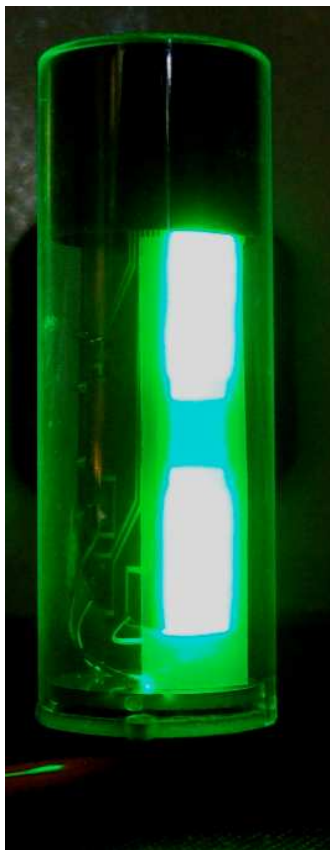


Emulation der EM87



Inhalt

Abbildungsverzeichnis	2
Historie dieser Doku.....	2
Vorwort	3
Display.....	4
Schaltung.....	5
Bauteilkosten	6
Software	7
Aufbau.....	9
Einbau in ein Gerät.....	13
Serienheizung.....	13
Triodenfunktion	13
Leuchtschirmspannung	13
Gitterstrom	13
Leuchtschirmstrom	13
Verwendung als EM84	13
HF-Störungen.....	13

Abbildungsverzeichnis

Abbildung 1: EM80-Emulator	3
Abbildung 2: EM80-Emulator-Schirmbild	3
Abbildung 3: EM87 von vorne.....	4
Abbildung 4: EM87 von der Seite.....	4
Abbildung 5: Prinzip- und Detailschaltbild des EM87-Emulators	5
Abbildung 6: 'C'-Source des EM87-Emulators	8
Abbildung 7: EM87-Platinen	9
Abbildung 8: Elektronik-Fotos 1.....	10
Abbildung 9: Elektronik-Fotos 2.....	11
Abbildung 10: Elektronik-Fotos 3.....	12
Abbildung 11: Einbau	14

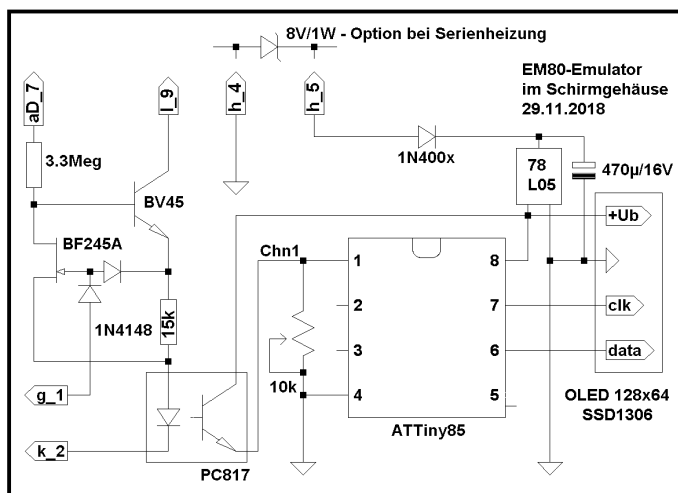
Historie dieser Doku

- 15.08.2019** Initialversion, Labormuster
- 23.08.2019** Prototyp-Fertigstellung
- 04.09.2019** Einbau des Emulators in ein Radio, Veröffentlichung der Doku
- 05.09.2019** Einbau-Kapitel erweitert

Vorwort

Die Ende 2018 von Peter (aka „laurel1“) und Wolfgang (aka „Rumgucker“) realisierte EM80-Emulation verwendete ein monochromes OLED-Display mit integriertem Spannungswandler, das von einem Controller seriell angesteuert wurde. Die Stromversorgung des Controllers und des Displays erfolgte aus der Röhrenheizung.

Die Steuerspannung des EM80-Emulators wurde in einem galvanisch getrennten Schaltungsteil gemessen und per Optokoppler an den Controller mit Display übertragen. Der Strom für die Elektrodennachbildung wurde von der Anoden- und Leuchtschirmspannung geliefert. So konnte die Trennung von Elektroden- und Heizkreis eingehalten werden.



