

Bohrmaschinen Steuerung

mit Bohrfeldbeleuchtung



Entwickelt und gebaut von:
Ulrich Laag
Okt. – Dez. 2025

Inhalt

| | | |
|----------|---|-----------|
| 1 | Übersicht und Bedienung | 3 |
| 1.1 | Bedienung | 3 |
| 2 | Beschreibung der Schaltung | 3 |
| 2.1 | Versorgung und Signale | 3 |
| 2.2 | Sicheres Abschalten | 3 |
| 2.3 | Schaltplan | 4 |
| 3 | Aufbau der Lampe | 5 |
| 3.1 | Bestückungsseite: | 5 |
| 3.2 | Lötseite: | 5 |
| 4 | Das Programm | 6 |
| 4.1 | Ablaufdiagramm der Hauptschleife | 6 |
| 4.2 | Programmcode | 7 |
| 5 | Bilder | 12 |
| 5.1 | Die Steuerung | 12 |
| 5.2 | Die Bedieneinheit | 14 |
| 5.3 | Die Bohrfeldbeleuchtung | 15 |
| 5.4 | Die Praxis | 18 |

1 Übersicht und Bedienung

Die Handbohrmaschine im Bohrstand wird, wie eine stationäre Bohrmaschine, über zwei Tasten bedient. Die Bohrfeldbeleuchtung benötigt keine eigenen Bedienelemente. Die Leuchtdauer orientiert sich am Lauf der Bohrmaschine.

Die Art der Ansteuerung ist einem Sicherheitsrelais nachempfunden und enthält mehrere redundante Elemente, was die Betriebssicherheit erhöht.

1.1 Bedienung

1.1.1 Einschalten

Durch Drücken der schwarzen Ein-Taste wird die Bohrfeldbeleuchtung sofort eingeschaltet. Um die Bohrmaschine zu starten, muss die Taste aus Sicherheitsgründen 2 Sekunden lang gedrückt gehalten werden. Die Beleuchtung bleibt aktiv, solange die Bohrmaschine läuft. Nach dem Ausschalten leuchtet sie noch 3 Minuten nach.

1.1.2 Ausschalten

Durch Drücken der roten Aus-Taste wird die Bohrmaschine sofort abgeschaltet. Die Beleuchtung bleibt für 3 Minuten aktiv, um beispielsweise ein neues Ausrichten des Werkstücks zu erleichtern. Mit einem weiteren Druck auf die Aus-Taste kann die Beleuchtung ausgeschaltet werden.

1.1.3 Einstellung der Helligkeit

Mit dem Potentiometer kann die Helligkeit der Bohrfeldbeleuchtung eingestellt werden. Dreht man den Regler an den linken Anschlag, bleibt die Beleuchtung dauerhaft aus.

2 Beschreibung der Schaltung

Der Schaltplan ist auf der nächsten Seite in Kapitel 2.3.

2.1 Versorgung und Signale

Bohrfeldbeleuchtung und Relais können von der unstabilisierten Gleichspannung versorgt werden. Der Mikrocontroller (MC) braucht zur sicheren Funktion eine stabile Spannung von 5V, die A1 liefert. Die Signale der Tasten (S1, S2) führen die hohe Betriebsspannung U_b . Mit Hilfe der Widerstände R2...R5 und der Z-Dioden D4, D5 wird sie auf den Logikpegel des MC reduziert.

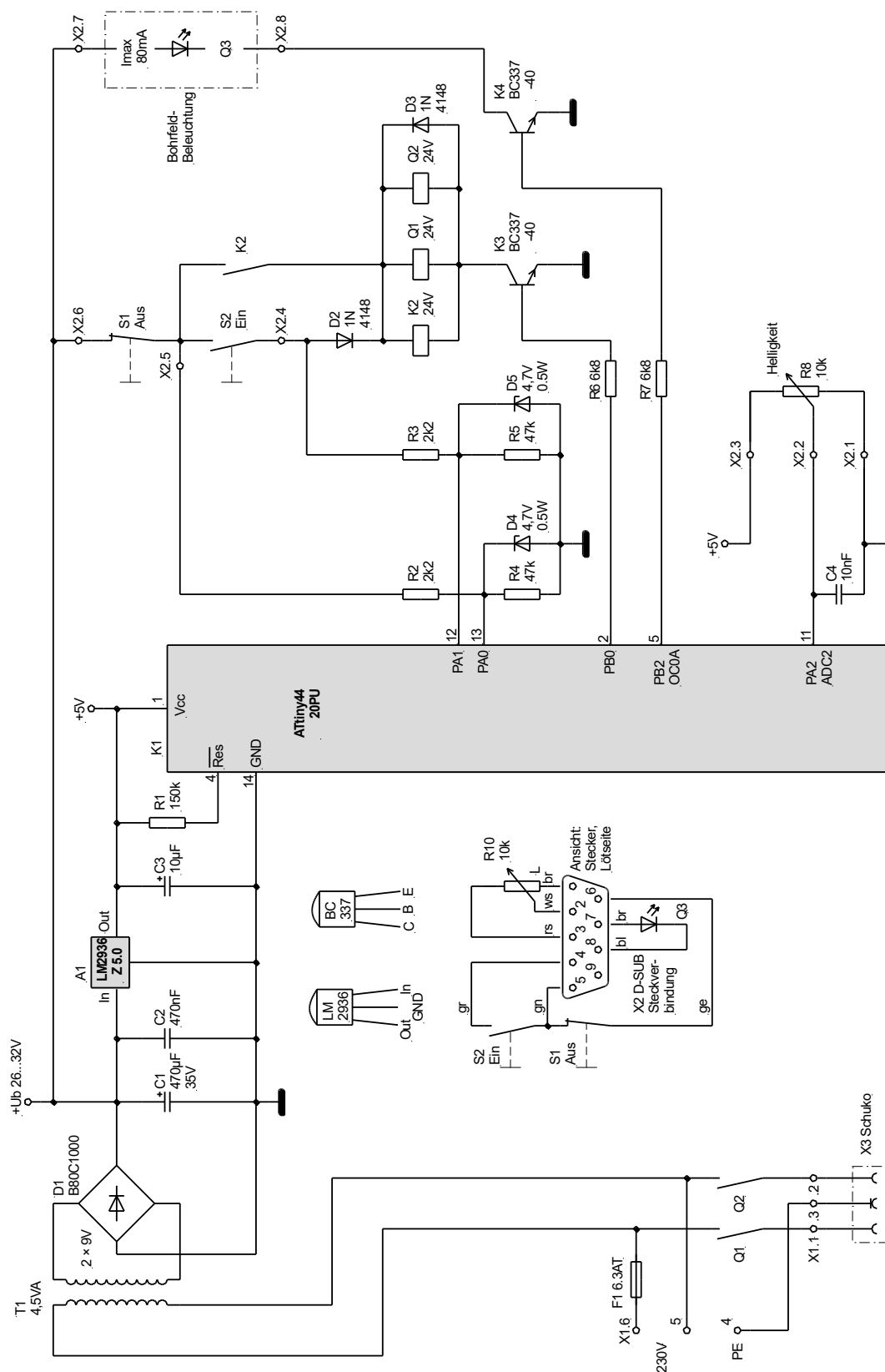
Durch D2 bekommt der Eingang PA1 nur High-Signal, wenn S2 betätigt ist.

