ATTiny85 Programmer

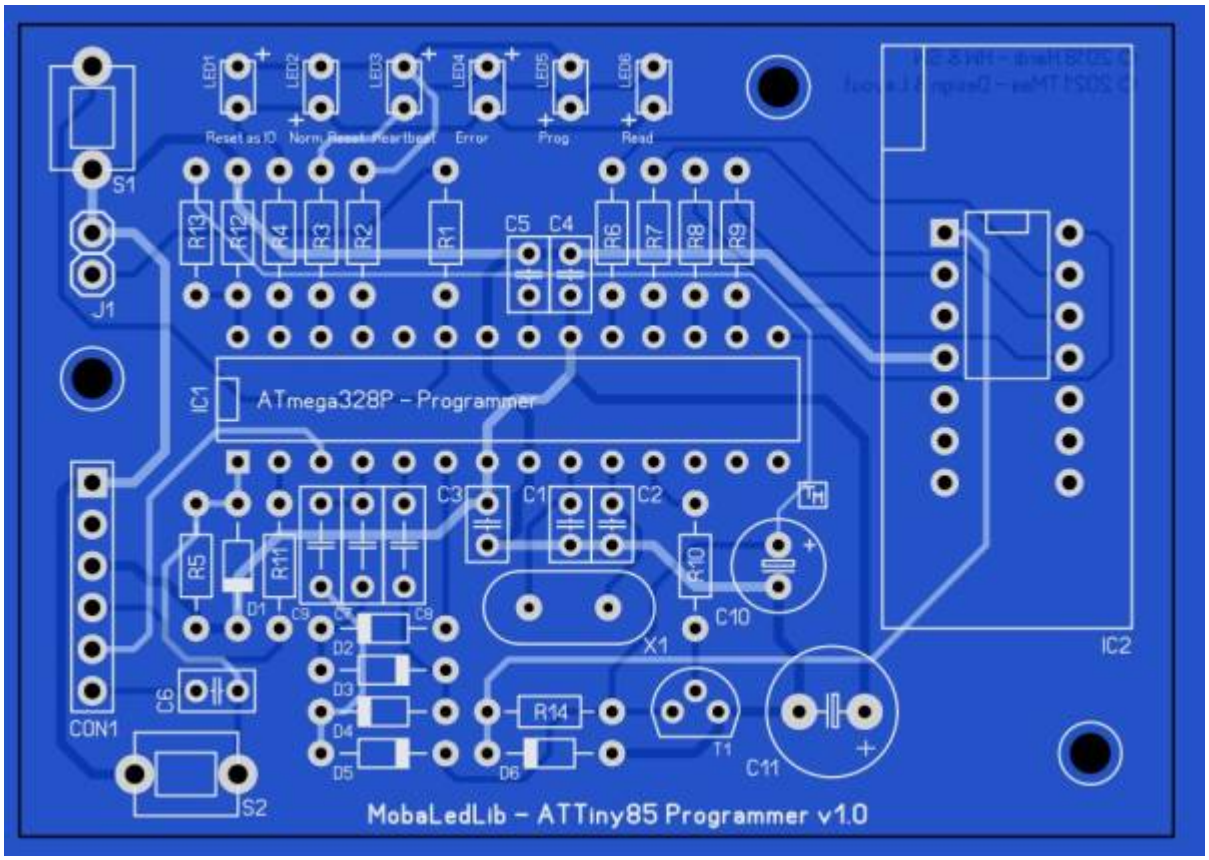
Mit „eigenem“ Prozessor

- ZIF-Fuß (Zero Insertion Force - *Null Einsteckkraft*)
- Kompakte Leiterplatte
- Leicht zu löten
    - Zitat von **Hardi**: Die Platine lässt sich sehr einfach bestücken und hat auf Anhieb funktioniert. (Beitrag #5832)



Bauanleitung

Lage der Komponenten (v1.0)



Um die Teile zu platzieren, ist es ratsam, immer mit den kleinsten (Bauform) Teilen zu beginnen !
Widerstände > Dioden > Kondensatoren > LED's, > und so weiter …

Platzieren Sie die Komponenten gemäß der Stückliste.

Stückliste v1.0

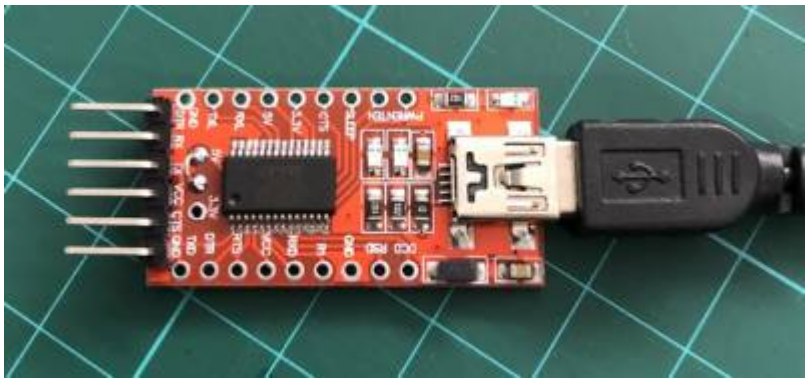
Anzahl	Bezeichnung	Beschreibung	Bestellnummer	Bemerkungen	Montagereihenfolge
1	Platine	MLL Programmer v1.0	Theo [1)]		

FTDI USB-Schnittstelle

Um die Daten aus der Exel Anwendung **Pattern Configurator** uber den Programmieradapter mit ATmega328p zum ATTiny85 laden zu können, wurde eine „separate“ serielle USB-FTDI-Schnittstelle verwendet. Der ATmega328p verfügt standardmäßig nicht über eine integrierte USB-Schnittstelle.