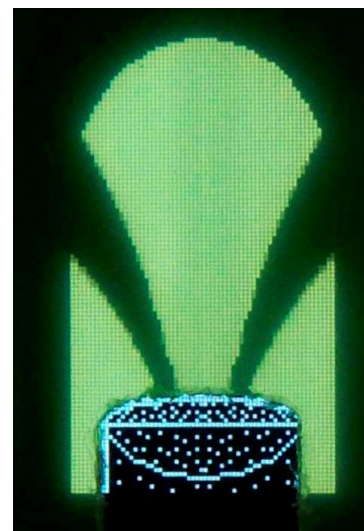
Nachteilig war der Schaltungsaufwand durch die zwei Schaltungsteile und die Metallabschirmung der gesamten Elektronik, um die bei der seriellen Ansteuerung des Displays entstehenden HF-Störungen zu reduzieren.

Abbildung 1: EM80-Emulator

Vorteilhaft war die freie Darstellung des Grafikdisplays. Es konnte sogar die Wölbung des EM80-Leuchtschirms nachgeahmt werden. Die grüne Färbung entstand durch eine passende Farbfolie.

Eine sehr detailreiche 60-seitige Doku mit Hard- und Software wurde in mehreren Foren verteilt.

Abbildung 2: EM80-Emulator-Schirmbild



Volker (aka „Radio-Volker“) benötigte jedoch eine EM87 und bildete zusammen mit Wolfgang in 2019 das EM87-Entwicklungsteam.

Display

Bei der EM87-Emulation wird ein Grafik-Displaymodul mit einer durch die Röhre vorgegebenen Leuchtbandhöhe von 34 mm und einer Einbaubreite von unter 20 mm (Röhreninnendurchmesser) benötigt.

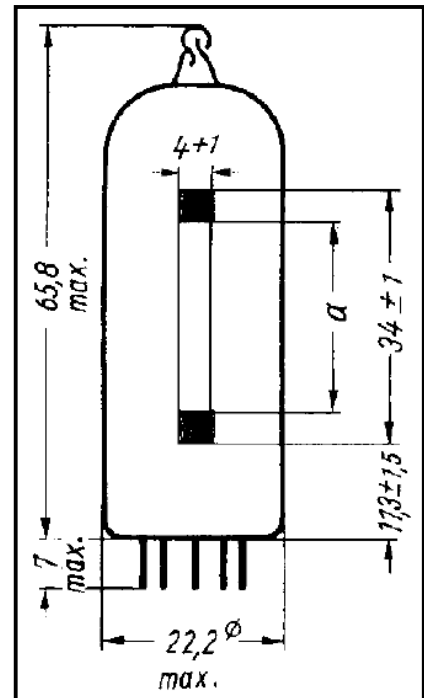


Abbildung 3: EM87 von vorne

Je schmaler das Modul, desto mehr gestattet es das originalgetreue Heranrücken der Leuchtfläche an die vordere Röhreninnenwand.

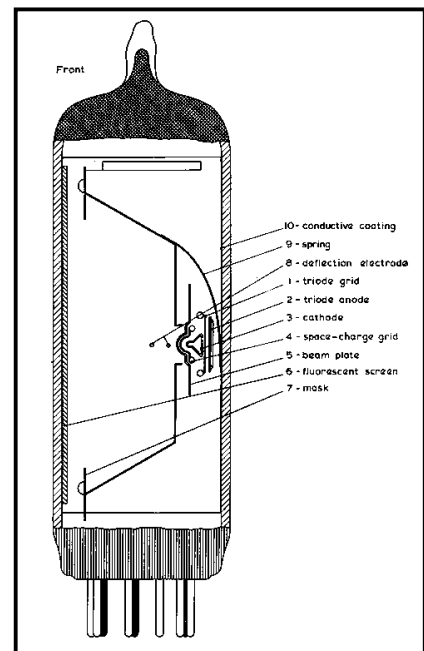


Abbildung 4: EM87 von der Seite

Ein derartiges schmales Display war nicht beschaffbar. Es musste eine eigene Konstruktion erdacht werden.

Schaltung

Die EM87 benötigt einen Leuchtschirmstrom von 2mA. Dieser Strom reicht aus, um eine Reihenschaltung aus Widerständen, 17 LEDs und eines Controllers zu versorgen. Treibertransistoren überbrücken die jeweils nicht leuchtenden LEDs. Ein Komparator im Controller misst die Gitterspannung und steuert die Treibertransistoren. Spannungswandler, ADC, Pegelwandler und Optokoppler werden nicht benötigt

Die 17 Anreih-LEDs (5 x 2mm) bilden das 5 mm breite und 34 mm hohe Leuchtband. Die äußeren 6 LEDs leuchten dauerhaft, weil sich auch der originale EM87-Schatten nur bis 22 mm aufspreizen lässt. Die restlichen 11 LEDs teilen sich in zwei spiegelbildliche Hälften. Es genügen somit 6 Treiber, die mit 4 Signalen gesteuert werden. Eine PWM erzeugt gleitende Helligkeitsübergänge, wodurch der Eindruck einer feinen Auflösung entsteht.