2.2 Sicheres Abschalten

Durch Drücken der Aus-Taste, sperrt der MC den Transistor K3, wodurch die Relais abfallen. Sollte K3 nicht sperren, werden die Relais dennoch stromlos, da S1 öffnet. Durch das Abfallen der Selbsthaltung (K1) bleibt dieser Zustand auch bestehen, wenn S1 wieder schließt. Unabhängig von der Funktion des MC, S1 unterbricht den Strom zu den Relais zuverlässig.

Da zwei Lastrelais vorhanden sind (Q1, Q2), reicht es aus, wenn eines abfällt, um die Bohrmaschine zu stoppen.

2.3 Schaltplan



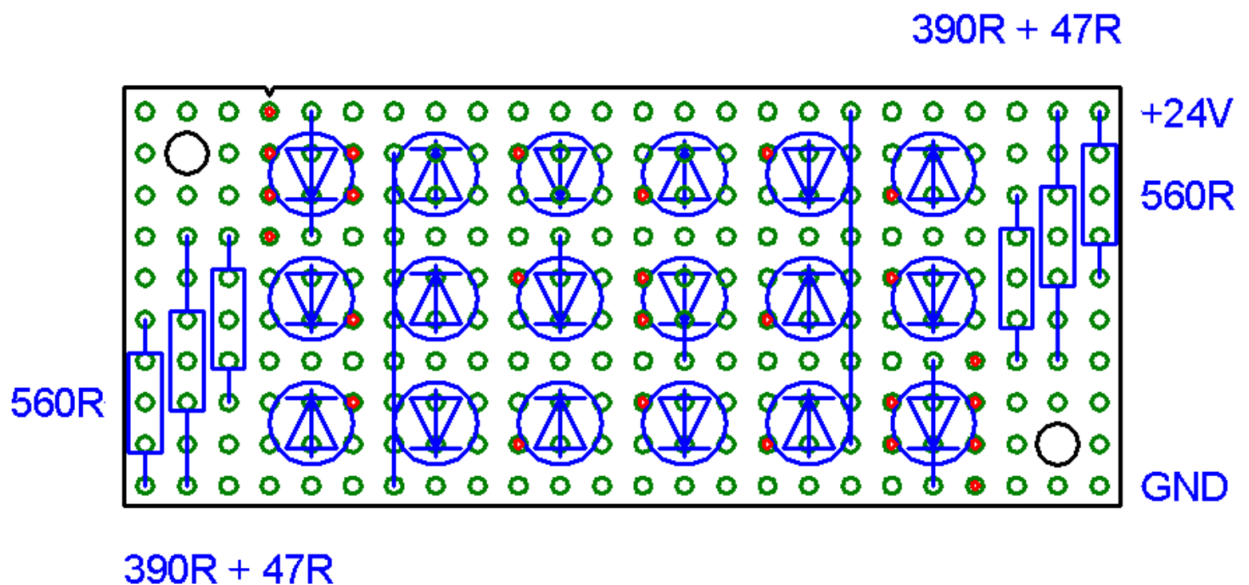
3 Aufbau der Lampe

Der Aufbau erfolgte auf einer Streifenrasterplatine. Die Streifen verlaufen in Längsrichtung.

LEDs: 5034W2C-BUC-E von Huiyuan
5mm, weiß, klar, 25000mcd, 15° Abstrahlwinkel

Zwei Reihen mit je 5 LED. Vorwiderstände je 390R + 47R
Zwei Reihen mit je 4 LED. Vorwiderstände 560R
Ergibt bei einer $U_F=2,8V$ und einer $U_B=24V$ einen I_F von 23mA
Und damit einen Gesamtstrom von 92mA. Das macht 2,2W

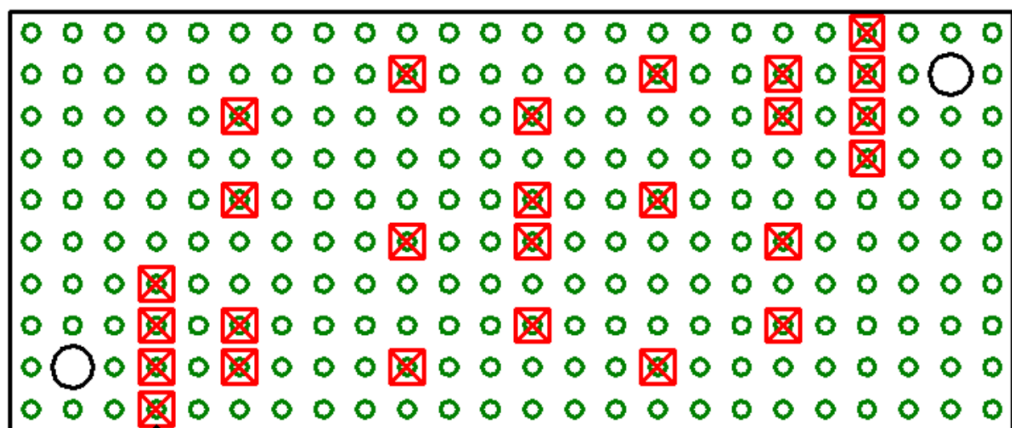
3.1 Bestückungsseite:



3.2 Lötseite:

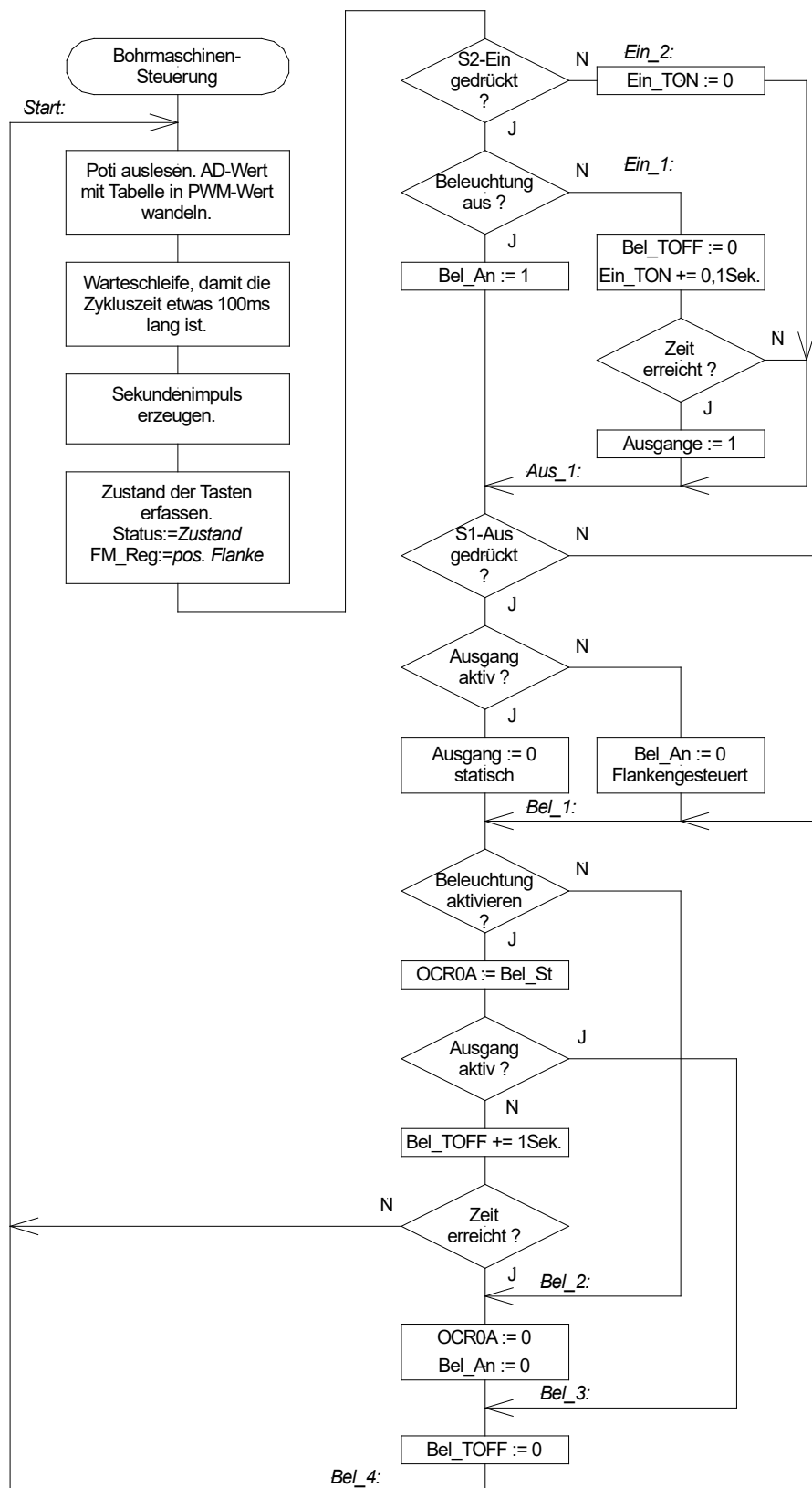
Gespiegelt dargestellt!

