

ATTiny85 Programmer

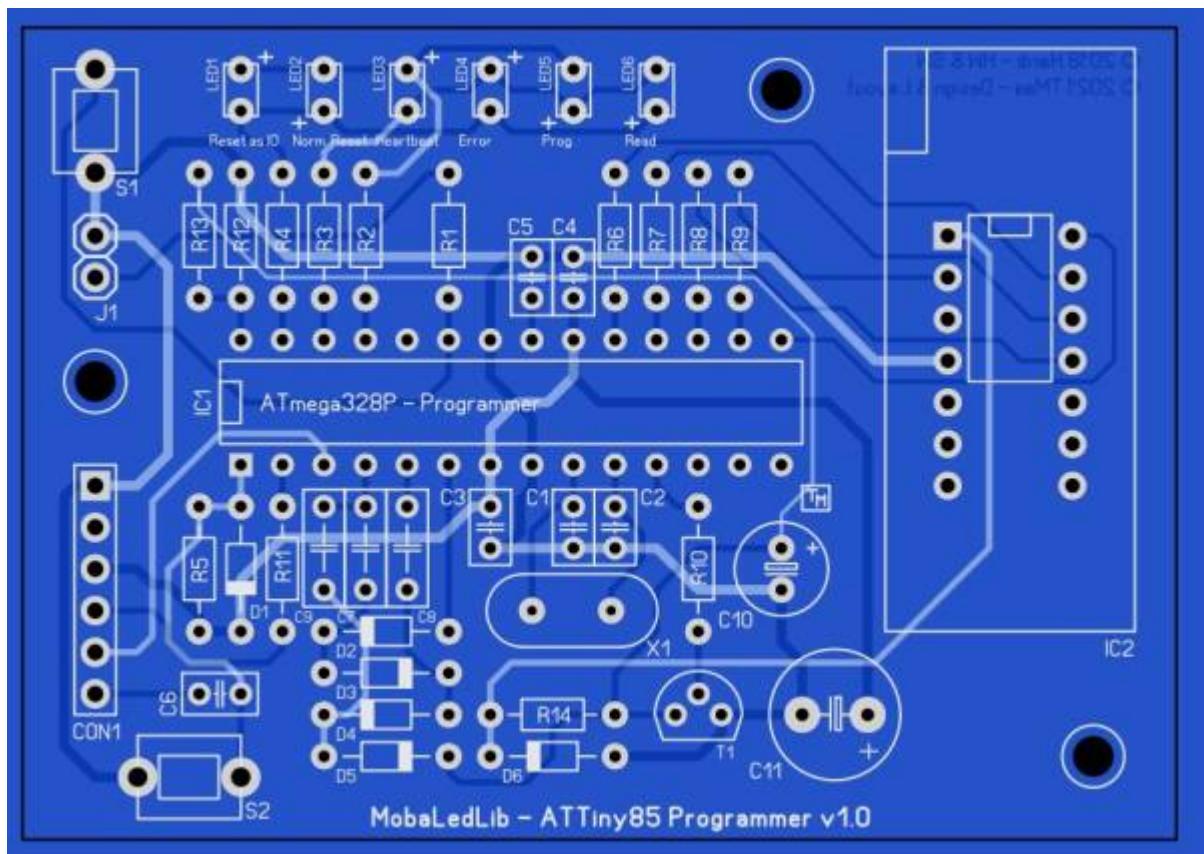
Mit „eigenem“ Prozessor

- ZIF-Fuß (Zero Insertion Force - *Null Einstekkraft*)
- 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 (Baufom) Teilen zu beginnen !

Platzieren Sie die Komponenten gemäß der Stückliste.

