

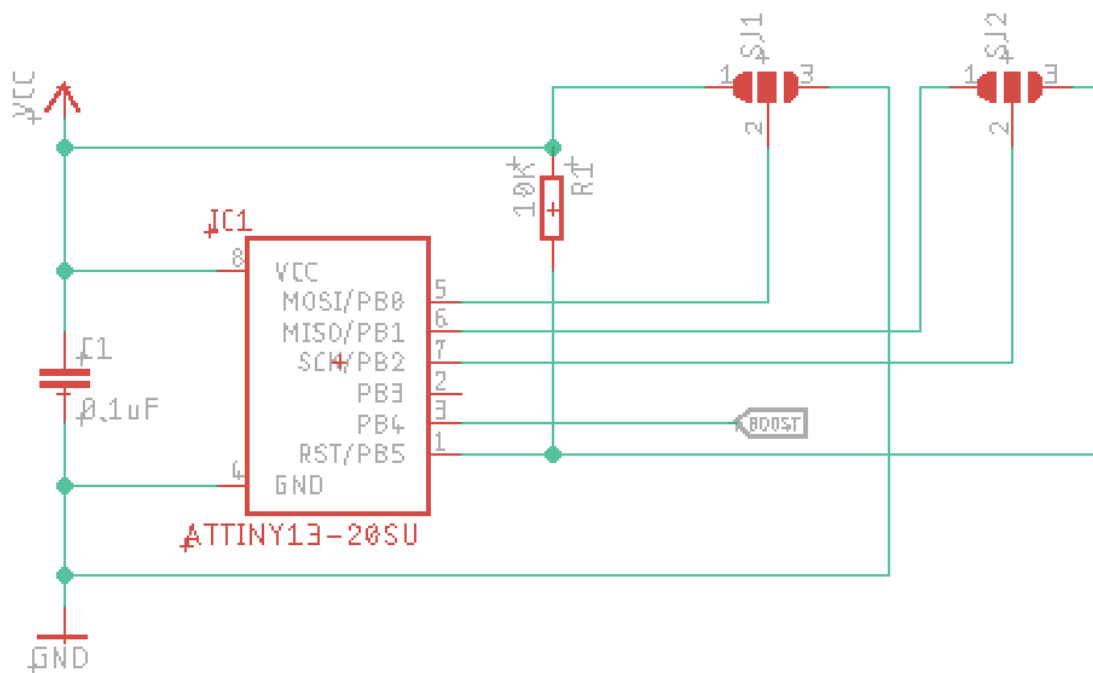
F1B-Flasher als Open Source-Projekt

Anfang 2000er habe ich das erste Mal ein Elektronik Timer bei Allard (van Wallen) gesehen. Er hatte mit dem „Basic Stamp“ was gebaut, Betriebsspannung war 9V Batterie gewesen. Also kein Problem für die F1A Modelle. Da hatte ich schon angefangen zu überlegen, wie ich die Elektronik Technologie an F1B anwenden kann. Leider waren die Komponenten mit der Stromaufnahme nicht so sparsam und die Batterien waren zu schwer, die so eine Schaltung speisen konnten. Danach ist es bei mir eingeschlafen. Kurze Zeit danach habe ich meine aktive Modellflieger Zeit beendet und mich mehr um das Gleitschirmfliegen gekümmert. Letztes Jahr habe ich entschieden wieder zurück zu kommen und mit dem aktiven F1B Wettkampffliegen aufzunehmen. Mein erster Besuch war Ungarn in Mai 2018 um zu sehen was hat sich in den letzten 10 Jahren getan, wo ich nicht mehr aktiv war. Die Entwicklung von elektronischen Geräten die wir mittlerweile in allen Modellen benutzen können hat mich sofort wieder fasziniert und auch motiviert, dass ich auch selbst was basteln soll. Ich habe dann experimentell ein Zeitschalter mit Arduino ähnliches selbst konzipiertes Board gebaut, mit LORA (LONG RANGE) Kommunikation, GPS, Höhenmesser, Bluetooth Connection. Ich habe gemerkt, mit den heutigen Technologien und Möglichkeiten, es ist ein Kinderspiel was zu konzipieren. Dennoch es soll nicht unterschätzt werden, dass es so viele kleine Details gibt, einen guten Zeitschalter von den andern hervorhebt. Da ich nicht so viel Zeit im Moment habe und mich mehr in das Voll Carbon Modellbau widmen möchte, habe ich meine Experimente eingestellt und fertige gekauft. Dennoch als ich gesehen habe, dass man für eine einfache Flasher 50€ bezahlen muss, der auch nichts kann außer zu blinken, wollte ich es doch selbst zu bauen.