Die roten gekreuzten Quadrate sind Leiterbahnunterbrechungen. Auf der Bestückungsseite als rote Punkte dargestellt.



4 Das Programm

4.1 Ablaufdiagramm der Hauptschleife



4.2 Programmcode

```

/*****
*           Bohrmaschinen-Steuerung                V3 mit ATtiny44          *
*           Dez. 2025                               Ulrich Laag              *
*                                                                 *
* Für das Programm gibt es ein Ablaufdiagramm.                      *
* Die Funktionsbeschreibung steht in der Projektbeschreibung.        *
*                                                                 *
* Die CPU wird mit der niedrigen Taktfrequenz des Watchdog-Timers   *
* betrieben. Das erhöht die Betriebssicherheit und reicht für diese  *
* Anwendung. Die Zykluszeit der Programmhauptschleife beträgt 100ms. *
* Das reicht völlig für die Zeitnahe Auswertung der Tasten. Die Verar- *
* beitung der beiden Zeitwerte (Anzugsverzögerung TON über Ein-Taste; *
* Ausschaltverzögerung TOFF der Beleuchtung) wurde so aber sehr verein- *
* facht. Für die lange Nachlaufzeit der Beleuchtung, wurde ein      *
* Sekundentakt im Statusregister eingerichtet.                      *
*                                                                 *
* Flash : Code 232 Byte, Data 1024 Byte,  Auslastung 30,7%          *
*****/
;Wichtige Konstanten im Programm
.equ F_CPU = 128000          ;CPU-Takt 128kHz vom Watchdog-Timer
                             ;ohne Teiler durch 8
                             ;Die Watchdog-Freq. ist aber sehr ungenau!
.equ ZW = 15                ;ZeitWert für die Verzögerungsschleife.
                             ;Gemessen ist die Zykluszeit ca. 100ms
.equ t_Bel = 180            ;Nachlauf der Beleuchtung 180 × 1Sek = 3Min
.equ t_Ein = 20             ;Einschaltverzögerungszeit Ein-Taste 2Sek.

;Umbenennung der I/O-Register
.equ In_P  = PINA           ;Eingangsport der Bedienelemente
.equ Out_P = PortB          ;Relaissteuerung

;Position der I/O-Pins in den Ports
.equ AusTa  = 0             ;Aus-Taste                                im In_P (PINA)
.equ EinTa  = 1             ;Ein-Taste                                im In_P (PINA)
.equ Relais = 0             ;Ansteuerung der Relais                  im Out_P (PortB)
.equ LED_Bel = 2            ;Ausgang für die Beleuchtung im Out_P (PortB)

;Funktionsnamen für die GP-Register
.def AD_L    = r9           ;Low -Byte des 10Bit AD-Wert
.def AD_H    = r10          ;High-Byte des 10Bit AD-Wert
.def Bel_St   = r11          ;Stärke der Beleuchtung die zum OCR0A-Reg. geht
.def Ein_TON  = r12          ;Tick-Zähler, Einschaltverzögerung der Ein-Taste
.def Bel_TOFF = r13          ;Tick-Zähler, zur Abschaltung der Beleuchtung
.def St_old   = r15          ;der Status aus dem letzten Zyklus
.def temp     = r16          ;erstes allgemeines Arbeitsregister
.def thelp    = r17          ;zweites allgemeines Arbeitsregister
.def Status   = r18          ;Status Register
.def FM_Reg   = r19          ;Flankenmarker für positive Flanken
.def Sekunde  = r20          ;Zählt von 0...10 und erzeugt den Sekundentakt

```

```

;Position der Flags im Status- und FM-Register
.equ Aus      = 0   ;True wenn die Taste Aus ist gedrückt ist
.equ Ein      = 1   ;True wenn die Taste Ein ist gedrückt ist
.equ Bel_An   = 2   ;True wenn die Beleuchtung an ist
.equ Sek_T    = 3   ;Sekundentakt

;//////////////////////////////////////
; Belegung des Flash-Speichers
;//////////////////////////////////////
.cseg
.org 0
;Vektortabelle der Interrupts (es gibt keine)
Reset: rjmp letsgo ; Reset
       rjmp no_isr ; ext. IRQ0
       rjmp no_isr ; Pin Change INT0
       rjmp no_isr ; Pin Change INT1
       rjmp no_isr ; Watchdog Timer
       rjmp no_isr ; Timer1 Capture
       rjmp no_isr ; Timer1 Compare A
       rjmp no_isr ; Timer1 Compare B
       rjmp no_isr ; Timer1 Overflow
       rjmp no_isr ; Timer0 Compare A
       rjmp no_isr ; Timer0 Compare B
       rjmp no_isr ; Timer0 Overflow
       rjmp no_isr ; Analog Comparator
       rjmp no_isr ; ADC Conversion Complete
       rjmp no_isr ; EEPROM Ready
       rjmp no_isr ; USI Start Handler
       rjmp no_isr ; USI Overflow Handler
no_isr: reti

;Initialisierung
letsgo: ldi temp, LOW(RAMEND) ;Stackpointer für den Systemstack
       out SPL, temp        ;einrichten.
       ldi temp, HIGH(RAMEND) ;wird aber nicht gebraucht
       out SPH, temp

       ;Timer 0 im Phase-Correct-PWM-Mode. Mit PWM-Frequenz = 250Hz
       ;Ausgangspin ist OC0A. Je höher der Wert im OCR0A-Register
       ;ist, um so heller leuchtet die Bohrplatz-Beleuchtung
       ldi temp, (1<<COM0A1)|(1<<WGM00)
       out TCCR0A,temp
       ldi temp, (1<<CS00)
       out TCCR0B,temp
       clr temp
       out OCR0A,temp ;Lampe aus

       ;Den AD-Wandler einrichten. Er wird einmal im Hauptprogramm
       ;gestartet. Es werden nur die höchsten 8Bit im ADCH ausgewertet
       ldi temp, (1<<ADEN) ;AD-Wandler aktivieren
       out ADCSRA,temp     ;Wandlungsfrequenz halbe F_CPU = 64kHz
       ldi temp, (1<<MUX1) ;AD-Wandler an ADC2
       out ADMUX,temp      ;Uref ist die Betriebsspannung
       clr temp            ;10Bit Wandlung
       out ADCSRB,temp     ;Free Running Mode