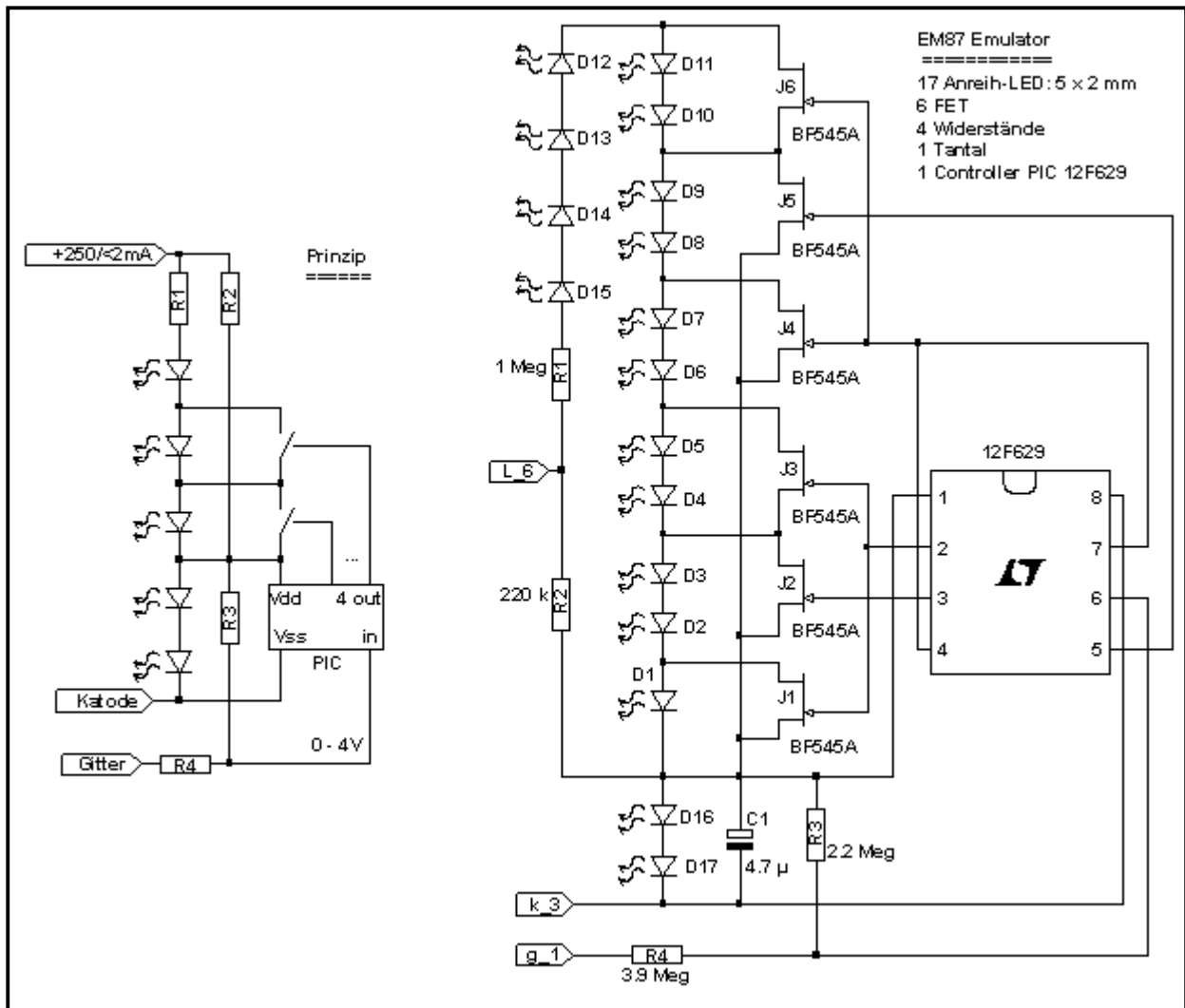


Abbildung 5: Prinzip- und Detailschaltbild des EM87-Emulators

Mit R1 wird die Helligkeit der LED-Kette bestimmt. Der geringe Strom moderner LEDs (0,25 mA) reicht nicht, um allein den Controller zu versorgen. Daher bezieht der Controller zusätzlichen Versorgungsstrom über R2.

Die Spannung über der LED-Kette ist abhängig von der jeweiligen Ansteuerung der FETs und kann von 12 bis 34 Volt schwanken. Trotzdem sind die Ströme durch R1, R2, LEDs und den Controller nahezu konstant

Die Controller-Versorgung wird von den zwei äußersten LEDs (D16, D17) stabilisiert, deren Leuchten eine ausreichende Controllerversorgung anzeigt.