Da ich die Open Source Community für meine Entwicklung benutzt habe, es ist meine Bringschuld, dass ich auch mein Design offen lege und zu den Modellfliegern offen zur Verfügung stelle. Daher möchte ich auch hier zu Thermiksense danken, dass sie mir die Möglichkeit anbieten.

Wie sein Name, ein Flasher muss halt nur blinken, dennoch ich war allein damit nicht ganz zufrieden, daher wollte ich ein Paar „nice to have“ reinpacken. Aus diesem Grund entschied ich ein ATtiny13 Chip als Controller zu nutzen und das minimalistische Arduino Kompatibel Bord zu haben. Somit kann ich die Blinkzeiten und Blink Rhythmus vollkommen individuell zu programmieren. Außerdem es hat mir die Möglichkeiten gegeben, dass ich 2 extra Ports habe, so dass ich Start und Stopp Sequenzen beeinflussen könnte.

Nach all diesen Überlegungen kam diese Schaltung heraus.



SJ1 und SJ2 sind die Ports, die ich dann während der Programmierung von dem Board nutzen kann. Nachdem Programmierung kann man während des Betriebs von SJ1 die #2 als PortB0, von SJ2 die #1 als PortB1 und die #2 als PortB2 nutzen und unterschiedliche Funktionen realisieren, wie z.B. starten und stoppen des Blinkens.

Für die Programmierung muss man die SJ1 und SJ2 komplett nutzen.

| SJ1 | SJ2 |
|----------|-----------|
| 1 – VCC | 1 – MISO |
| 2 – MOSI | 2 – SCK |
| 3 – GND | 3 – RESET |

Als Programmierboard habe ich eine andere Arduino UNO Board genommen. Dafür gibt es unzählige Beiträge im Internet. Ich beschreibe es hier ganz kurz. Arduino UNO Board wird mit folgenden Ports zu Flasher angeschlossen

ARDUINO UNO

| | | |
|--------|-------|-----------------|
| PIN 13 | SCK | also SJ2 PIN #2 |
| PIN 12 | MISO | also SJ2 PIN #1 |
| PIN 11 | MISO | also SJ1 PIN #2 |
| PIN 10 | RESET | also SJ2 PIN #3 |

| | | |
|-----|-----|-----------------|
| VCC | VCC | also SJ1 PIN #1 |
| GND | GND | also SJ1 PIN #3 |

Manchmal braucht man ein 10µF Kondensator zwischen Arduino UNO VCC und Arduino UNO GND, aber nicht immer, daher würde ich erst ohne anfangen.

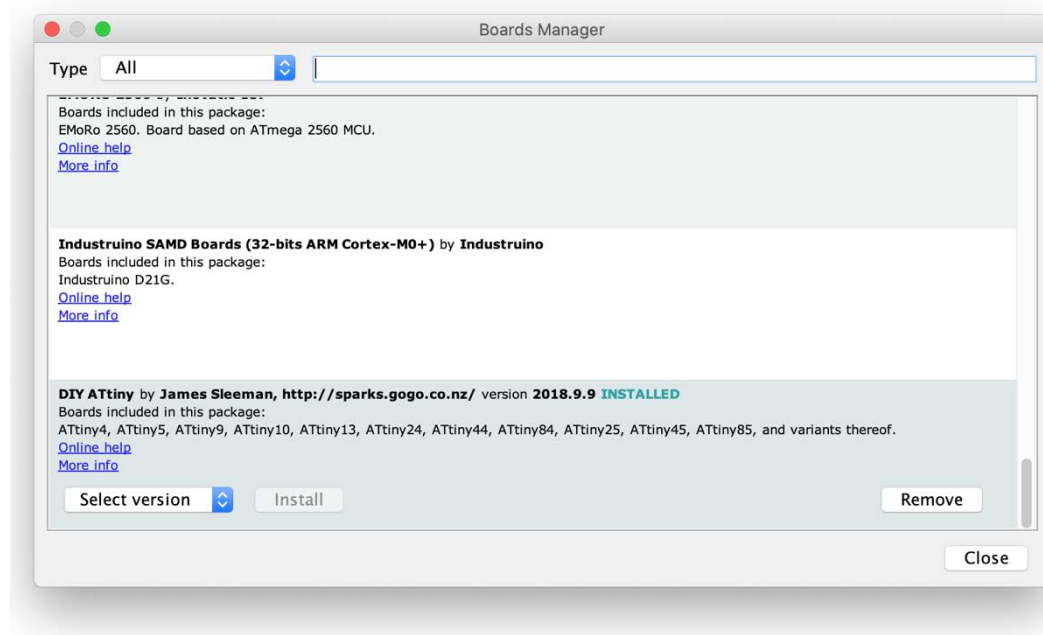
Wenn alles korrekt angeschlossen ist, startet man die Arduino IDE und lädt man über
Files -> Examples -> 11 ArduinoISP -> ArduinoISP (Ich habe englische IDE, auf
Deutsch muss man schauen, wie die Pfade genau heißen)