```



```

sbi DIDR0,ADPS2      ;Stromsparen am AD-Eingang

ldi temp,(1<<Relais)|(1<<LED_Bel)
out DDRB,temp        ;die Ausgänge aktivieren
clr temp
out Out_P,temp        ;Alle Ausgänge aus
mov Status,temp       ;Status und Flankenmerker zurücksetzen
mov FM_Reg,temp

clr Sekunde

;//////////////////////////////////////
; Das Hauptprogramm
;//////////////////////////////////////
Start: sbi ADCSRA,ADSC      ;AD-Wandlung starten
Mess:  sbis ADCSRA,ADIF     ;und warten, bis sie abgeschlossen ist
       rjmp Mess
       in AD_L,ADCL         ;neuen Messwert einlesen
       in AD_H,ADCH

       ldi z1,low(PWM_Wert<<1) ;Jedem Messwert ist ein PWM-Wert zuge-
       ldi zh,high(PWM_Wert<<1) ;ordnet. Die Werte stehen in der Tabelle
       add z1,AD_L          ;PWM_Wert. Der Messwert wird gegen den
       adc zh,AD_H          ;PWM_Wert getauscht und bestimmt damit
       lpm Bel_ST,z         ;die Beleuchtungs-Stärke.

wait:  ldi thelp,ZW          ;Die Zykluszeit der Hauptschleife wird im
       dec temp             ;wesentlichen durch diese Warteschleife
       brne wait           ;bestimmt. Mit der Zahl ZW lässt sich die
       dec thelp           ;Zykluszeit einstellen.
       brne wait

       cbr Status,1<<Sek_T ;einmal in der Sekunde geht das Flag Sek_T
       inc Sekunde         ;auf True
       cpi Sekunde,10
       brne sek
       clr Sekunde
       sbr Status,1<<Sek_T

sek:   in temp,In_P          ;Zustand der Tasten laden und im Status
       ldi thelp,(1<<Aus)    ;ablegen. Das Signal der Aus-Taste wird
       eor temp,thelp       ;vorher invertiert.
       cbr temp,~((1<<EinTa)|(1<<AusTa)) ;Wenn Aus gedrückt ist,
       mov thelp,temp       ;ist das Aus-Bit im Status-Register True
       mov temp,Status
       cbr temp,(1<<EinTa)|(1<<AusTa)
       or temp,thelp
       mov Status,temp

       eor temp,St_old      ;bei einer positiven Flanke wird das ent-
       and temp,Status      ;sprechende Bit im Flanken-Register
       mov FM_Reg,temp      ;gespeichert.
       mov St_old,Status    ;neue wird das alte werden

```

```

;Aktionen beim Drücken der Ein-Taste
    sbrs Status,Ein      ;Ein-Taste gedrückt?
    rjmp Ein_2           ;Sprung wenn nicht
    sbrc Status,Bel_An   ;Beleuchtung aus?
    rjmp Ein_1           ;Sprung, wenn Licht schon an
    sbr Status,1<<Bel_An;Beleuchtung aktivieren
    rjmp Aus_1

Ein_1:  clr Bel_TOFF      ;Nach jedem Druck startet TOFF neu
        inc Ein_TON       ;Einschaltverzögerung weiterschalten
        ldi temp,t_Ein    ;Ist die Zeit erreicht?
        sub temp,Ein_TON
        brcc Aus_1        ;Sprung, wenn Zeit noch nicht um
        sbi Out_P,Relais   ;sonst Ausgang aktivieren
        clr Bel_TOFF      ;damit der Nachlauf immer gleich ist.
        rjmp Aus_1

Ein_2:  clr Ein_TON

;Aktionen beim Drücken der Aus-Taste
Aus_1:  sbrs Status,Aus   ;Aus-Taste gedrückt?
        rjmp Bel_1        ;Sprung wenn nicht
        in temp,Out_P     ;Ist der Ausgang aktiv?
        tst temp
        breq Aus_2        ;Sprung wenn nicht
        cbi Out_P,Relais   ;Abschalten des Ausgangs
        rjmp Bel_1

Aus_2:  sbrc FM_Reg,Aus   ;Beleuchtung abschalten wenn Flanke
        cbr Status,1<<Bel_An;Beleuchtung aus

;Beleuchtung Ein- und Ausschalten und prüfen, ob sie noch an sein soll
Bel_1:  sbrs Status,Bel_An
        rjmp Bel_2
        out OCR0A,Bel_St  ;Beleuchtung mit dem neusten Helligkeits-
                           ;wert versorgen
        sbic Out_P,Relais  ;ist der Ausgang aktiv? Solang er aktiv
        rjmp Bel_3        ;ist, bleibt die Lampe an.

        sbrs Status,Sek_T ;Das erhöhen von Bel_OFF erfolgt nur einmal
        rjmp Bel_4        ;in der Sekunde
        inc Bel_TOFF      ;Ausschaltverzögerung weiterschalten
        ldi temp,t_Bel    ;Ist die Zeit erreicht?
        sub temp,Bel_TOFF
        brcc Bel_4        ;Sprung, wenn Zeit noch nicht um

Bel_2:  clr temp
        out OCR0A,temp    ;Beleuchtung ausschalten
        cbr Status,1<<Bel_An
Bel_3:  clr Bel_TOFF
Bel_4:  rjmp Start

```

```
;Die Werte, die in das OCR0A-Register geladen werden und so den Tastgard  
;des PWM-Signals bestimmen. Sie folgen einer Exponential-Funktion, damit  
;der Drehwinkel- und Helligkeitszunahme besser aufeinander abgestimmt  
;sind.  
;Für jeden der 1024 möglichen Messwerte gibt es einen PWM-Wert  
;In der Nähe des Links- und Rechtsanschlag des Potis sind alle Werte 0  
;oder 255. So gibt es klare Aus- und Max-Einstellungen.
```

PWM_Wert:

