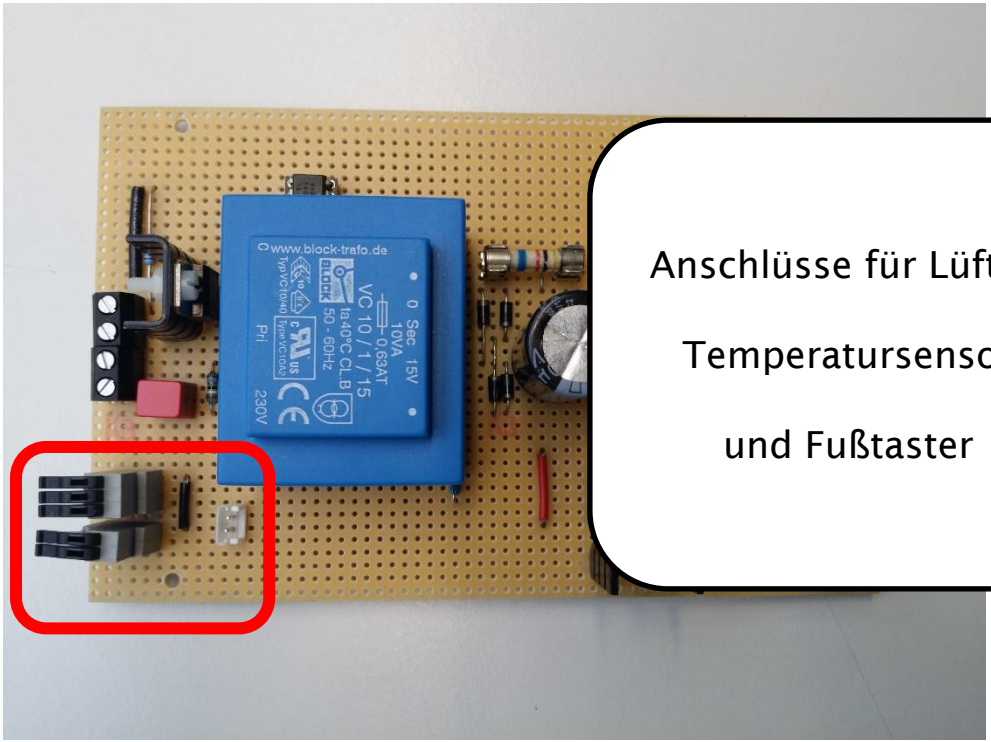
Platine

Optokoppler MOC3041

Ansteuerung Triac

Sichere Trennung
230V zu Kleinspannung

Platine

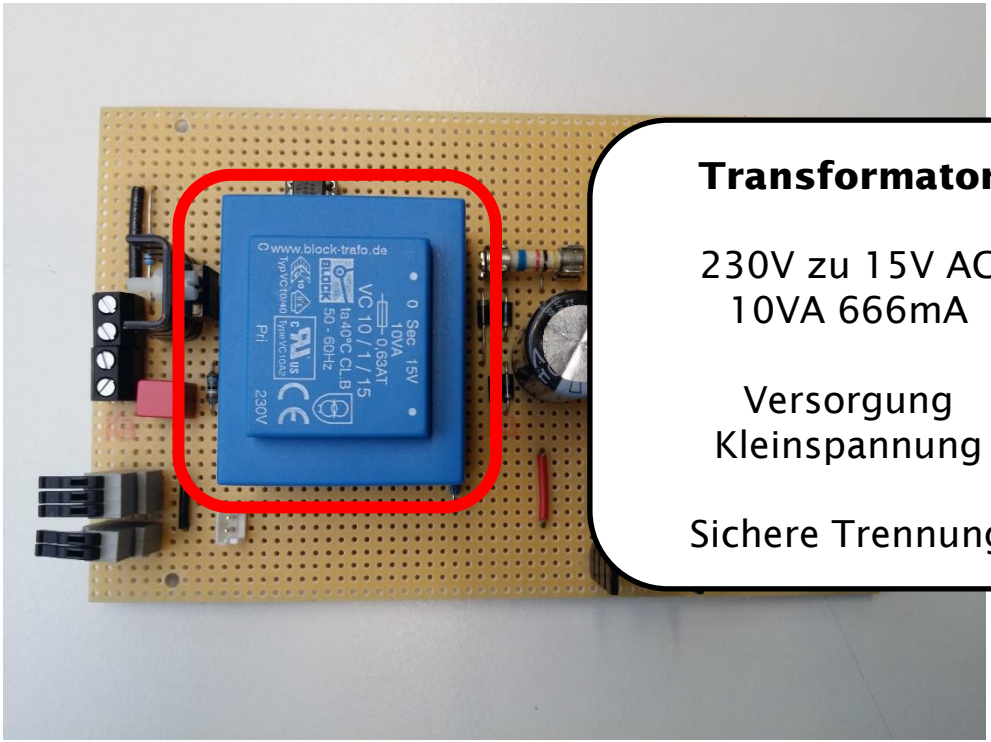


Anschlüsse für Lüfter,

Temperatursensor

und Fußtaster

Platine



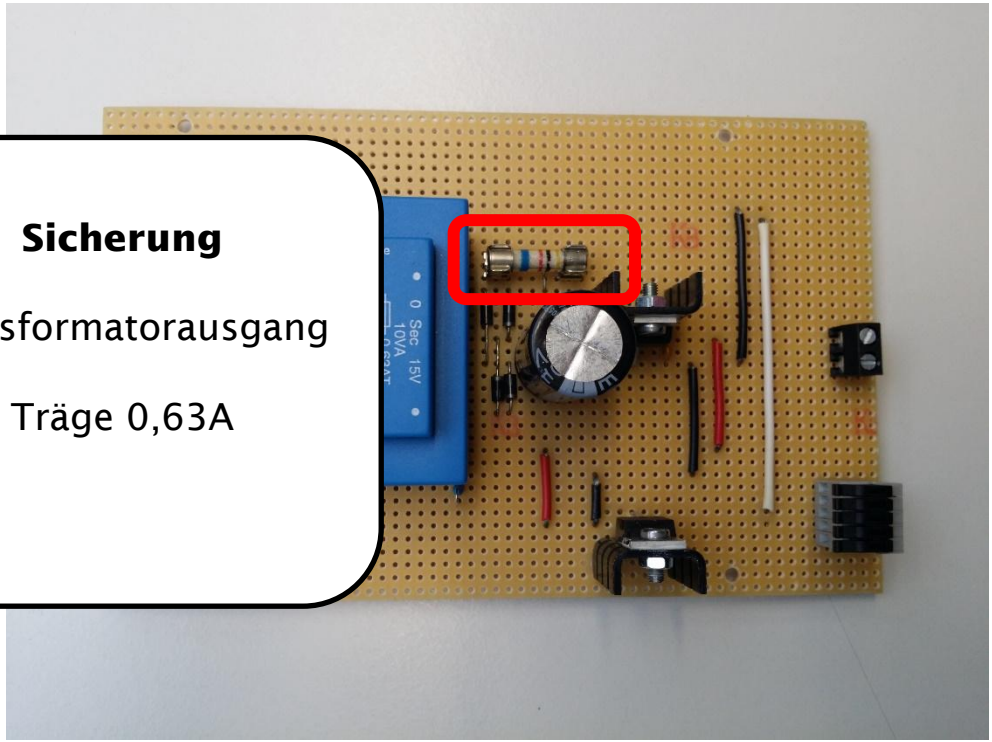
Transformator

230V zu 15V AC
10VA 666mA

Versorgung
Kleinspannung

Sichere Trennung

Platine

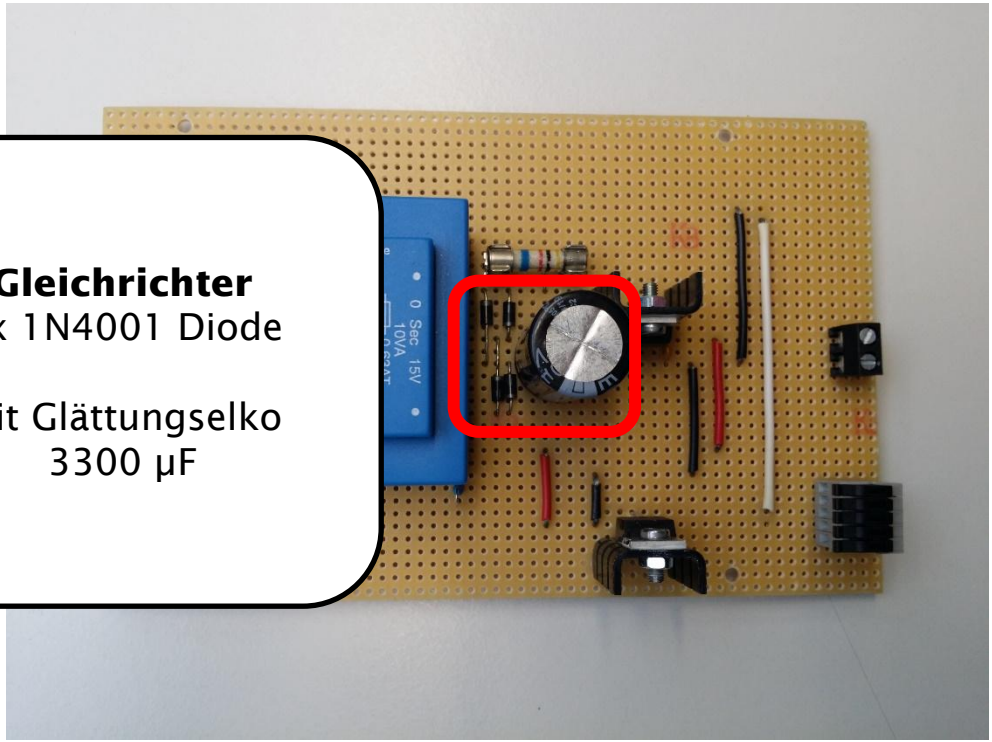


Sicherung