Wenn diese FTDI-Schnittstelle mit einem „schwarzen“ Kondensator ausgestattet

ist, unten rechts direkt neben den Löchern, ist es möglich, die Schnittstelle mit Ihren persönliche Einstellungen zu programmieren. (dies ist mit einem braun/gelben Kondensator Ausführung nicht möglich!) Hierfür können Sie das Tool verwenden, das von der FTDI-Site (FT_Prog) heruntergeladen werden kann. Die Schnittstelle darf dann nicht mit der MobaLedLib-Anwendung verbunden werden.



Seien Sie beim Platzieren der Schnittstelle vorsichtig!

Siehe Foto unten für die richtige Position

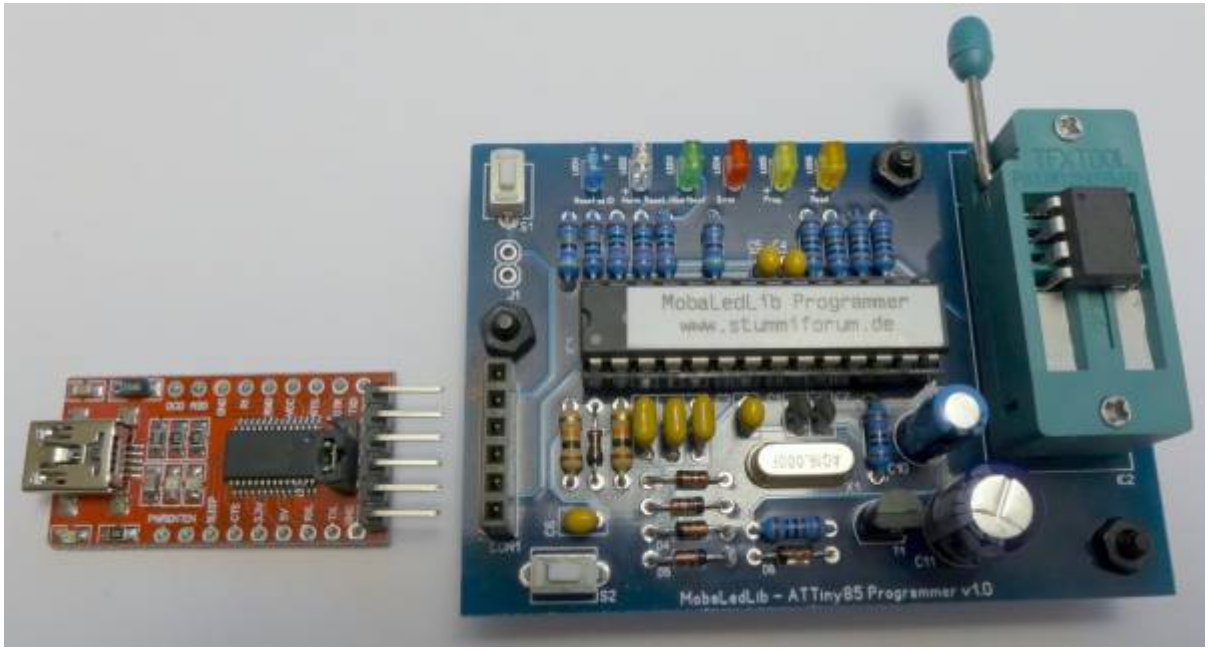


Foto: Hardi

Bootloader

Glücklicherweise muss Folgendes nur einmal passieren !

im Gegensatz zu einem Arduino Uno (der Träger des originalen MLL ATTiny85 Programmierer-Shields), bei dem das erstellte Programm mit den Pattern Configurator in den Mikroprozessor geladen wird, ist

dies mit dieser Version, mit einem ATmega328p, nicht möglich. Der Grund dafür ist, dass im Mikroprozessor kein Bootloader installiert ist. Ein Bootloader ist eine Art Übersetzer, der die für den Mikroprozessor erforderlichen maschinensprache aus den vom Programmgenerator angebotenen Anweisungen extrahiert.

