

# ATTiny85 Programmer

## Mit „eigenem“ Prozessor

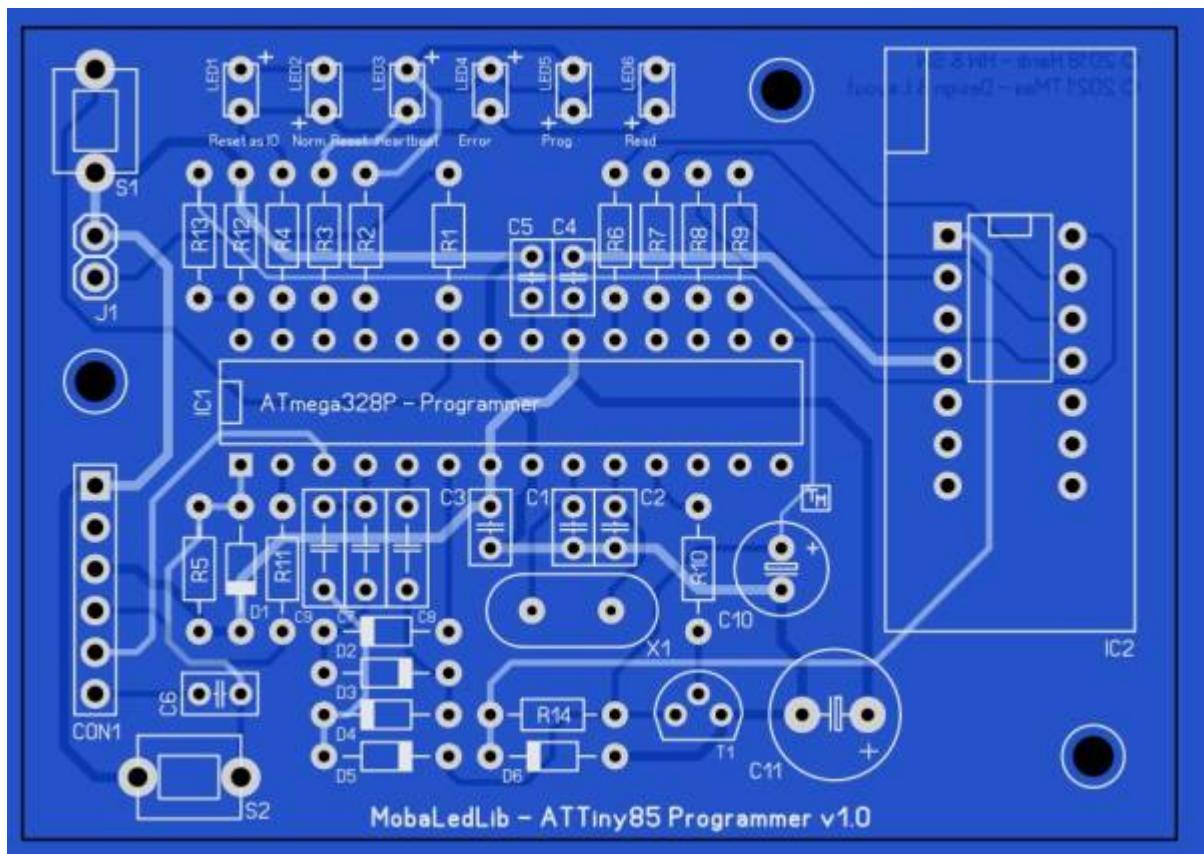
- ZIF-Fuß (Zero Insertion Force - *Null Einstekkraft*)
- Kompakte Leiterplatte
- Leicht zu löten 😊



---

## Bauanleitung

### Lage der Komponenten (v1.0)



**Um die Teile zu platzieren, ist es ratsam, immer mit den kleinsten (Bauform SMD) Teilen zu beginnen !  
Transistoren, IC's, 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 <sup>1)</sup>		