Nachdem man ArduinoISP Sketch geladen hat, kann man sie zu Arduino UNO hochladen. Ab
jetzt sind wir bereit unser kleines Board zu programmieren. Allerdings uns fehlt noch das
richtige Board und Bootloader von ATtiny13. Ich gebe hier was ich benutze, aber das ist nicht
die einzige Möglichkeit. Man kann über Google andere Varianten auch finden.

Um das richtige Board zu laden, geben wir in den Einstellungen von IDE als additional Board
Manager URL:

https://raw.githubusercontent.com/sleemanj/optiboot/master/dists/package_gogo_diy_attiny_index.json

Wenn man hier den URL richtig eingetragen hat, kann man dann unter Tools -> Board ->
Board Manager aufrufen und unser Board Manager installieren. Man sucht in dem Board
Manager DIY ATtiny by James Sleeman und installiert die letzte Version davon.



Nachdem die Installation fertig ist, kann man dann unter Tools -> Board das richtige wählen
und zwar ATtiny13. Wenn man weiß was man tut, kann man natürlich mit den restlichen
Parametern spielen, aber sonst sollte man alles bei Voreinstellungen lassen.

Sobald man als Board ATtiny13 gewählt hat, steht nichts mehr im Weg unser kleines Board
zu programmieren.

Als erstes soll man natürlich Bootloader installieren damit die kleine Schaltung sich wie ein
Arduino verhalten kann. Das tut man in dem man Tools -> Burn Bootloader wählt.

Sobald Bootloader auf dem Board ist, kann man Sketchs programmieren und auf den Board laden.

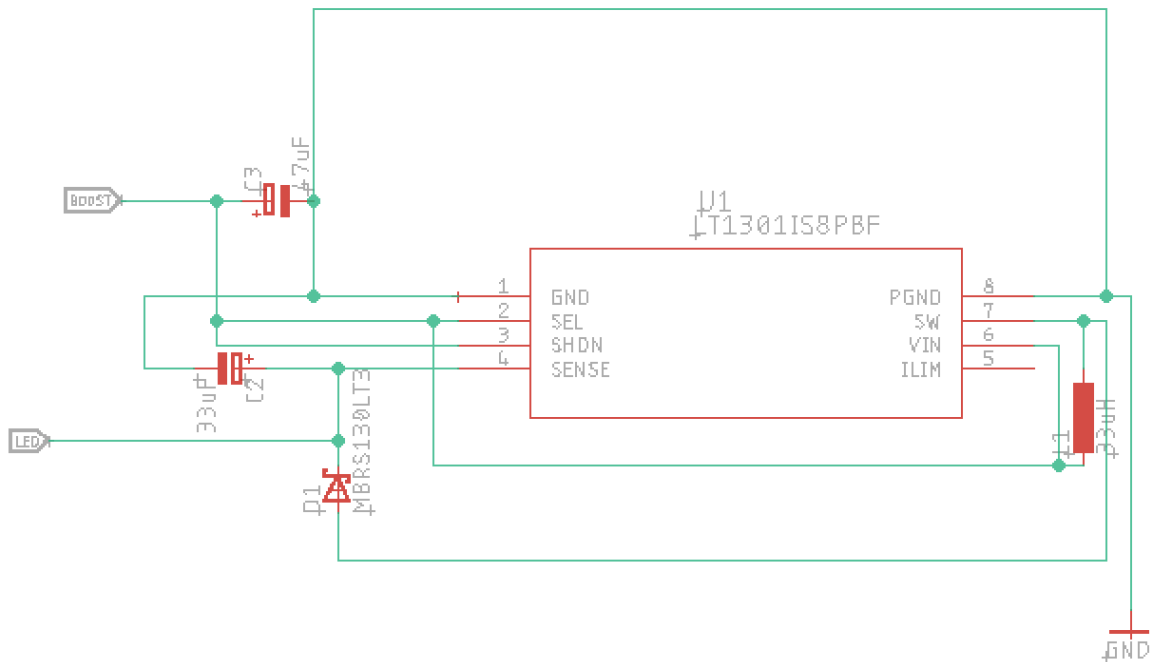
Ich habe als Beispiel immer diesen kleinen Sketch benutzt:

```
const uint8_t LED_PIN = 4;

void setup()
{
  pinMode(LED_PIN, OUTPUT);
}

void loop()
{
  digitalWrite(LED_PIN, HIGH);
  delay(50);
  digitalWrite(LED_PIN, LOW);
  delay(50);
  digitalWrite(LED_PIN, HIGH);
  delay(50);
  digitalWrite(LED_PIN, LOW);
  delay(50);
  digitalWrite(LED_PIN, HIGH);
  delay(50);
  digitalWrite(LED_PIN, LOW);
  delay(900);
}
```