Transformatorausgang

Träge 0,63A

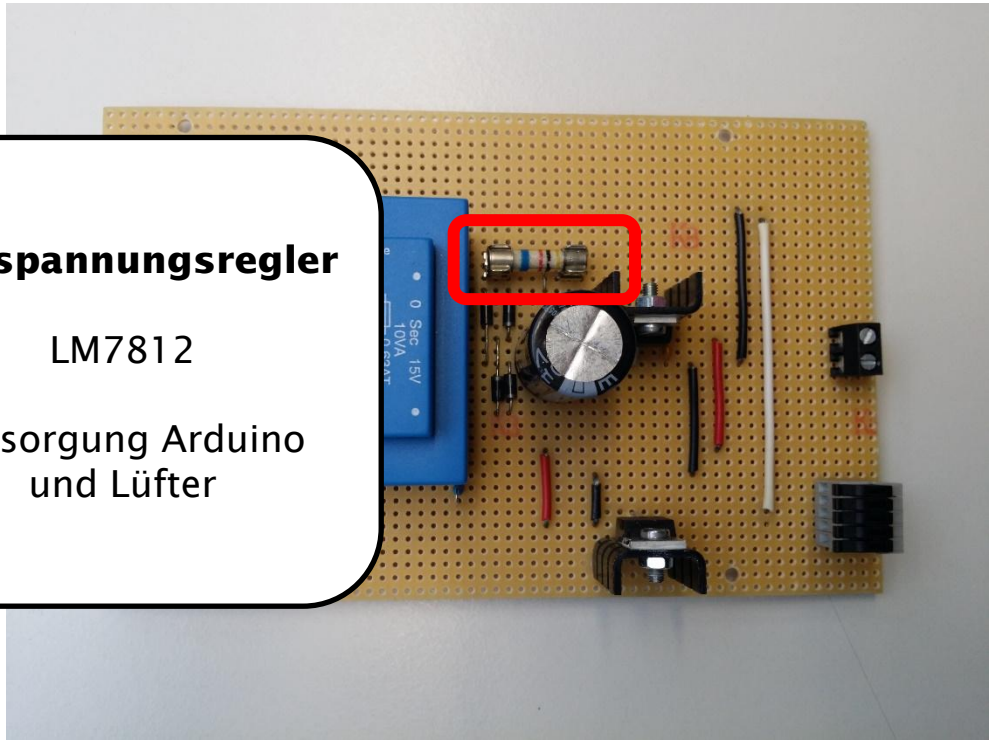
Platine



Gleichrichter
4x 1N4001 Diode

Mit Glättungselko
3300 µF

Platine

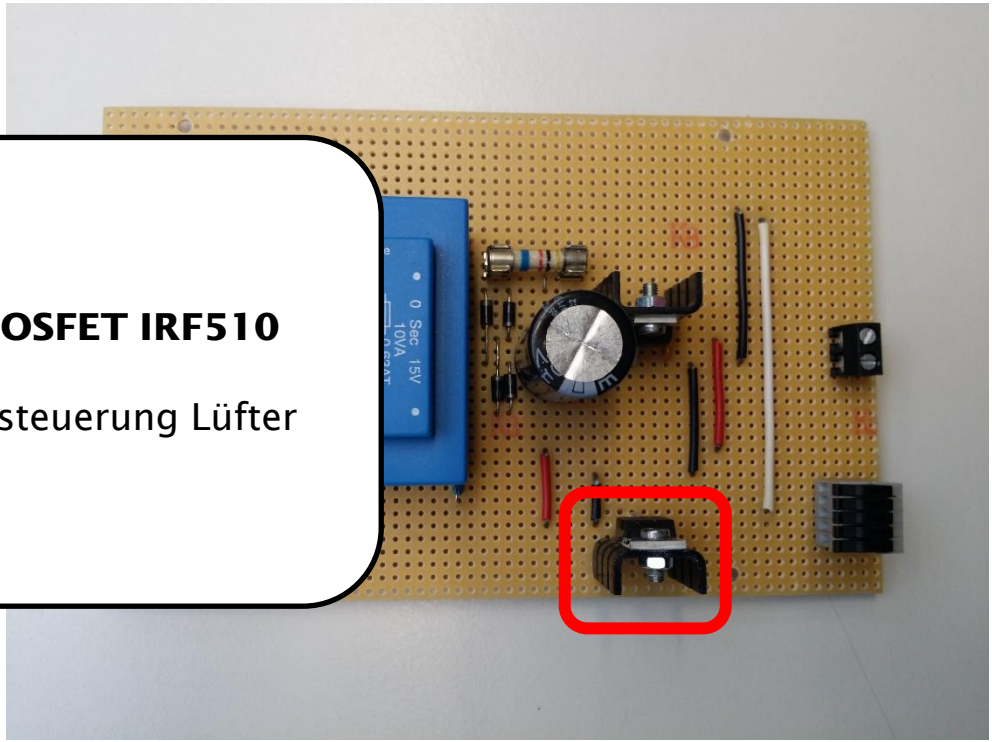


Festspannungsregler

LM7812

Versorgung Arduino
und Lüfter

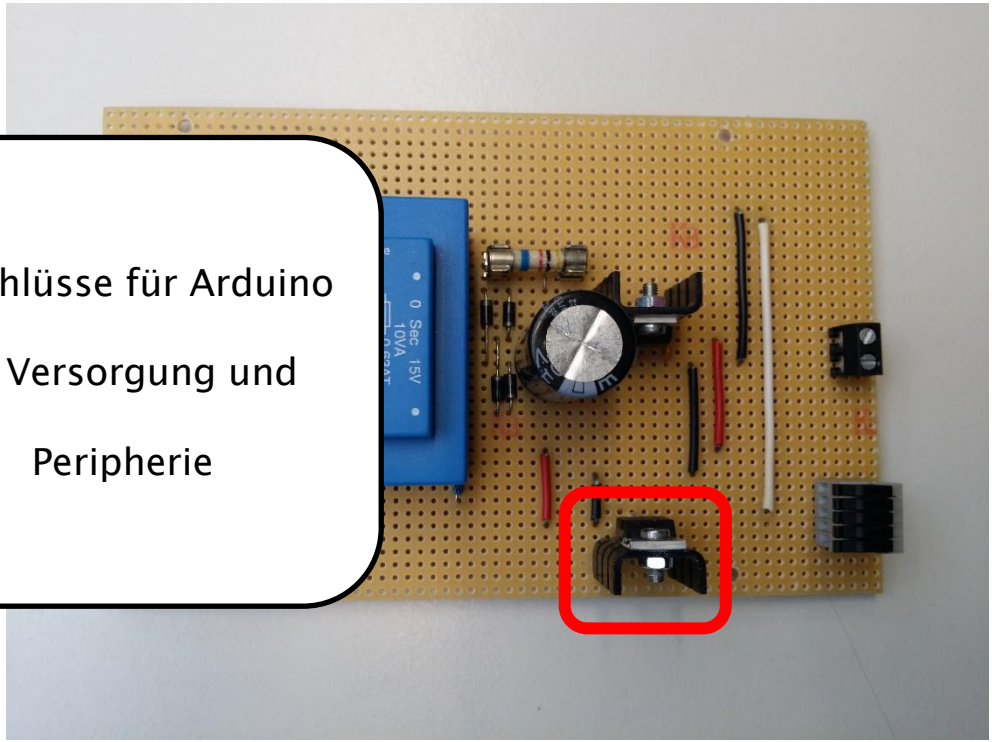
Platine



MOSFET IRF510

Ansteuerung Lüfter

Platine

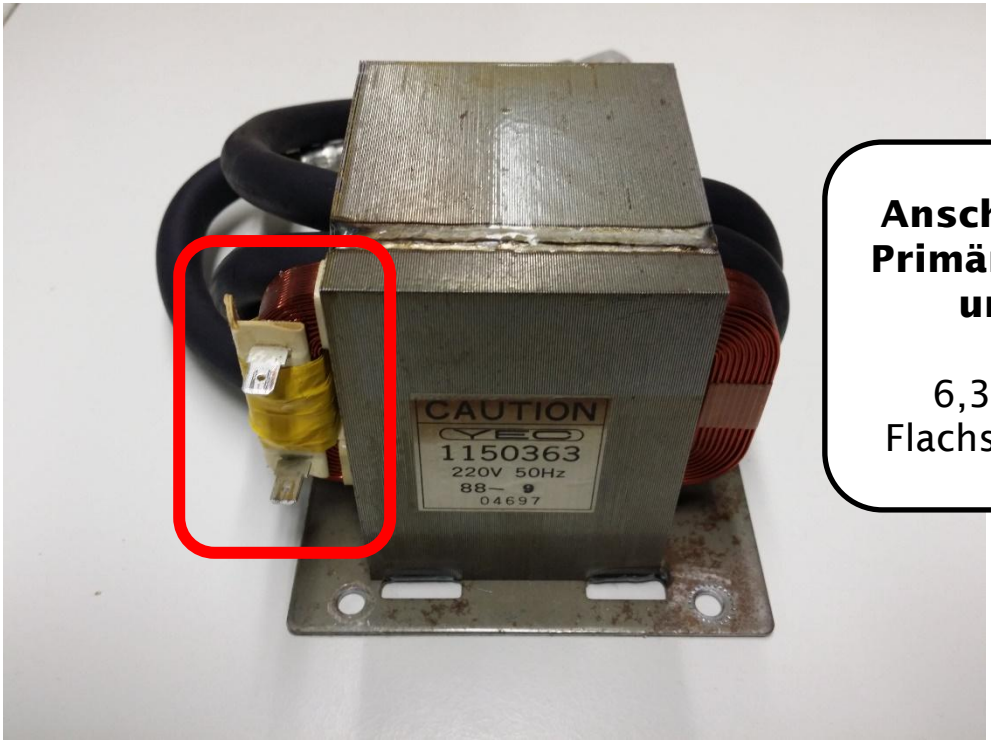


Anschlüsse für Arduino

12 Versorgung und

Peripherie