## 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. Auch 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

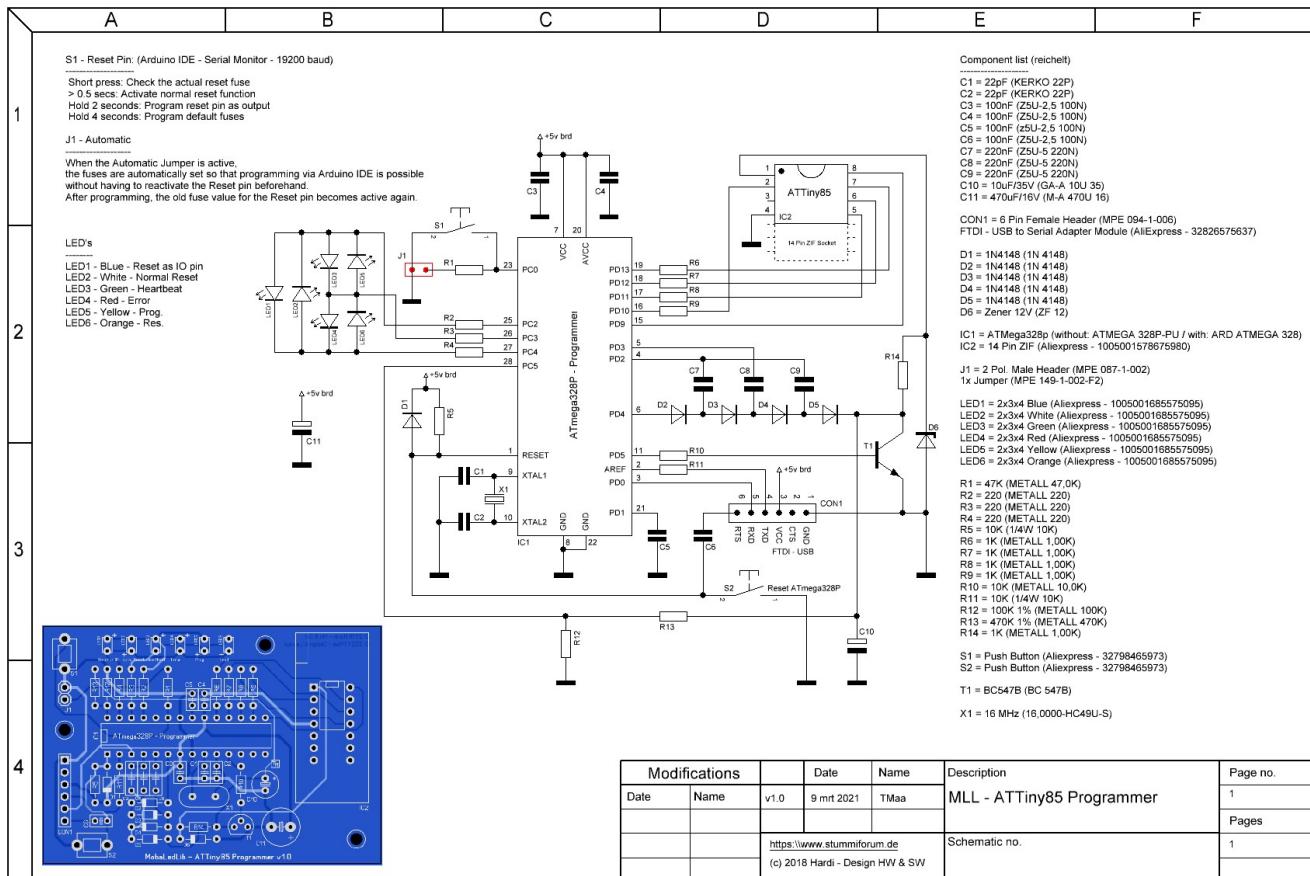
- Zur Eigenfertigung der Leiterplatten, z. B. bei [JLCPCB](#), stehen die Gerber-Dateien zur Verfügung: [MLL\\_ATTiny85\\_Programmer\\_v10.zip](#)
- 

## Schaltplan

Last

update:

2023/08/07 spezial:user:theo:attiny\_programmer\_tmaa https://wiki.mobaledlib.de/spezial/user/theo/attiny\_programmer\_tmaa?rev=1691398619  
09:57



1)

Anfrage für eine Platine kann per PN an [Theo](#) gesendet werden (stummi: **TMa**)

From:

<https://wiki.mobaledlib.de/> - **MobaLedLib** Wiki

Permanent link:

[https://wiki.mobaledlib.de/spezial/user/theo/attiny\\_programmer\\_tmaa?rev=1691398619](https://wiki.mobaledlib.de/spezial/user/theo/attiny_programmer_tmaa?rev=1691398619)

Last update: **2023/08/07 09:57**