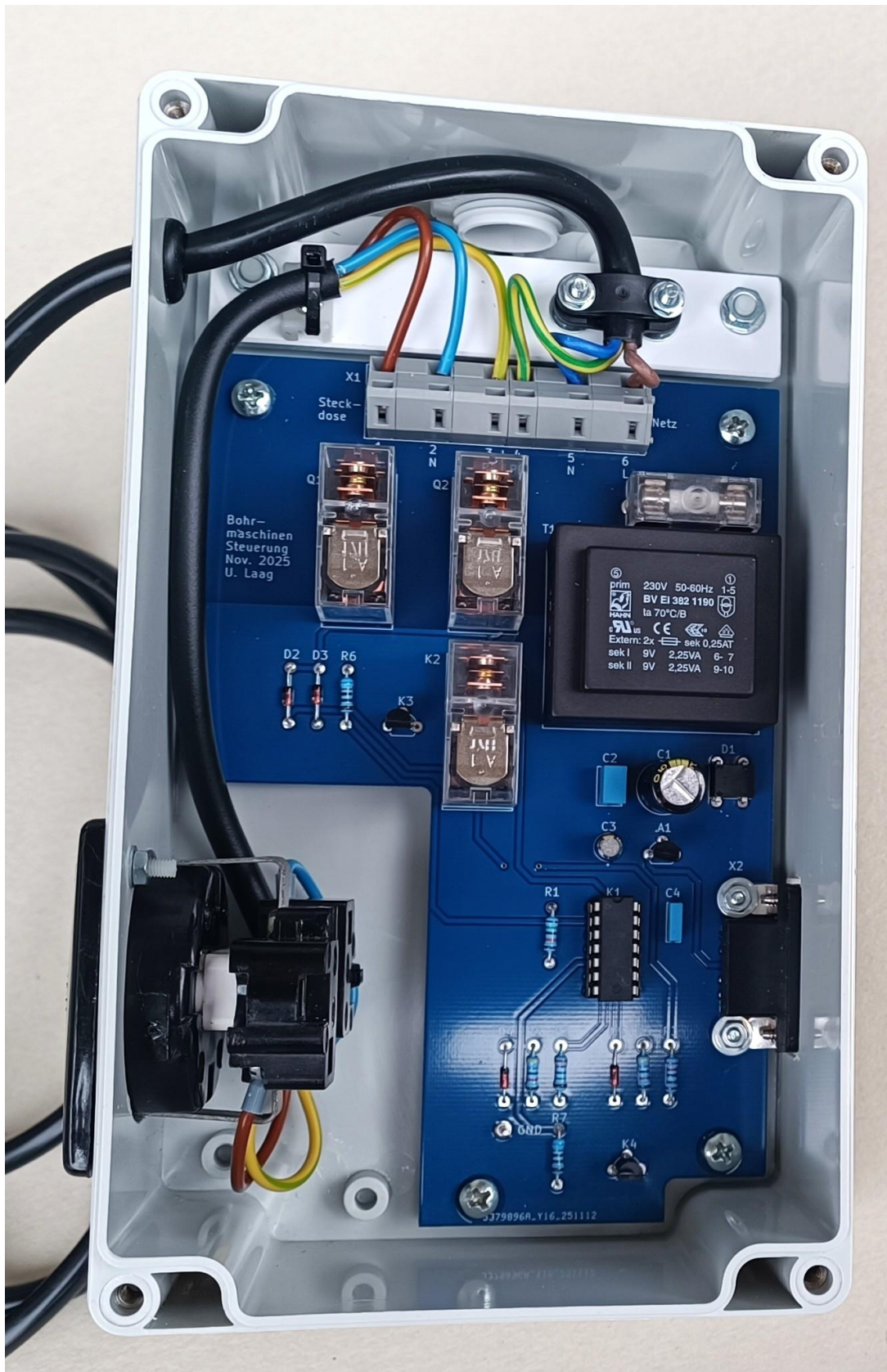
```
.db 0, 0, 0, 0, 0, 0, 0, 0  
.db 0, 0, 0, 0, 0, 0, 0, 0  
.db 25, 25, 25, 25, 25, 25, 25, 25  
.db 25, 26, 26, 26, 26, 26, 26, 26  
.db 26, 26, 26, 26, 26, 26, 26, 26  
.db 26, 27, 27, 27, 27, 27, 27, 27  
.db 27, 27, 27, 27, 27, 27, 27, 27  
.db 27, 28, 28, 28, 28, 28, 28, 28  
.db 28, 28, 28, 28, 28, 28, 28, 28  
.db 29, 29, 29, 29, 29, 29, 29, 29  
.db 29, 29, 29, 29, 29, 29, 29, 30  
.db 30, 30, 30, 30, 30, 30, 30, 30  
.db 30, 30, 30, 30, 30, 31, 31, 31  
.db 31, 31, 31, 31, 31, 31, 31, 31  
.db 31, 31, 31, 32, 32, 32, 32, 32  
.db 32, 32, 32, 32, 32, 32, 32, 32  
.db 32, 33, 33, 33, 33, 33, 33, 33  
.db 33, 33, 33, 33, 33, 34, 34, 34  
.db 34, 34, 34, 34, 34, 34, 34, 34  
.db 34, 34, 35, 35, 35, 35, 35, 35
```

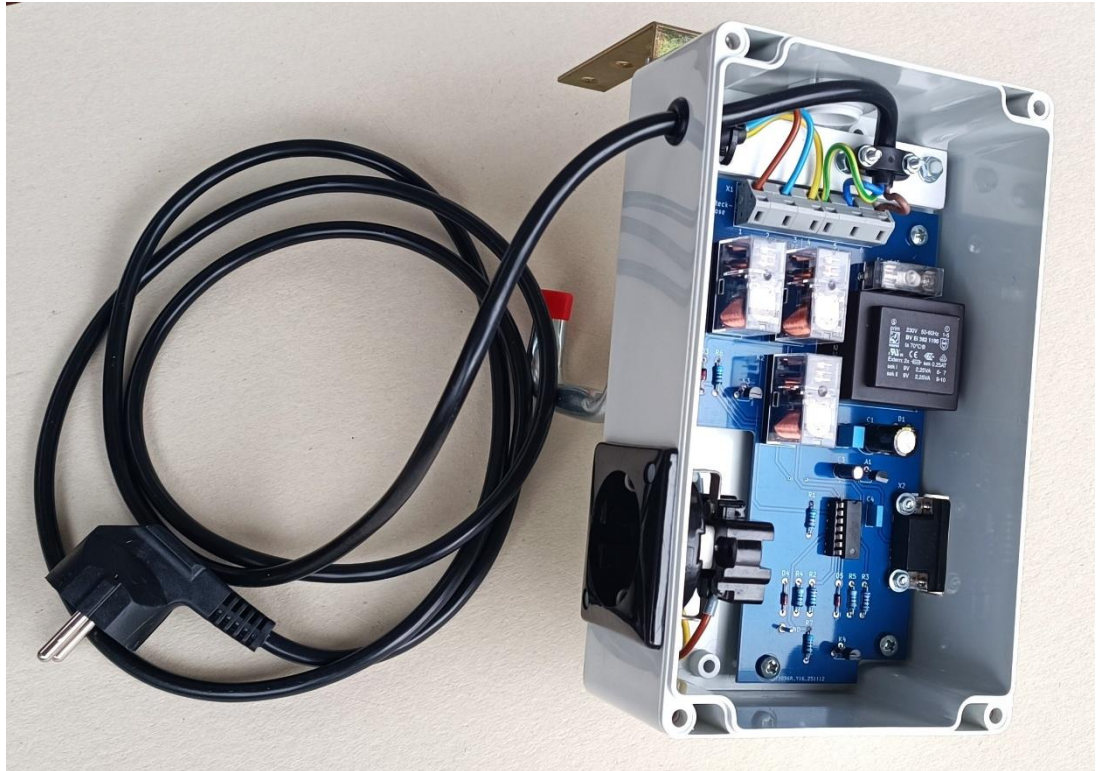
```
; . . . und viele Werte dazwischen . . .
```

```
.db 200, 201, 201, 202, 202, 202, 203, 203  
.db 204, 204, 205, 205, 206, 206, 207, 207  
.db 208, 208, 209, 209, 210, 210, 211, 211  
.db 212, 212, 213, 213, 214, 214, 215, 215  
.db 216, 216, 217, 217, 218, 218, 219, 219  
.db 220, 220, 221, 221, 222, 222, 223, 223  
.db 224, 224, 225, 225, 226, 227, 227, 228  
.db 228, 229, 229, 230, 230, 231, 231, 232  
.db 232, 233, 234, 234, 235, 235, 236, 236  
.db 237, 237, 238, 239, 239, 240, 240, 241  
.db 241, 242, 242, 243, 244, 244, 245, 245  
.db 246, 246, 247, 248, 248, 249, 249, 250  
.db 251, 251, 252, 252, 253, 254, 254, 255  
.db 255, 255, 255, 255, 255, 255, 255, 255  
.db 255, 255, 255, 255, 255, 255, 255, 255
```