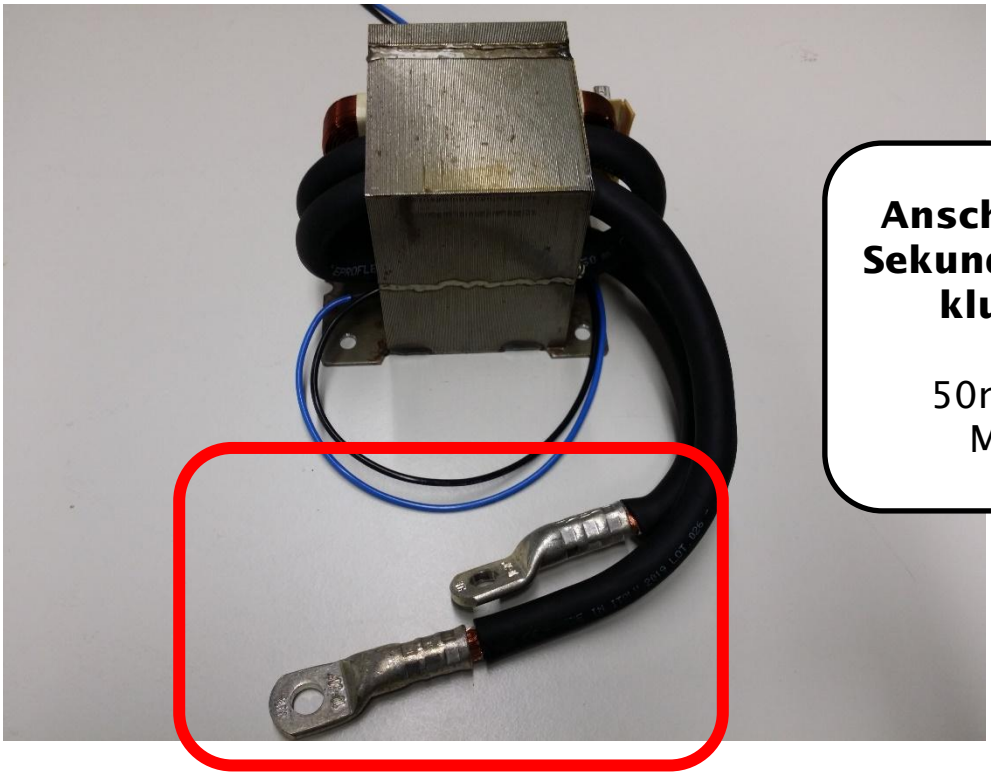
Transformator



Anschlüsse Primär-wicklung

6,3mm Flachstecker

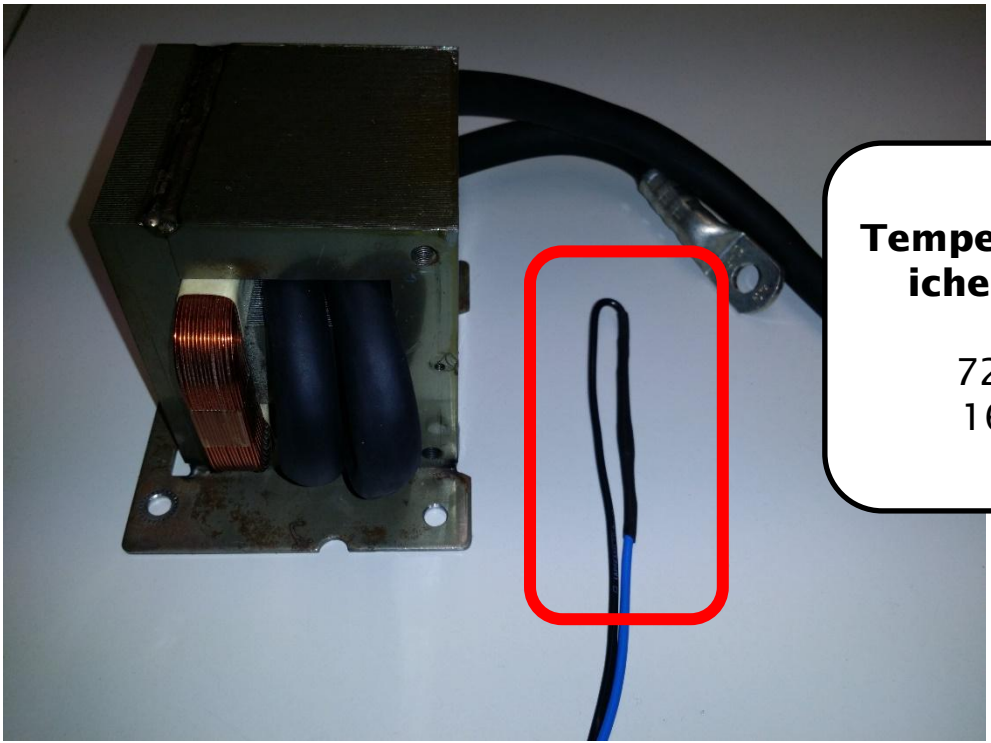
Transformator



Anschlüsse Sekundär-wicklung

50mm²
M8

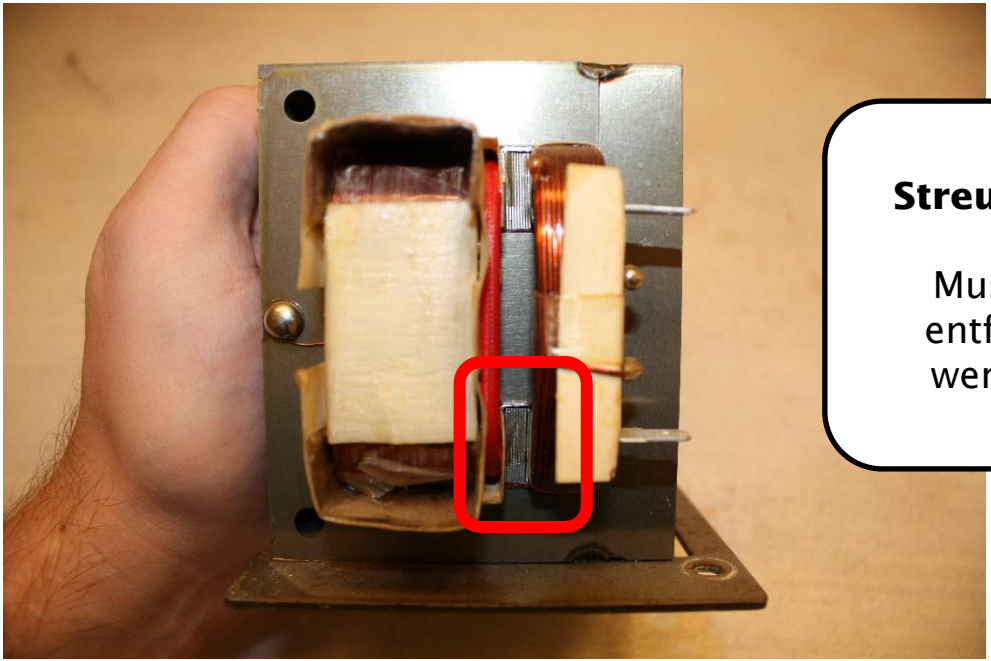
Transformator



Temperatur-sicherung

72°C
16A

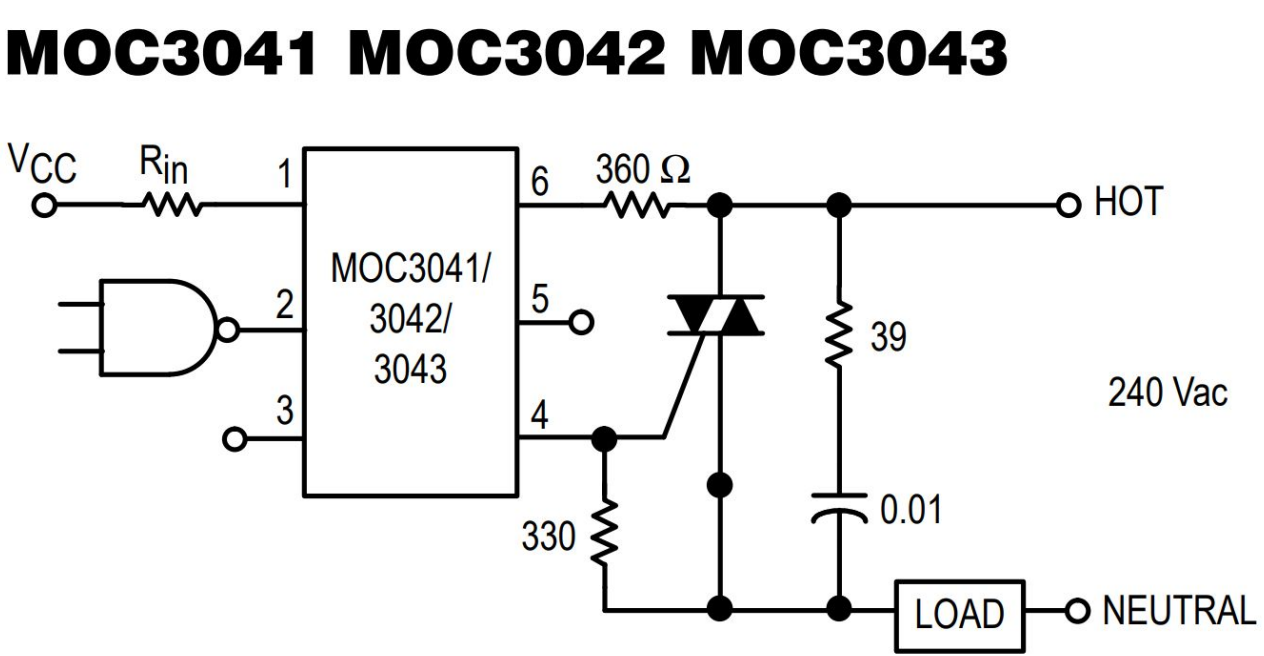
Transformator



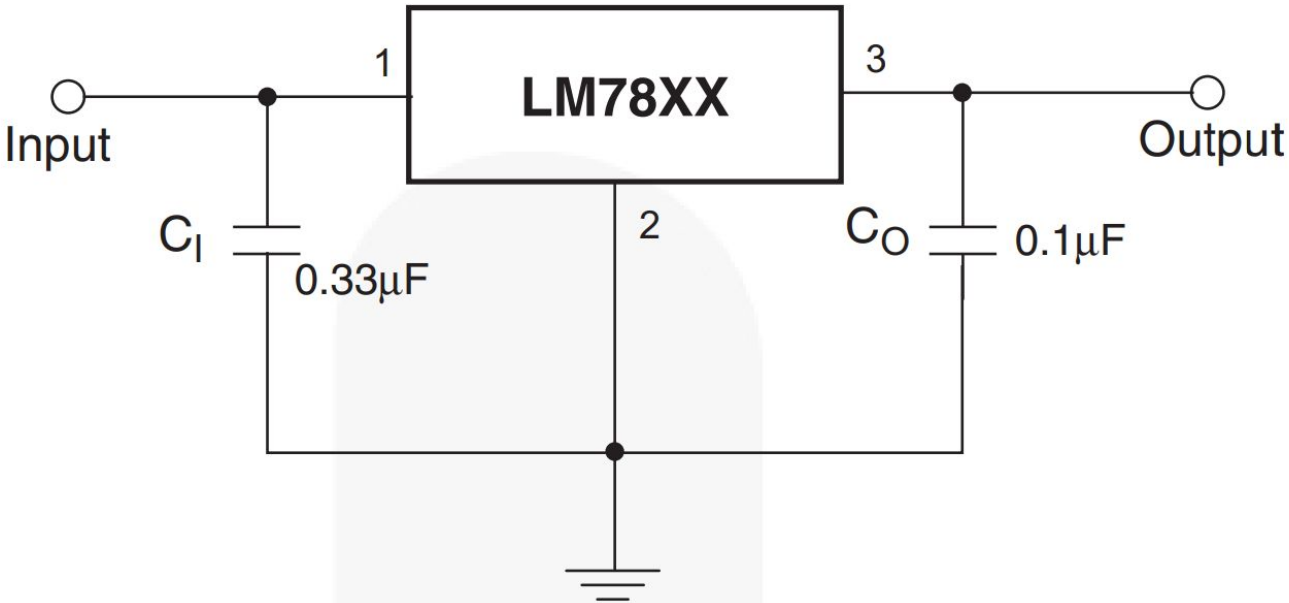
Streu-Joch

Musste entfernt werden