Die Schaltung arbeitet mit einer Leuchtschirmspannung zwischen +100 und +300V. Eine negative Leuchtschirmspannung zerstört die Elektronik.

Es wird nur die Funktion der drei Elektroden Leuchtschirm (Pin 6), Triodengitter (Pin 1) und Katode (Pim 3) emuliert. Alle anderen Anschlüsse einer EM87 sind unbeschaltet.

Bauteilkosten

inkl. Teile-Versand, Steuern und Zuschlägen. Stand August 2019.

6 Stück NXP BF545A	€ 1,80
17 Stück China LED	€ 0,60
1 Stück PIC 12F629	€ 1,00
1 Stück Zentrifugenröhrchen, Außendurchmesser=22mm, Außenhöhe=62mm	€ 0,80
4 Widerstände, 1 Tantal, Platinen, DIL8-Fassung für PIC, 10 Steckstifte	€ 1,00

Software

Der 12F629 besitzt keinen ADC. Der Komparator wird per Referenzspannung als 4-Bit-ADC missbraucht, dessen Wandlungsergebnis mit der Schattenwurfkennlinie und der PWM-Tabelle verrechnet wird, bevor die „FOREVER“-Schleife erneut startet.

```
#include "allpic.h"

#pragma config = 0b000111000100 // hex-datei: fuses des 12F629
                                // BODEN

#include "timeloop.h"

#define _FET_46      0 // GPIO-Bitposition
#define _UIN        1 // GPIO-Bitposition
#define _FET_5      2 // GPIO-Bitposition
#define _GP3        3 // GPIO-Bitposition
#define _FET_2      4 // GPIO-Bitposition
#define _FET_13     5 // GPIO-Bitposition

// Schatten wird in 7 vollen Stufen gesteuert
#define MM_0        0
#define MM_2        _BV(_FET_13)
#define MM_6        _BV(_FET_2)
#define MM_10       _BV(_FET_13) | _BV(_FET_2)
#define MM_14       _BV(_FET_46)
#define MM_18       _BV(_FET_5)
#define MM_22       _BV(_FET_46) | _BV(_FET_5)

// 23 Stufen entstehen durch PWM
const uns8 SCHATTEN[] = {
    MM_0,MM_0,MM_0,MM_0, //mm_0
    MM_2,MM_0,MM_2,MM_0, //mm_1
    MM_2,MM_2,MM_2,MM_2, //mm_2
    MM_6,MM_2,MM_2,MM_2, //mm_3
    MM_6,MM_2,MM_6,MM_2, //mm_4
    MM_6,MM_6,MM_6,MM_2, //mm_5
    MM_6,MM_6,MM_6,MM_6, //mm_6
    MM_10,MM_6,MM_6,MM_6, //mm_7
    MM_10,MM_6,MM_10,MM_6, //mm_8
    MM_10,MM_10,MM_10,MM_6, //mm_9
    MM_10,MM_10,MM_10,MM_10, //mm_10
    MM_14,MM_10,MM_10,MM_10, //mm_11
    MM_14,MM_10,MM_14,MM_10, //mm_12
    MM_14,MM_14,MM_14,MM_10, //mm_13
    MM_14,MM_14,MM_14,MM_14, //mm_14
    MM_18,MM_14,MM_14,MM_14, //mm_15
    MM_18,MM_14,MM_18,MM_14, //mm_16
    MM_18,MM_18,MM_18,MM_14, //mm_17
    MM_18,MM_18,MM_18,MM_18, //mm_18
    MM_22,MM_18,MM_18,MM_18, //mm_19
    MM_22,MM_18,MM_22,MM_18, //mm_20
    MM_22,MM_22,MM_22,MM_18, //mm_21
    MM_22,MM_22,MM_22,MM_22 //mm_22
};
```

```

const uns8 UG_KURVE[] = {          // cref -> SCHATTEN-Index
    0, 4, 4, 8, 12, 16, 20, 24, 28, 32, 36, 44, 52, 60, 72, 88};

#define USEC_1000    1000

void main(void)
{
    // ***** initialisierungen *****
    RP0 = TRUE;                // erstmal alle Spezialregister...
    #asm
        DW /*CALL*/ 0x2000 + /*ADRESSE*/ 0x03FF // oscal abholen
    #endasm
    OSCCAL = W;                // und Oszillorkalibrierung speichern

    OPTION = 0;                // global weak-pullup ein (je 20kOhm)
    WPU = 0;
    TRISIO = _BV(_UIN) | _BV(_GP3);    // eingänge

    RP0 = FALSE;                // normale register und ram
    CMCON = 0b00010100;        // komparator "100" und inverted

    timeloop_init();          // timer initialisieren

    uns8 cref = 0;            // Comp-ref-Spannung

    FOREVER {                  // ewige schleife
        RP0 = TRUE;            // Spezialregister...
        VRCON = 0b10100000 | cref; // referenzspannung setzen
        RP0 = FALSE;          // normale register

        usec(25);              // damit sich alles beruhigt
        if(COUT){              // 4-bit-"ADC"
            if(cref < 15) cref++; // aktuelle comparator-referenzspannung
        }
        else if(cref) cref--;

        uns8 schatten_index = UG_KURVE[cref]; // ref in schattenwurf umrechnen
        TMR1L = ~(USEC_1000 % 256);
        TMR1H = ~(USEC_1000 / 256);
        TMR1IF = FALSE;        // das int-flag verwenden
        do {                    // pwm gehörige zeit laufen lassen
            GPIO = SCHATTEN[schatten_index + 0];
            GPIO = SCHATTEN[schatten_index + 1];
            GPIO = SCHATTEN[schatten_index + 2];
            GPIO = SCHATTEN[schatten_index + 3];
        } while(!TMR1IF);      // obwohl ints ausmaskiert sind
    }
}

/* ENDE */