5 Bilder

5.1 Die Steuerung



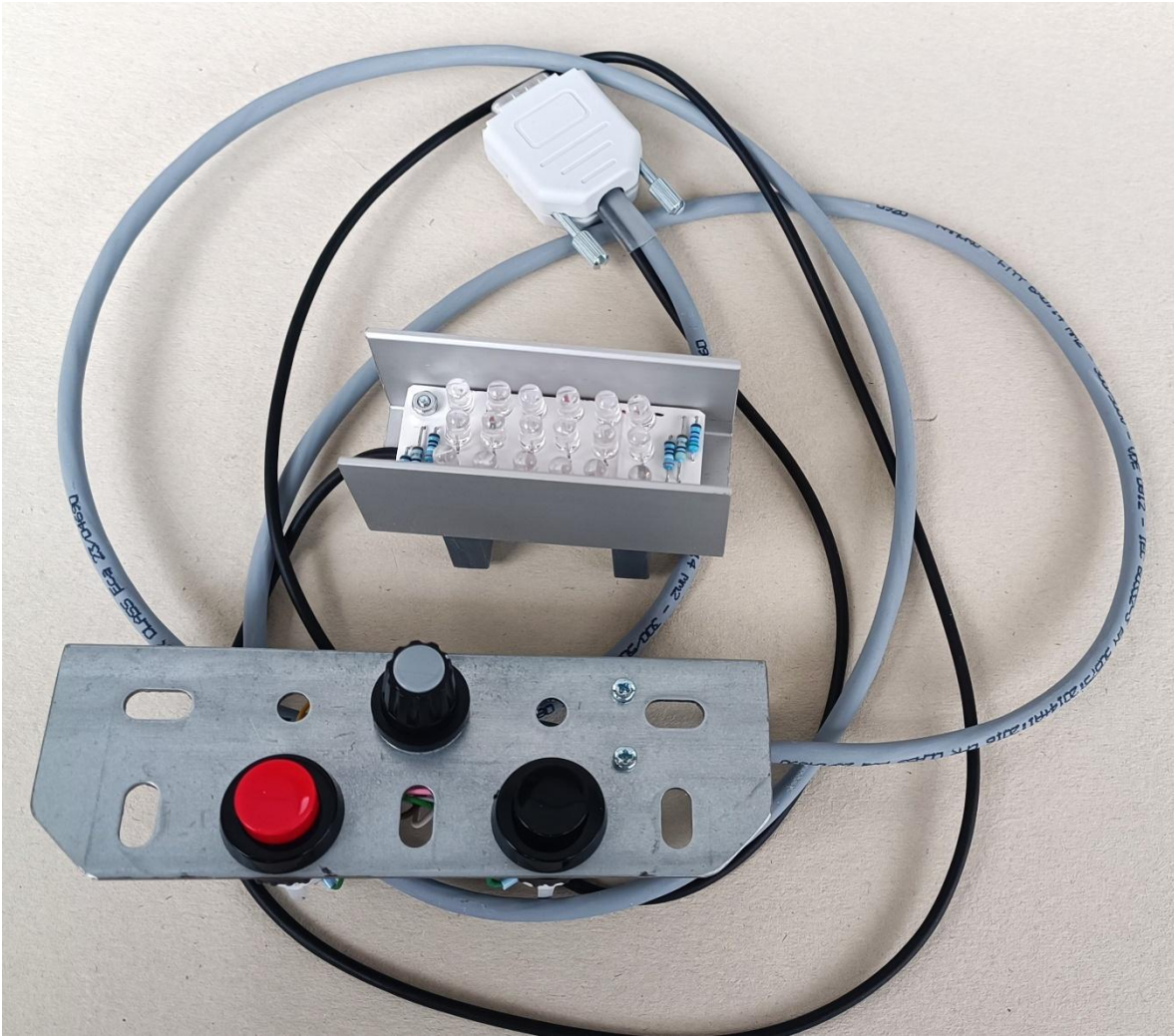


Der Hacken mit der roten Kappe auf der Rückseite des Steuergehäuses, dient zum Halten der aufgewickelten Bohrmaschinenleitung.

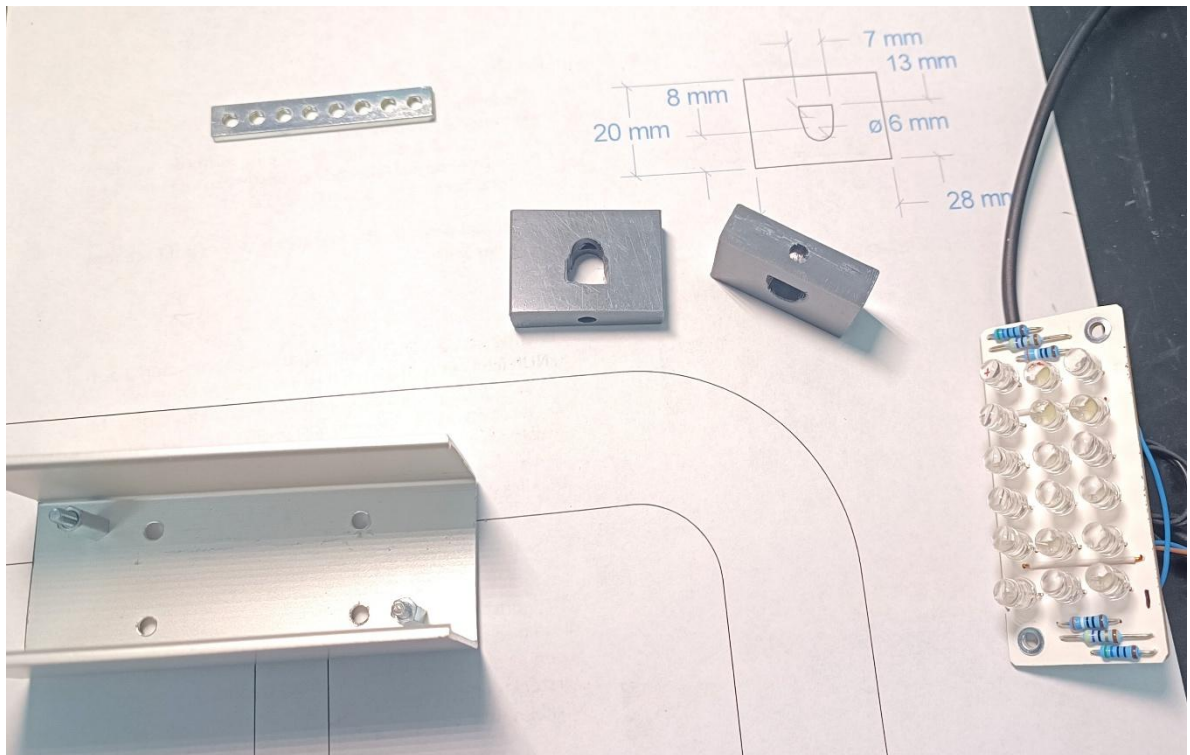


Die D-Sub-Buchse für den Anschlussstecker von Bedieneinheit und Bohrfeldbeleuchtung.