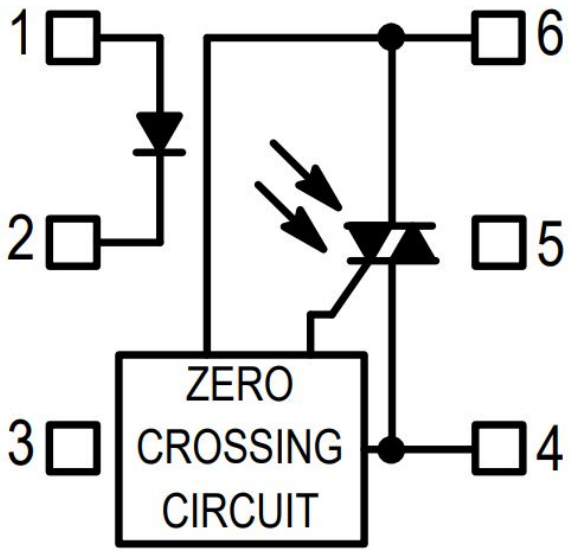
Schaltung


MOC3041 MOC3042 MOC3043
VCC
Rin
1
2
3
MOC3041/
3042/
3043
6
5
4
360 Ω
HOT
39
240 Vac
0.01
330
LOAD
NEUTRAL

Schaltung Festspannungsregler

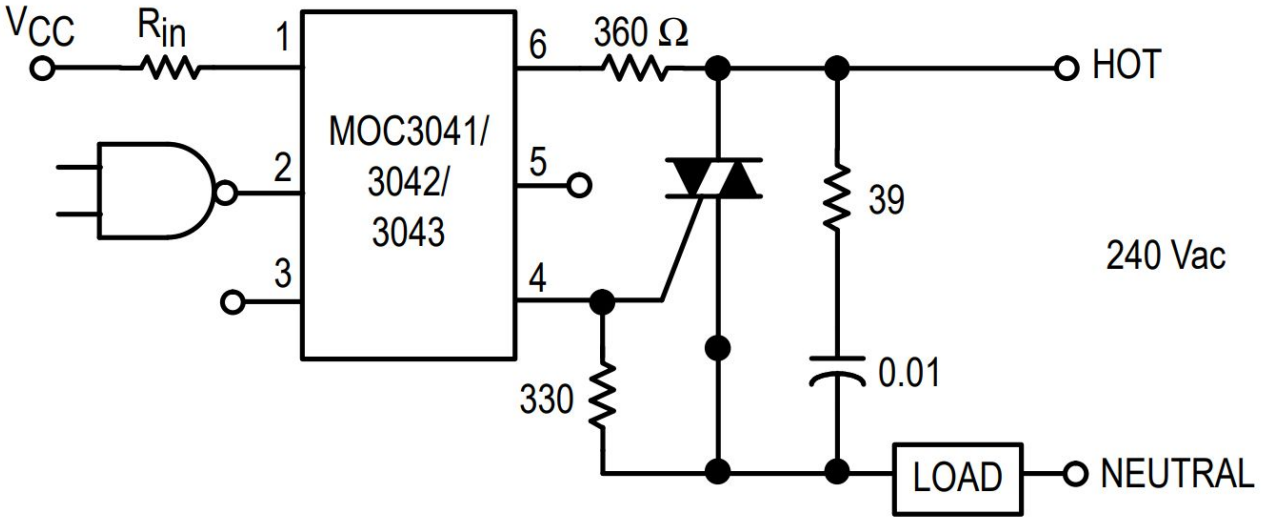


Schaltung Optokopler



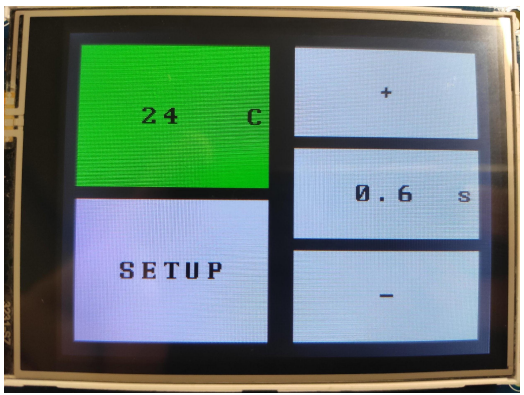
Schaltung Halbleiterschalter

MOC3041 MOC3042 MOC3043

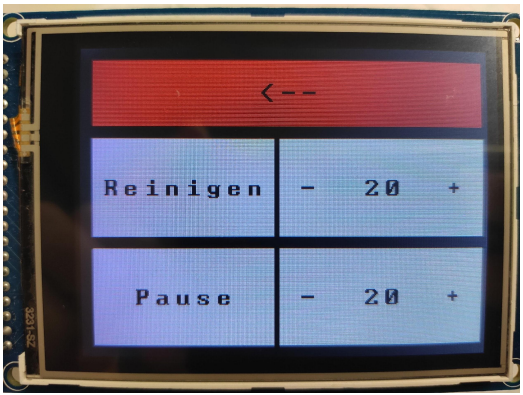


Software

- Zeichenausgabe mit UTFT.h
- 2 Menüs zur übersichtlichen Bedienung