```

Abbildung 6: 'C'-Source des EM87-Emulators

Aufbau

In ein 62 x 22mm großes Plastikröhrchen werden vier - mit SMD- und Durchsteckbauteilen bestückte - Platinen eingeschoben. Zwei runde Platinen tragen die Steckstifte und bilden den Röhrenboden. Die mittlere Platine stellt das eigentliche Display mit den 6 Treiber-Transistoren dar. Die rechte Platine trägt den Controller. Controller und Display werden zu einem Sandwich zusammengesetzt. Die LEDs befinden sich zwischen den Platinen.

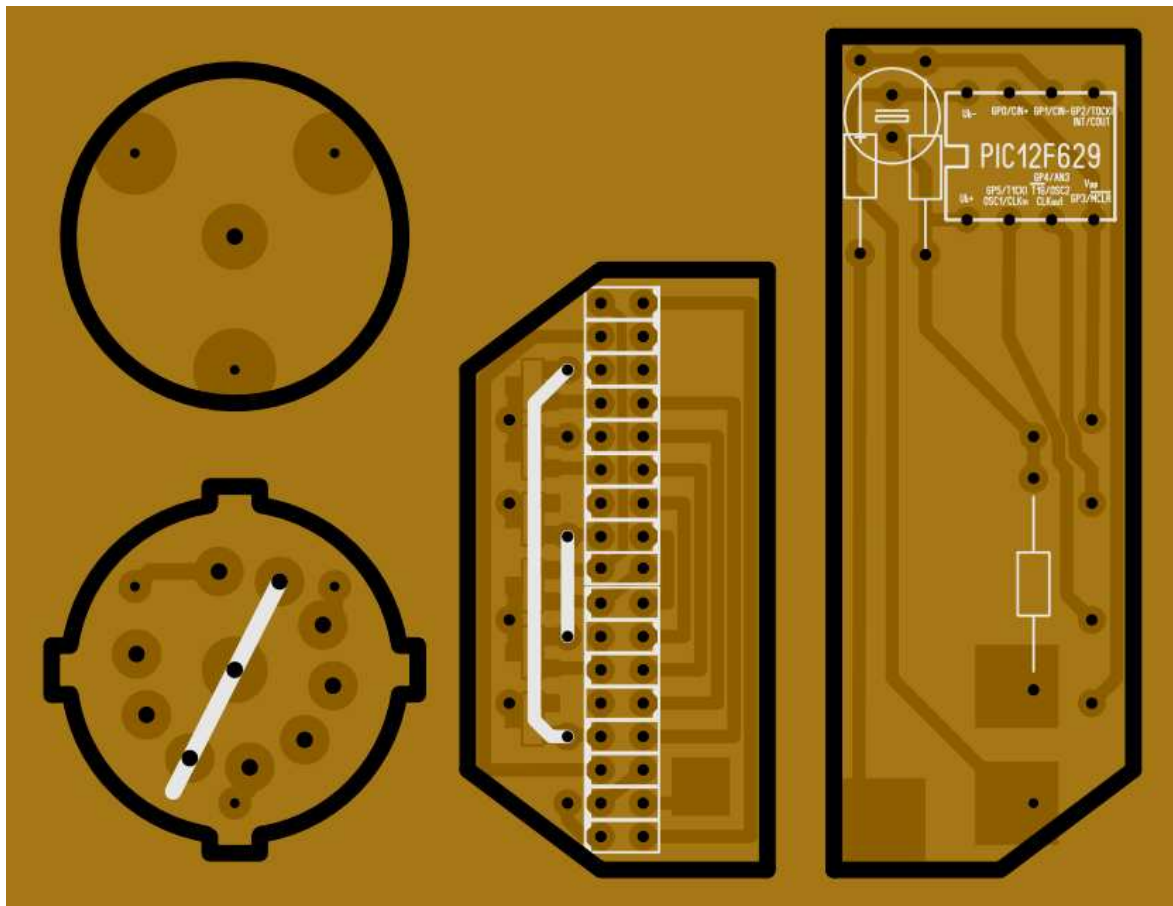


Abbildung 7: EM87-Platinen