Wenn man jetzt zu dem BOOST Pin ein LED anschließt, kann man das Blinken betrachten. Hiermit sind wir mit dem Controller fertig. Allerdings das ist nicht alles. Wir möchten ja ein LED Blinker haben, welche wir auch unter Umständen von 2 Km Entfernung immer noch sehen möchten. Bei der Größe von LEDs und bei der Stromstärke, dies ist nicht möglich. Daher müssen wir hier ein Trick anwenden. Unseren Speisestrom erhöhen, welchen wir als BOOST bezeichnen. Also wir brauchen ein Booster, der von unserem 3,7 Volt (Normaler 1s LiPo Akku, welche wir für unser Zeitschalter nutzen) ca. 12V machen wird. Hier ist Vorsicht geboten, wenn wir ein 2V bis 5V LED mit 12V Speisen, wird er ganz kurze Zeit sehr hell leuchten und wenn wir die Zeit länger halten es wird nie wieder leuchten. Wenn man mein Programm oben schaut, wird merken, dass meine Leuchtzeiten (PIN, HIGH) 50 Millisekunden sind. Das ist ausreichend um sichtbar zu sein. Diese Art von Verfahren heißt Step-Up Converter und da wir sehr simples brauchen reicht uns völlig ein Beispiel Schema von dem Hersteller der Step-Up Chip zu nehmen. Ich habe mir die LT1301 3.3/5 Volt zu 12 V Step-Up Converter gewählt. In dem Application Notes von dem Chip gab es schon was ich brauchte und es sieht so aus:



Hier muss man beachten, dass C1 und C2 Elektrolyt sind.

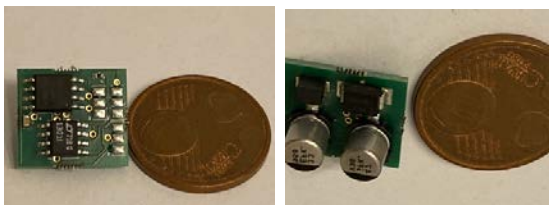
L1 ist eine Spule 33 uH.

D1 ist eine Schottky-Diode, ich nutze MBS130LT3.

BOOST ist der Eingang von dem Controller, LED ist der Ausgang wo Blinker LED's angeschlossen werden.

Als ich die Schaltung auf ein Breadboard gebaut und getestet habe, war ich mit dem Ergebnis zufrieden. Daher habe ich die Schaltung mit Eagle CAD gezeichnet und auch dafür die PCB vorbereitet. Die Dateien kann ich gerne zur Verfügung stellen. Dafür reicht ein Email an mich.

Da ich es für mich schon produzieren wollte, habe ich den Hersteller meines Vertrauens Online kontaktiert und die Dateien Online hochgeladen. Nach 3 Woche standen die Blinker bei mir zu Hause.



Die genauen Maße meiner Entwicklung sind 14 X 14 X 8 mm

Ich denke für F1B habe ich die Kondensatoren ein wenig groß gewählt. Für mich ist es kein Problem, den Platz habe ich schon, aber für einen oder anderen kann das groß sein, also beim Herstellen darauf achten.

Ich habe diese in erster Linie für mich produziert, dennoch habe ich im Moment 200 Stück, also wenn jemand interesse hat und selber nicht basteln/produzieren möchte, kann gerne welche von mir käuflich erwerben. Stück kostet 20€ nackt.

Für programmierung

- Einfache Blonksequenz 5 €
- Gewünschte Funktionen (egal welche, was natürlich möglich ist) plus gewünschte Blink Sequenzen 20€

Wenn jemand zu den Schaltungen oder Programmierung Fragen hat, gerne per Email.

Viel Spass beim basteln.

Can Tezcan

ctezcan65@gmail.com