- Farbwechsel des Temperaturfeldes bei Überschreitung einer Grenztemperatur



Menü 1



Menü 2

Software

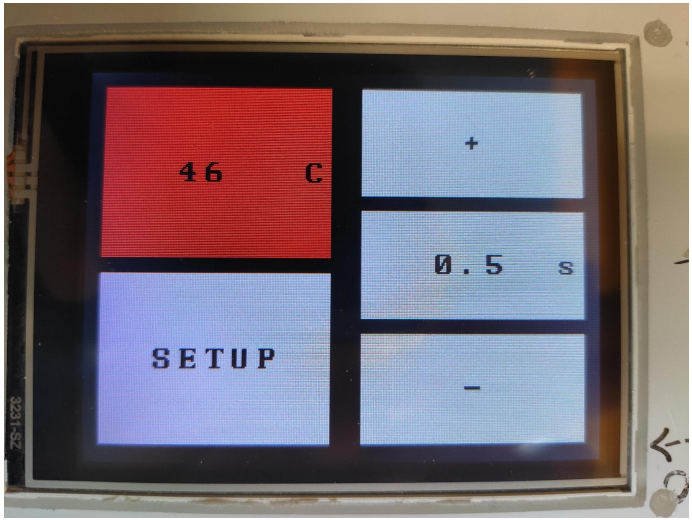
- Methode Rechteckzeichnen der Klasse Menü

```
void Rechteckzeichnen(int XKoord1, int YKoord1, int XKoord2, int YKoord2, String AUSGABE, int Farbe)
{
  switch (Farbe) {
  case 1 :
  myGLCD.setColor(Weiss[0], Weiss[1], Weiss[2]);
  myGLCD.setBackColor(Weiss[0], Weiss[1], Weiss[2]);