Stückliste v1.0

| Anzahl | Bezeichnung | Beschreibung | Bestellnummer | Bemerkungen | Montagereihenfolge |
|--------|----------------|--|--------------------|-------------|--------------------|
| 1 | Platine | MLL Programmer v1.0 | Theo ¹⁾ | | |
| 2 | C1, C2 | Keramik-Kondensator 22 pF, 5 %, NPO, 100 V, RM 2,54 | KERKO 22P | | |
| 4 | C3, C4, C5, C6 | Vielschicht-Kerko 100 nF, 50/100 V, Z5U 20%, RM 2,5 | Z5U-2,5 100N | | |
| 3 | C7, C8, C9 | Vielschicht-Keramikkondensator 220N, 20% | Z5U-2,5 220N | | |
| 1 | C10 | Elko, radial, 10 µF, 35 V, RM 2,0, 1000h, 105°C, 20% | GA-A 10U 35 | | |
| 1 | C11 | Elko, radial, 470 µF, 16 V, RM 3,5, 85°C, 2000h, 20% | M-A 470U 16 | | |

| Anzahl | Bezeichnung | Beschreibung | Bestellnummer | Bemerkungen | Montagereihenfolge |
|--------|------------------------------------|---|------------------------------|--|--------------------|
| 1 | CON1 | Buchsenleisten 2,54 mm, 1x06, gerade | MPE 094-1-006 | | |
| 1 | | FTDI USB Interface | FT232RL | Jumper auf 5V-Auswahl stellen! | - |
| 5 | D1, D2, D3, D4, D5 | Schalt-Diode, 100 V, 150 mA, DO-35 | 1N 4148 | | |
| 1 | D6 | Zenerdiode, 12 V, 0,5 W, DO-35 | ZF 12 | | |
| 1 | IC1 | Arduino - ATmega328 mit Arduino Bootloader | ARD ATMEGA 328 | | |
| 1 | IC2 | 14 Pin ZIF | 2 Stück Zif-Buchse 14-polige | | |
| 1 | J1 | Stiftleisten 2,54 mm, 1x02, gerade | MPE 087-1-002 | | |
| 1 | | Jumper 2,54 mm, geöffnet, grün | MPE 149-1-002-F2 | | |
| 1 | LED1, LED2, LED3, LED4, LED5, LED6 | LED 2x3x4 mm | LED Kit primäre farbe | LED1-Blau, LED2-Weiss, LED3-Grün, LED4- Rot, LED5-Gelb, LED6-Orange (kompakte Bauform für RM 2,54, nebeneinander auf einem Raster platzierbar) | |
| 1 | R1 | Widerstand, Metallschicht, 47,0 kOhm, 0207, 0,6 W, 1% | METALL 47,0K | | |
| 3 | R2, R3, R4 | Widerstand, Metallschicht, 220 Ohm, 0207, 0,6 W, 1% | METALL 220 | | |
| 3 | R5, R10, R11 | Widerstand, Metallschicht, 10,0 kOhm, 0207, 0,6 W, 1% | METALL 10,0K | | |
| 5 | R6, R7, R8, R9, R14 | Widerstand, Metallschicht, 1,00 kOhm, 0207, 0,6 W, 1% | METALL 1,00K | | |
| 1 | R12 | Widerstand, Metallschicht, 100 kOhm, 0207, 0,6 W, 1% | METALL 100K | | |
| 1 | R13 | Widerstand, Metallschicht, 470 kOhm, 0207, 0,6 W, 1% | METALL 470K | | |

