

Platinenherstellung

Gliederung:

- 1) Problem
- 2) Entwurf der Platine
- 3) Herstellung & Teile
- 4) Bestückung
- 5) Test & Coding

Problem:

Bei der RGB-LED Matrix aus dem CT-Projekt des letzten Halbjahres gab es Kontaktprobleme, da alles von Hand auf sehr wenig Platz verlötet war. Auch nach langem Suchen konnte das Problem lediglich sporadisch durch Festdrücken der Lochrasterplatine gelöst werden. So blieb nur noch eine Lösung übrig: Übertragung des Projekts auf eine speziell angefertigte Platine.

Entwurf der Platine:

Nach einigem Versuchen mit Eagle erfolgte ein Wechsel auf KiCAD für das Design der Platine, da die Einschränkungen in der kostenlosen Version zu groß waren (nicht ernsthaft unter Gentoo lauffähig, onlinegebunden). Die Einarbeitung in KiCAD ist relativ einfach, der schematische Aufbau war schnell fertig und funktionsfähig. Jedoch fiel an dem ersten Aufbau ein Problem auf: die Stromversorgung. Die 2,5A über Micro-USB reichten für eine volle Beleuchtung in weiß nicht aus. Somit musste eine Stromversorgung über ein externes Netzteil her. Die Entscheidung fiel auf einen TPS51275B um 12V 4A auf 5V zu bucken. Jedoch wurde dadurch die Platine deutlich komplexer, weshalb das Autorouting-Tool freeroute zur Hilfe gezogen werden musste. Nach einigen Versuchen ergab sich schließlich ein akzeptables Ergebnis, welches produziert werden konnte.

Herstellung und Teile:

Aus Kostengründen wurden sämtliche Teile direkt aus China bestellt, was zwar billiger war und es vereinfachte, gleich ein ganzes Set zu beschaffen, jedoch fehlt insbesondere bei den DIP-16 Sockeln definitiv Qualität. Die Platinen wurden zu günstigen Preisen bei JLCPCB bestellt, welche die Platinen schnell (2 Tage) produzieren und für China auch schnell versenden (Theorie: 10-20 Tage; Praxis: 11 Tage).

Bestückung:

Die SMD Bauteile für die externe Stromversorgung ließen sich trotz 0603 Bauform sehr einfach und entspannt löten, allerdings nur mit ausreichend Flussmittel. Jedoch wurde eine falsche Library für die Mosfets gewählt, weshalb zuerst 11V statt 5V ausgegeben wurden, was zu dem Durchbrennen zweier LEDs führte. Nach kunstvollem Umbiegen spuckte der Buck-Converter erfolgreich 5,25V an dem stabilen 5V Ausgang aus. Beim nächsten Anschluss des Netzteils kamen erstaunlicherweise 5,9V bzw. etwas später 5,72V aus dem selben Ausgang, was definitiv zu viel Spannung für den Microcontroller ist. Außerdem funktionierte der selbst konfigurierte 5V Ausgang gar nicht. Dies könnte sich entweder auf eine billige Qualität bzw. ein Fake-Produkt des Chips zurückführen lassen oder darauf, dass irgendein Bauteil bei den ersten Tests einen Schlag abbekommen hat, weshalb weshalb zwei Optionen blieben: Entweder der Verwurf der externen Stromversorgung oder eine komplette Neuverlötung der SMD-Bauteile. Zum Glück

bekommt man aus China keine Einzelware, sondern alles mindestens im 5-er Pack. Nach der Verlötlung der zweiten Platine gab der Buck-Converter saubere 5V aus. Das weitere Löten sämtlicher Throughhole-Bauteile verlief auf der Hauptplatine und auf der Platine für die Siebensegmentanzeige reibungslos und zügig. Nach dem erneuten Löten der Header an den ESP32 (Erstes mal nach der Lieferung mit Headern bestückt, dann entlötet, um im ersten Testaufbau der Matrix zu verwenden, dann wieder entlötet und für den Steckplatz auf der Hauptplatine wieder mit Headern versorgt) stellte sich heraus, dass GPIO0 keinen Kontakt mehr zur Platine herstellen konnte, d.h. jetzt habe ich einen 99% funktionsfähigen ESP32, der aber nicht für den Einsatz auf der Platine taugt, da GPIO0 essentiell nötig (der Steuerpin für die Schieberegister der Farben der Hauptmatrix) ist. Daher: neuer ESP32 bestellt, allerdings wurde zuerst einer mit falschem Pinlayout (abweichend von dem passend für die Platine bzw. auch von den Bildern des Artikels), von demher entstanden weitere Verzögerungen im Zusammenbau, explizit dem Test der großen Matrix.

Test & Coding:

Zuerst wurden die Siebensegmentanzeigen getestet. Nachdem festgestellt wurde, dass eine Verbindung zu einem Segment fehlt erfolgte eine Fehlersuche: In der Schematik ist die Verbindung vorhanden im PCB-Design jedoch nicht mehr, KiCAD zeigt die fehlende Verbindung aber auch nicht an. Dies konnte einfach durch eine manuelle Überbrückung der Kontakte behoben werden.

Das nächste Problem war die NOT_RESET Leitung: Diese hängt am ESP32 an Pin 35, welcher sich aus bisher unbekannten Gründen nur als Eingang verwenden lässt, d.h. auch hier musste manuell von 3,3V auf GPIO35 überbrückt werden. Danach wollte der ESP32 im Sockel jedoch nicht mehr starten: Nach einiger Fehlersuche stellte sich heraus, dass alle Schieberegister auf der Hauptplatine falsch herum eingebaut wurden. Nach dem Wenden der 74HC595 funktionierte die Platine und die Stromversorgung einwandfrei.

Nach dem erfolgreichen Inbetriebnehmen der Siebensegmentanzeigen stellte sich ein Problem heraus: die Schieberegister und die Siebensegmentanzeigen der zweiten Platine ziehen so viel Strom, dass der Spannungswandler von 5V auf 3,3V auf dem ESP32 deutlich warm wird bzw. der Strom des sekundären 5V Outputs der externen Stromversorgung nicht für den Betrieb der Siebensegmentanzeigen taugt, d.h. einen Brownout des ESP32 auslöst und einen parallelbetrieb der beiden Netzteile erforderlich macht.

Die kleine Matrix ließ sich schnell zusammenbauen und bereitete keine nennenswerten Probleme. Da die Algorithmen der beiden Matrizen sehr ähnlich funktionieren sollte auch die große Matrix kaum Probleme bereiten. Im weiteren Casedesign (ein halbfertiges Case vom letzten Halbjahr war schon vorhanden) tauchte ein weiteres Problem auf: mit allen Flachbandanschlüssen ist das Case zu dünn, d.h. es ist nicht Platz, um das Case liegend zu verwenden und in der Deckplatte unterhalb der kleinen Matrix wird es eine unschöne Aussparung für den Connector geben müssen, trotz dass dort eine deutlich dünnere Plexiglasplatte statt dicken Holz eingesetzt wird.

In der großen Matrix ergaben sich einige Probleme mit LED's und Kontakten, welche sich aber überwiegend einfach lösen ließen.

Auch beim restlichen Zusammenbau ergaben sich keine nennenswerten Probleme. Da es keine offizielle 4-Gewinnt-Musik gibt, wurde einfach „Shooting Stars“ gewählt.

Ein bemerkenswerter Bug wurde in der Coding Phase gefunden:

```
for(int i=19;i>9;i++) for(int j=0;j<=6;j++)  
sorgt dafür, dass j größer als 6 werden kann.
```