Der Controller wird von einem im Plastikröhrchen montierten „Getter“ (ringförmiger Plastikstreifen) verdeckt. Ein schwebender Widerstand (R1) und drei Drähte bilden die Verbindung zwischen den runden Bodenplatinen und der Elektronik. Als „Leuchtschirm“ wurde ein matten Klebestreifen von innen eingeklebt. Insgesamt entsteht so ein dem Original ähnliches Aussehen.

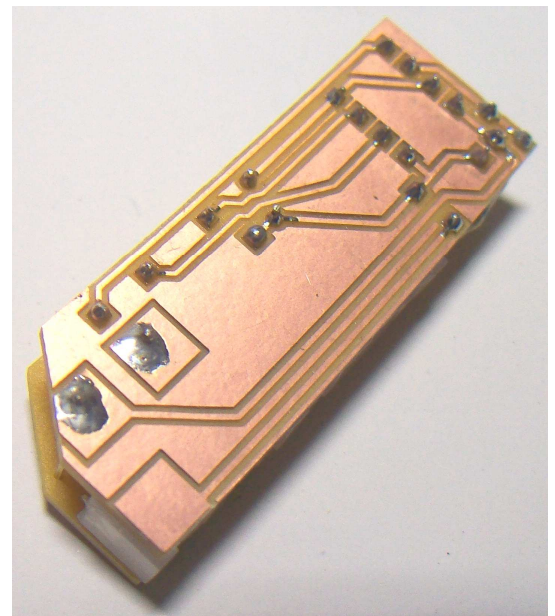
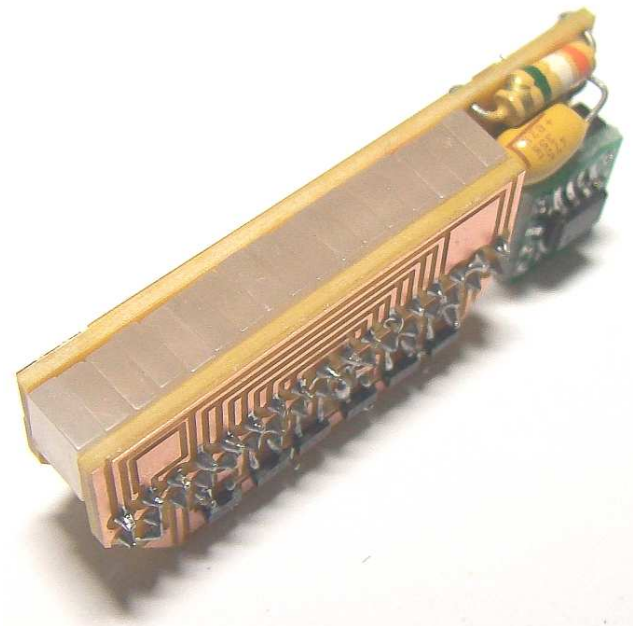
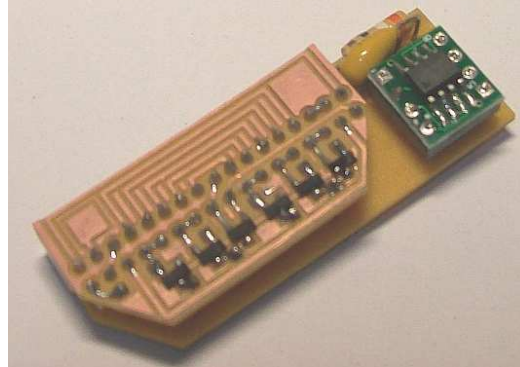
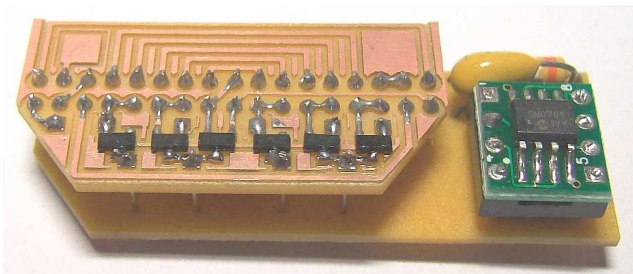
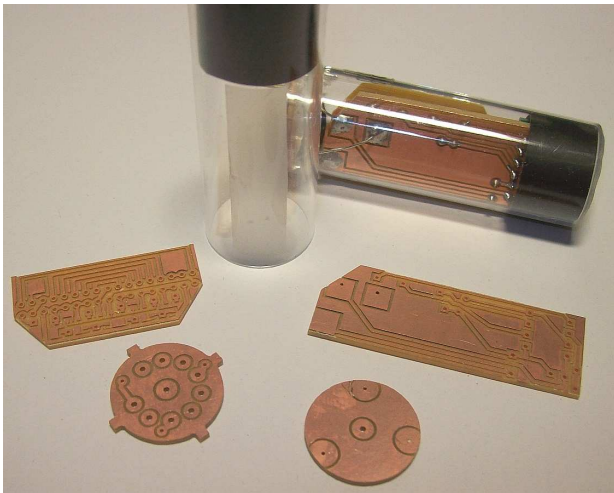