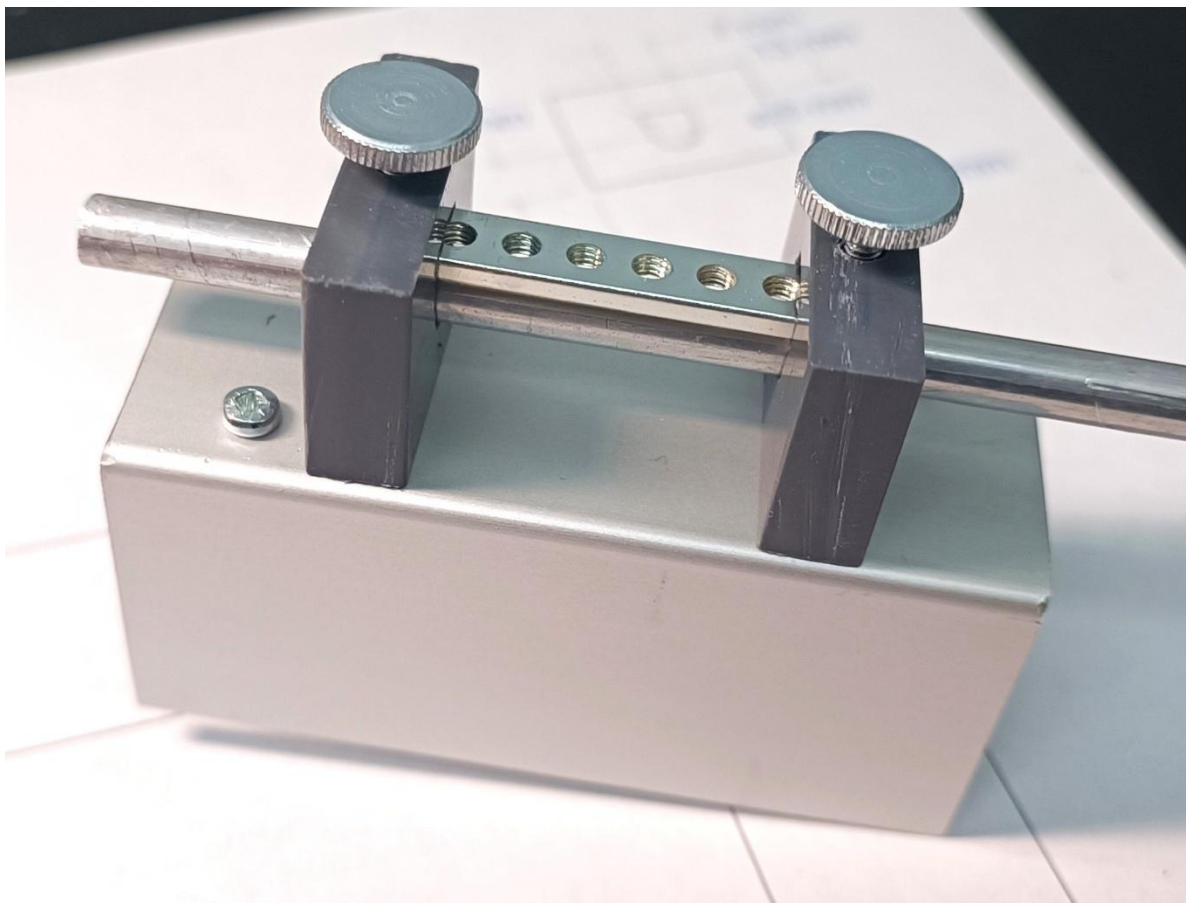
5.2 Die Bedieneinheit

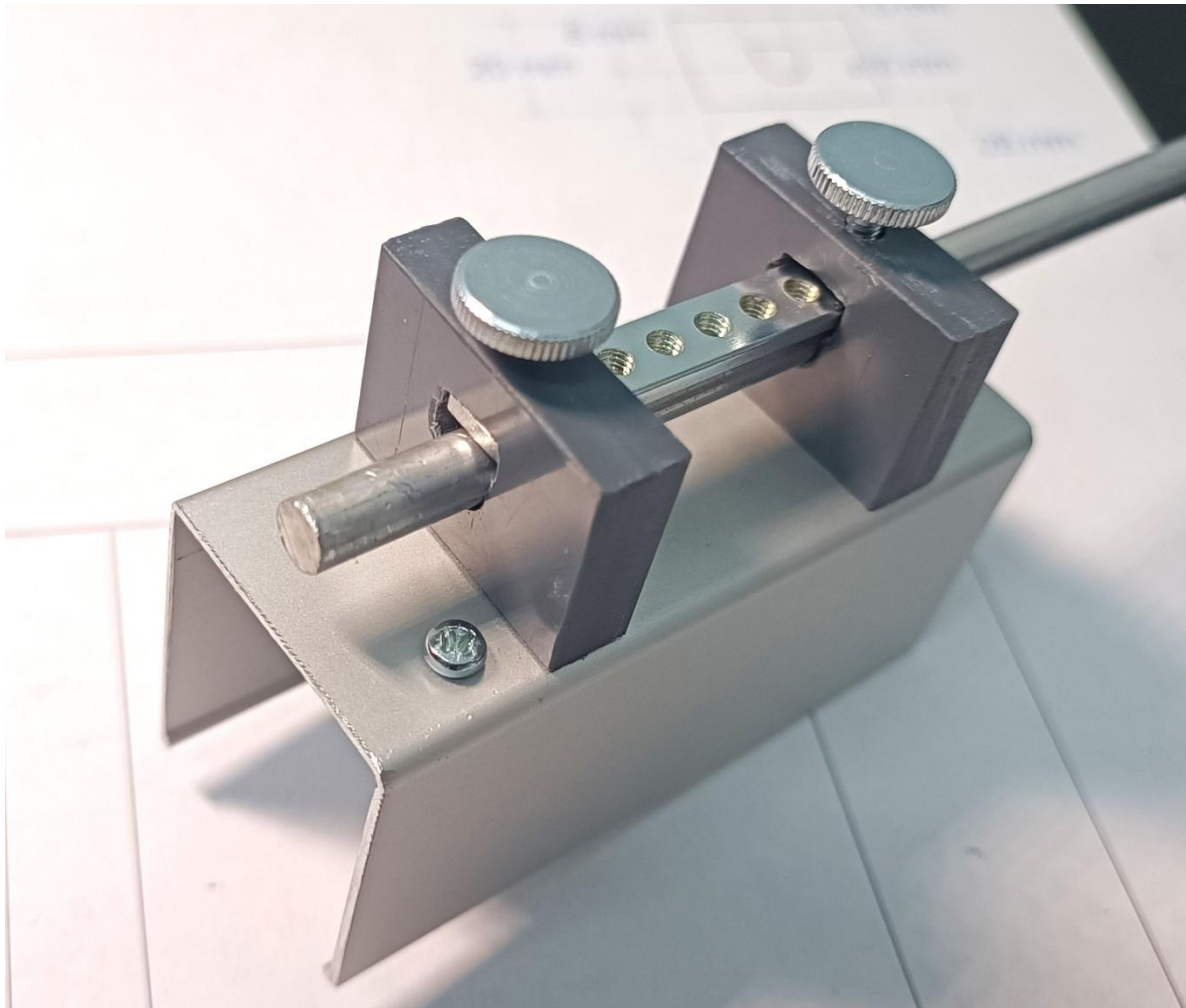


5.3 Die Bohrfeldbeleuchtung



Der Gehäusebau und rechts die Platine mit den LEDs.