FTDI USB-Schnittstelle

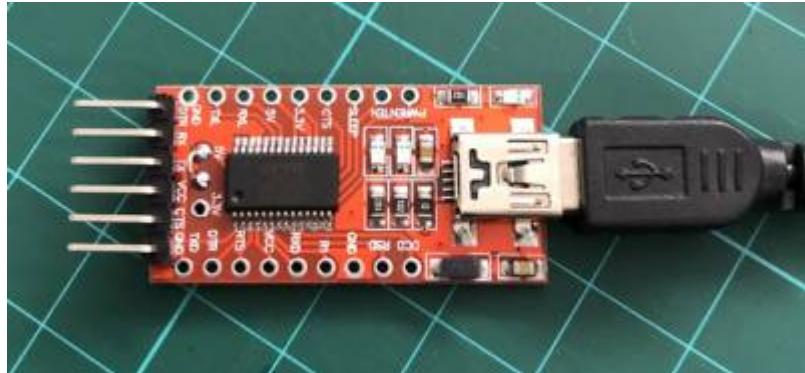
Um die Daten aus der Excel Anwendung **Pattern Configurator** über den Programmieradapter mit ATmega328p zum ATTiny85 laden zu können, wurde eine „separate“ serielle USB-FTDI-Schnittstelle verwendet. Der ATmega328p verfügt standardmäßig nicht über eine integrierte USB-Schnittstelle.