Abbildung 8: Elektronik-Fotos 1

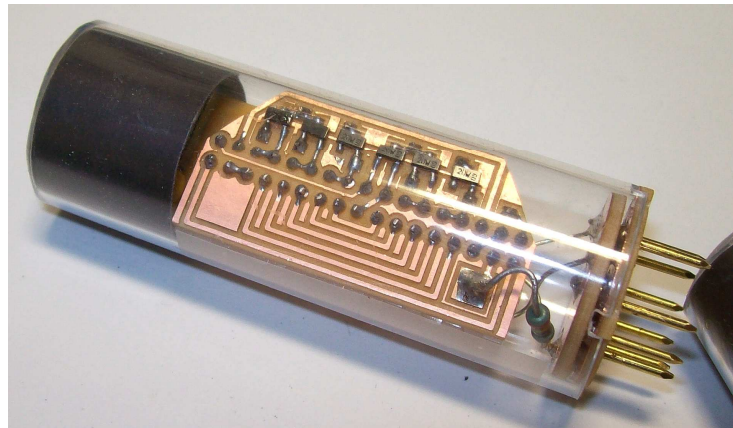
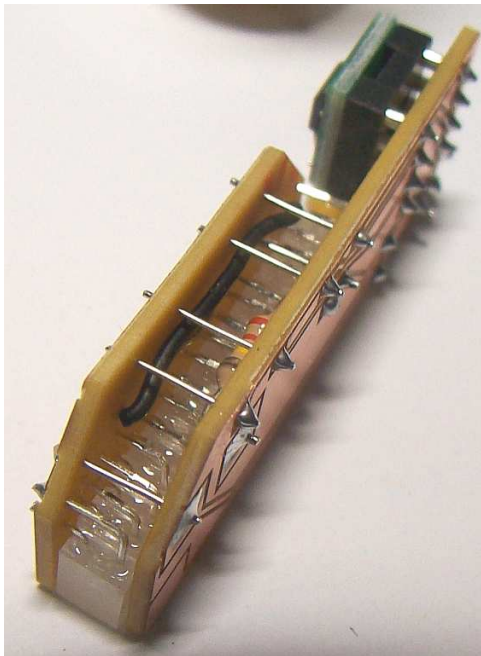


Abbildung 9: Elektronik-Fotos 2

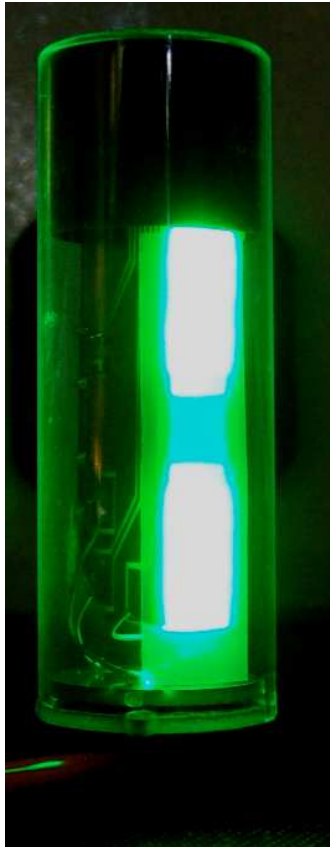
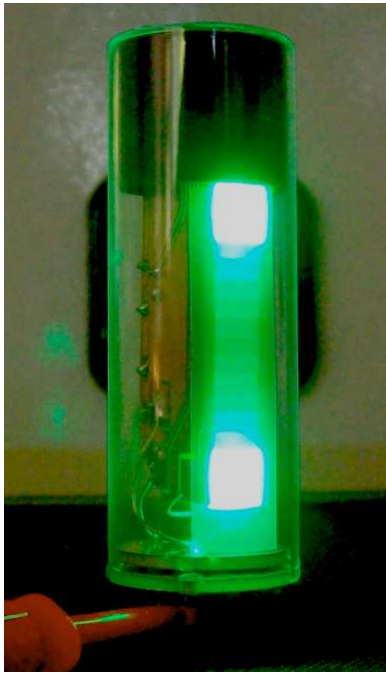