  break;
  case 2 :
  myGLCD.setColor(Schwarz[0], Schwarz[1], Schwarz[2]);
                :
                :
  myGLCD.fillRect(XKoord1, YKoord1, XKoord2,YKoord2);//Xunten, Yunten, Xoben, Yoben;
  myGLCD.setFont(BigFont);
  myGLCD.setColor(Schwarz[0], Schwarz[1], Schwarz[2]);

  int AusgabeX, AusgabeY;

  AusgabeX = (((XKoord1-XKoord2)- (AUSGABE.length()*16))/2) + XKoord2;
  AusgabeY = YKoord1-((YKoord1-YKoord2)/2+8);

  myGLCD.print(AUSGABE,AusgabeX,AusgabeY);
```

Software

- Aufruf der Methode Rechteckzeichnen

```
m1.Rechteckzeichnen(SETUP[0], SETUP[1], SETUP[2], SETUP[3], "SETUP", 1);
```



Software

- Der Schweißvorgang

```
timer++;
delay (1);


----------------------------------------Ausgang(Schweißen)----------------------

    if (digitalRead(TasterPin) == LOW && Temperatur < 60)//Taster gedrückt
      {timer = 0;}

    if(timer == 500)                    //500 entspricht 0,5s
    { digitalWrite(TrafoPin, HIGH);}

    if(timer == 500 + Reinigungszeit)
      { digitalWrite(TrafoPin, LOW);}

    if(timer == 500 + Reinigungszeit + Pausenzeit)
      { digitalWrite(TrafoPin, HIGH); }

    if(timer == 500 + Reinigungszeit + Pausenzeit + Dauerzeit*1000)
      { digitalWrite(TrafoPin, LOW); }

    if(timer >1500)
    {
       sensors.requestTemperatures();
       Temperatur = sensors.getTempCByIndex(0);
    }
```