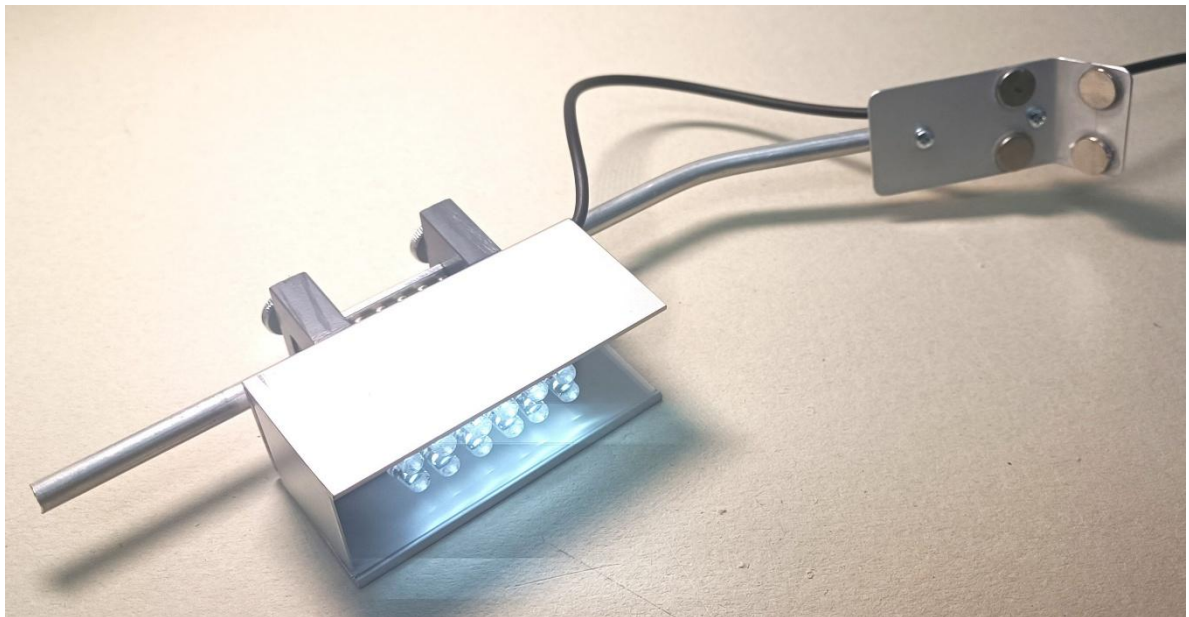




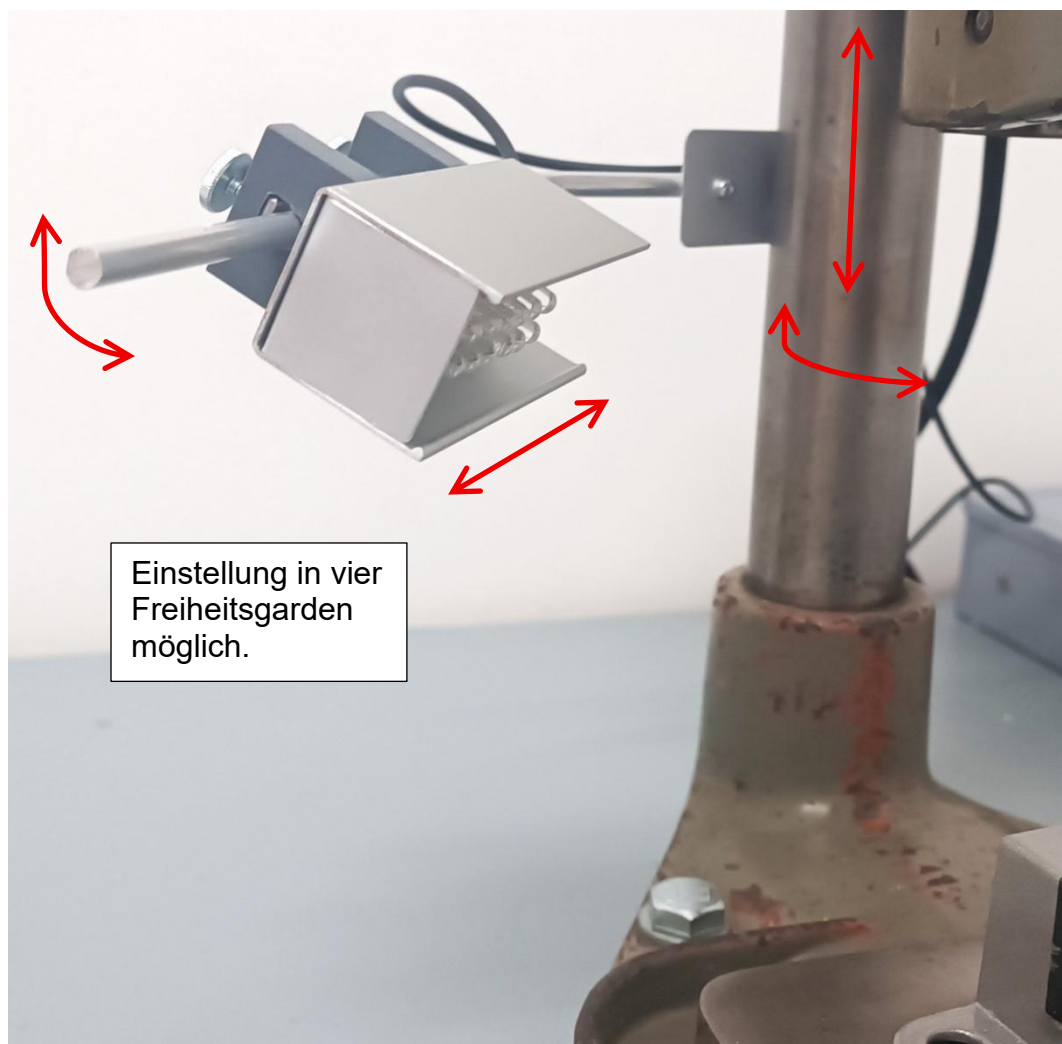
Durch die Rändelschrauben kann man leicht die Fixierung zwischen Lampe und Alustange lösen. Später ist die linke Seite noch durch einen Aluwinkel geschlossen worden.



Noch mal die Rasterplatine, abgeklebt mit weißer Folie.



Hier ist der Blechwinkel auf der linken Seite des LED-Gehäuses zu sehen. Er verhindert einen Lichtaustritt in Richtung des Bedieners.
Auf der gewinkelten Platte rechts, sind 4 Neodym-Magnete festgeklebt. Damit wird die Lampe an der Säule des Bohrmaschinenständers festgemacht.



5.4 Die Praxis