Abbildung 10: Elektronik-Fotos 3

Einbau in ein Gerät

Serienheizung

Der Heizer wird nicht emuliert und ist unbeschaltet. Er würde daher den Stromfluss in einem Serienheizkreis unterbrechen. Abhilfe: Lämpchen 6V/0.3A als Ersatzheizung verwenden.

Triodenfunktion

Die Anode der Triode wird nicht emuliert und ist unbeschaltet.

Leuchtschirmspannung

Der Emulator arbeitet mit Gleichspannung zwischen +100 und +300V. Die LED-Helligkeit ist proportional zur Spannungshöhe. Zur Reduzierung der LED-Helligkeit kann man in Reihe mit dem Leuchtschirmanschluss Pin 6 einen Widerstand schalten. Negative Leuchtschirmspannungen zerstören den Emulator und sind verboten. Pulsierende Gleichspannungen führen zu Funktionsstörungen.

Gitterstrom

Es fließen bis zu 2 μ A Strom aus dem Gitter heraus, die bei zu hochohmiger Ansteuerung zu Fehlanzeigen führen.

Leuchtschirmstrom

Der Leuchtschirmstrom beträgt - wie beim Original - unter 2 mA. Er ist jedoch konstant.

Verwendung als EM84

Es ist ein Widerstand 3.9 Meg in Reihe mit dem Gitteranschluss Pin 1 zu löten.

HF-Störungen

Der Emulator erzeugt HF-Störungen, die ggfls. mit einem Metallschirm reduziert werden müssen.

Der Einbau in ein Grundig Mandello klappte ohne Probleme.

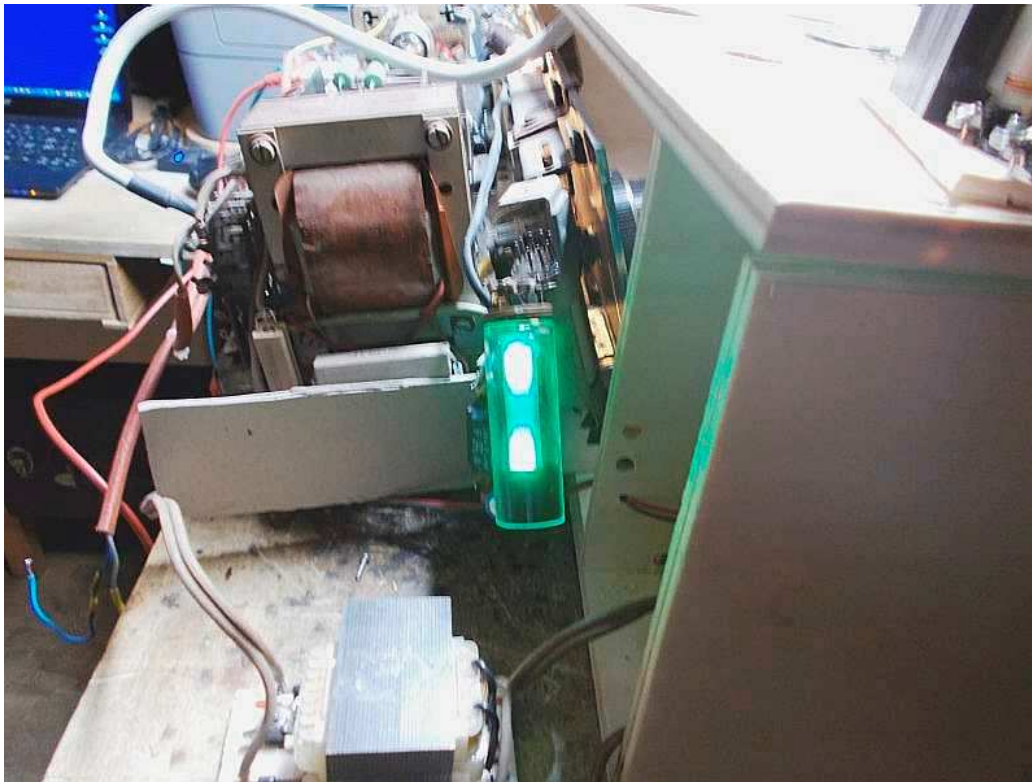


Abbildung 11: Einbau