Aber keine Sorge, zum Glück haben wir dafür eine Lösung. Man kann den ATmega328p mit bereits installiertem Bootloader kaufen, etwas teurer, oder ihn selbst in den Mikroprozessor brennen. (Preisunterschied ca. € 2,-) In der Teileliste habe ich die Version mit Bootloader. Sich selbst hochzuladen, ist gar nicht so schwer – schauen Sie sich dazu die Arduino-Seite an. (From Arduino to a Microcontroller on a Breadboard)

Nun muss das Programmierprogramm geladen werden. Hierfür nutzen wir den Pattern Configurator. Verbinden Sie den Programmieradapter über die FTDI-Schnittstelle mit dem Computer und öffnen Sie die Excel-Anwendung Pattern Configurator. Gehen Sie zum Farbkreis, Spezialmodule und wählen Sie **Prog. ISP**. Das Programmierprogramm wird jetzt in den ATmega328p geladen.

Der ATTiny85-Programmieradapter ist jetzt, genau wie die ursprüngliche MLL-Version, einsatzbereit

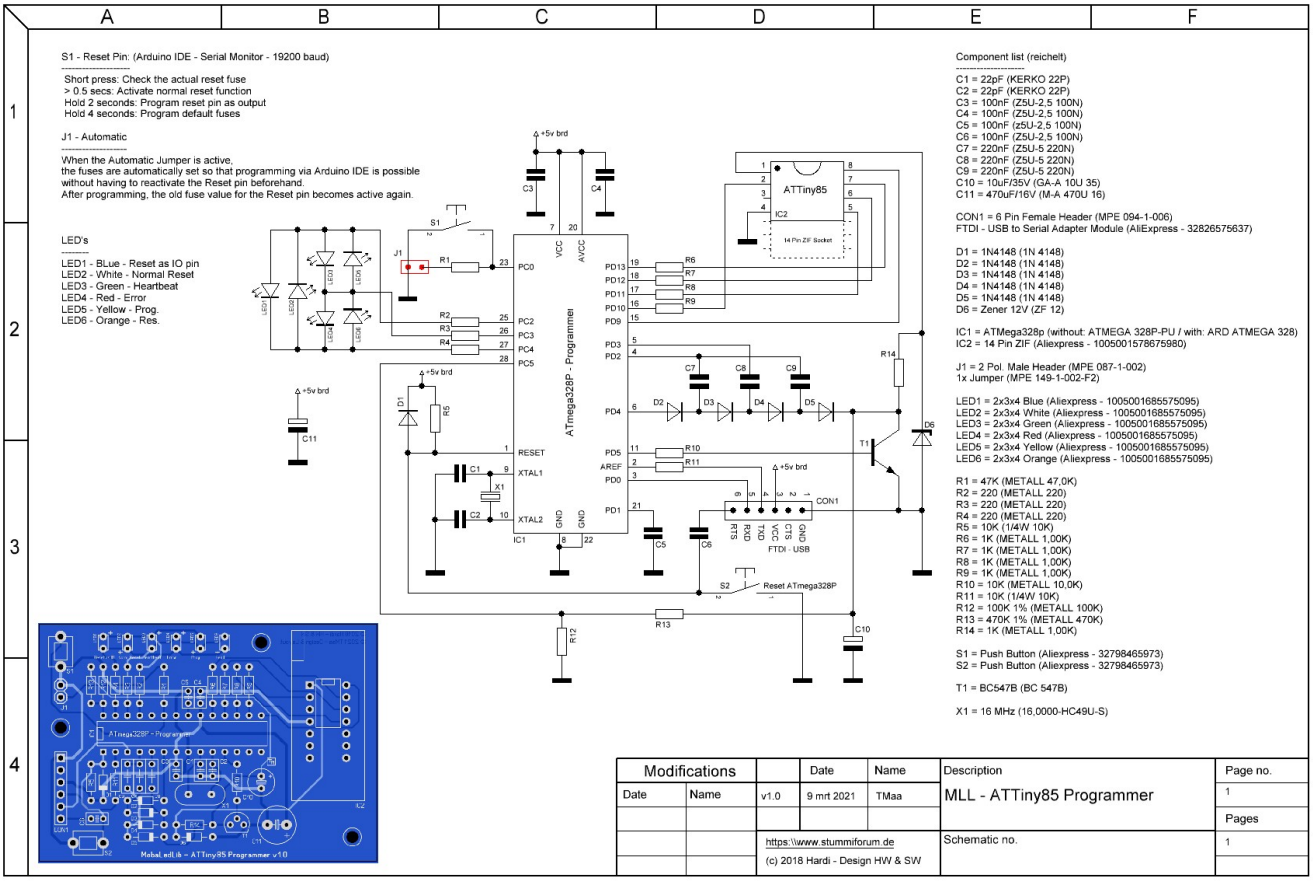
---

Gerber-Dateien

- Zur Eigenfertigung der Leiterplatten, z. B. bei JLCPCB, stehen die Gerber-Dateien zur Verfügung: MLL_ATTiny85_Programmer_v10.zip

---

Schaltplan



1)

Anfrage für eine Platine kann per PN an Theo gesendet werden (stummi: **TMaa**)

From:
https://wiki.mobaledlib.de/ - **MobaLedLib Wiki**

Permanent link:
https://wiki.mobaledlib.de/spezial/user/theo/attiny_programmer_tmaa?rev=1691403289

Last update: **2023/08/07 11:14**