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

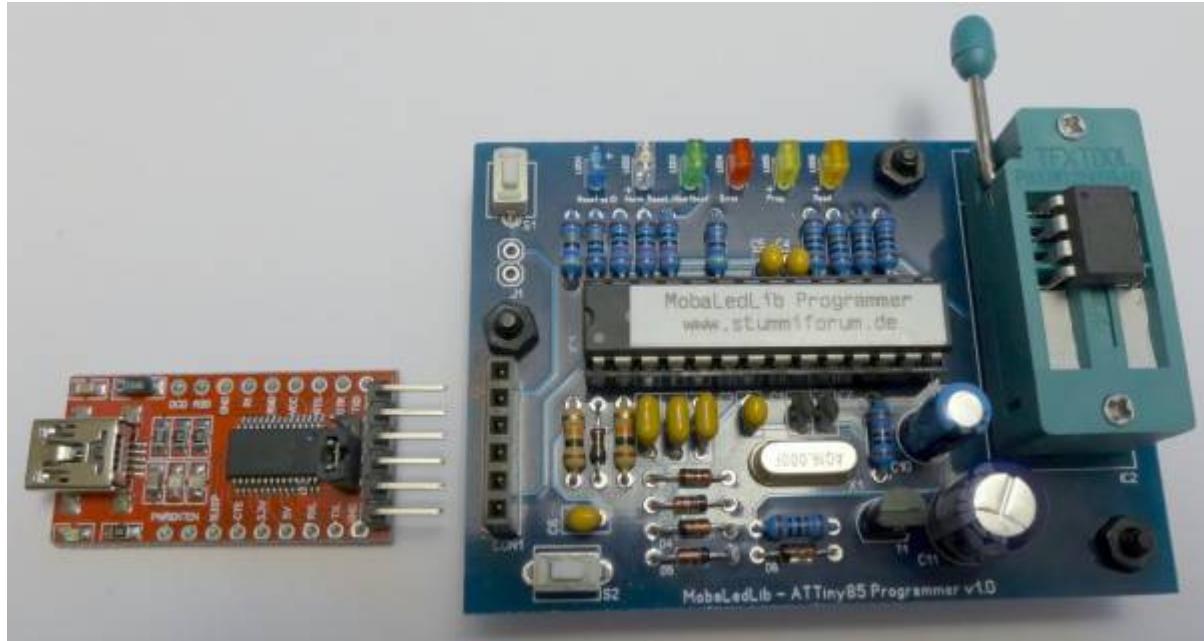


Foto: Hardi

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

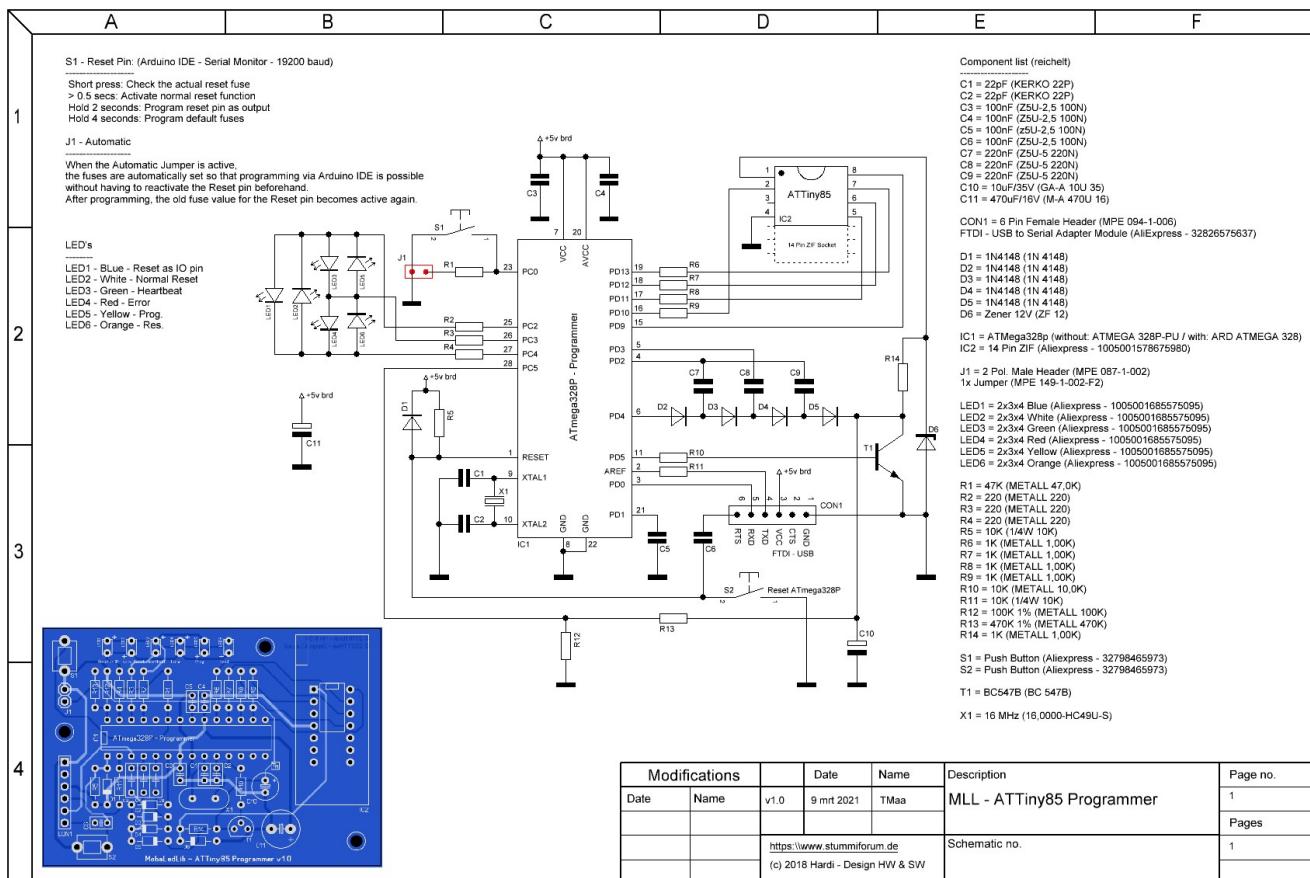
- Zur Eigenfertigung der Leiterplatten, z. B. bei [JLCPCB](#), stehen die Gerber-Dateien zur Verfügung: [MLL_ATTiny85_Programmer_v10.zip](#)
-

Schaltplan

Last

update:

2023/08/07 spezial: user: theo: attiny_programmer_tmaa https://wiki.mobaledlib.de/spezial/user/theo/attiny_programmer_tmaa?rev=1691409574
12:59



1)

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

From:

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

Permanent link:

https://wiki.mobaledlib.de/spezial/user/theo/attiny_programmer_tmaa?rev=1691409574

Last update: **2023/08/07 12:59**

