

ATTiny85 Programmer

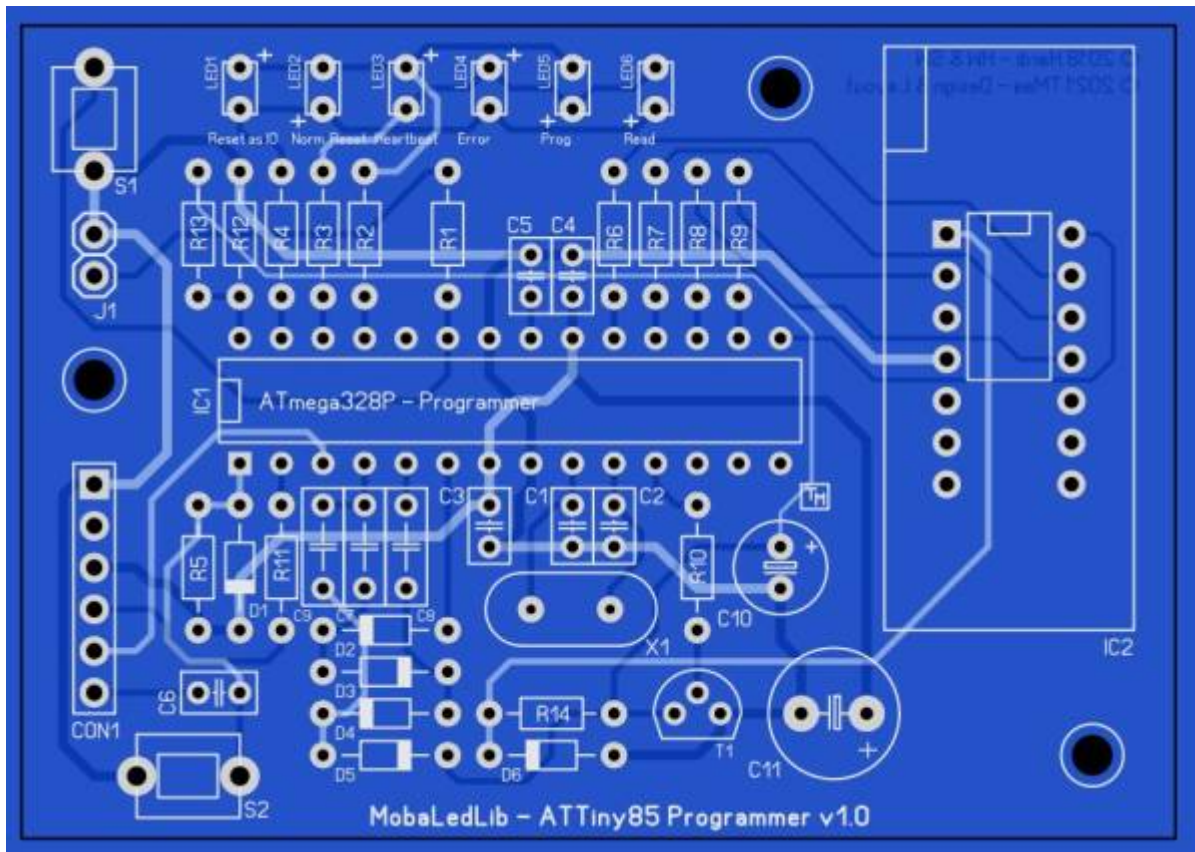
Mit „eigenem“ Prozessor

- 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



i 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⁾		
2	C1, C2	Keramik-Kondensator 22 pF, 5 %, NPO, 100 V, RM 2,54	KERKO 22P		3
4	C3, C4, C5, C6	Vielschicht-Kerko 100 nF, 50/100 V, Z5U 20%, RM 2,5	Z5U-2,5 100N		3
3	C7, C8, C9	Vielschicht-Keramikkondensator 220N, 20%	Z5U-2,5 220N		3
1	C10	Elko, radial, 10 µF, 35 V, RM 2,0, 1000h, 105°C, 20%	GA-A 10U 35	Achten Sie auf die richtige Ausrichtungsrichtung	10
1	C11	Elko, radial, 470 µF, 16 V, RM 3,5, 85°C, 2000h, 20%	M-A 470U 16	Achten Sie auf die richtige Ausrichtungsrichtung	10

Anzahl	Bezeichnung	Beschreibung	Bestellnummer	Bemerkungen	Montagereihenfolge
1		Buchsenleisten 2,54 mm, 1x06, gerade	MPE 094-1-006		9
1	CON1	FTDI USB Interface	FT232RL	Jumper auf 5V-Auswahl stellen! - Achten Sie auf die richtige Ausrichtungsrichtung	-
5	D1, D2, D3, D4, D5	Schalt-Diode, 100 V, 150 mA, DO-35	1N 4148	Achten Sie auf die richtige Ausrichtungsrichtung	2
1	D6	Zenerdiode, 12 V, 0,5 W, DO-35	ZF 12	Achten Sie auf die richtige Ausrichtungsrichtung	2
1	IC1	Arduino - ATmega328 mit Arduino Bootloader	ARD ATMEGA 328	Achten Sie auf die richtige Ausrichtungsrichtung	12
1		IC-Sockel, 28-polig, doppelter Federkontakt	GS 28-S		4
1	IC2	14 Pin ZIF	2 Stück Zif-Buchse 14-polige	Achten Sie auf die richtige Ausrichtungsrichtung	11
1	J1	Stiftleisten 2,54 mm, 1x02, gerade	MPE 087-1-002		8
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) - Achten Sie auf die richtige Ausrichtungsrichtung	6
1	R1	Widerstand, Metallschicht, 47,0 kOhm, 0207, 0,6 W, 1%	METALL 47,0K		1
3	R2, R3, R4	Widerstand, Metallschicht, 220 Ohm, 0207, 0,6 W, 1%	METALL 220		1
3	R5, R10, R11	Widerstand, Metallschicht, 10,0 kOhm, 0207, 0,6 W, 1%	METALL 10,0K		1
5	R6, R7, R8, R9, R14	Widerstand, Metallschicht, 1,00 kOhm, 0207, 0,6 W, 1%	METALL 1,00K		1
1	R12	Widerstand, Metallschicht, 100 kOhm, 0207, 0,6 W, 1%	METALL 100K		1
1	R13	Widerstand, Metallschicht, 470 kOhm, 0207, 0,6 W, 1%	METALL 470K		1
2	S1, S2	Schalter DIP 2 Pins 3mmx 6mmx 4,3mm	PCB Taster	Kompakte Bauform, platzsparend	7
1	X1	Standardquarz, Grundton, 16,000000 MHz	16,0000-HC49U-S		5

FTDI USB-Schnittstelle

Um die Daten aus der Exel 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

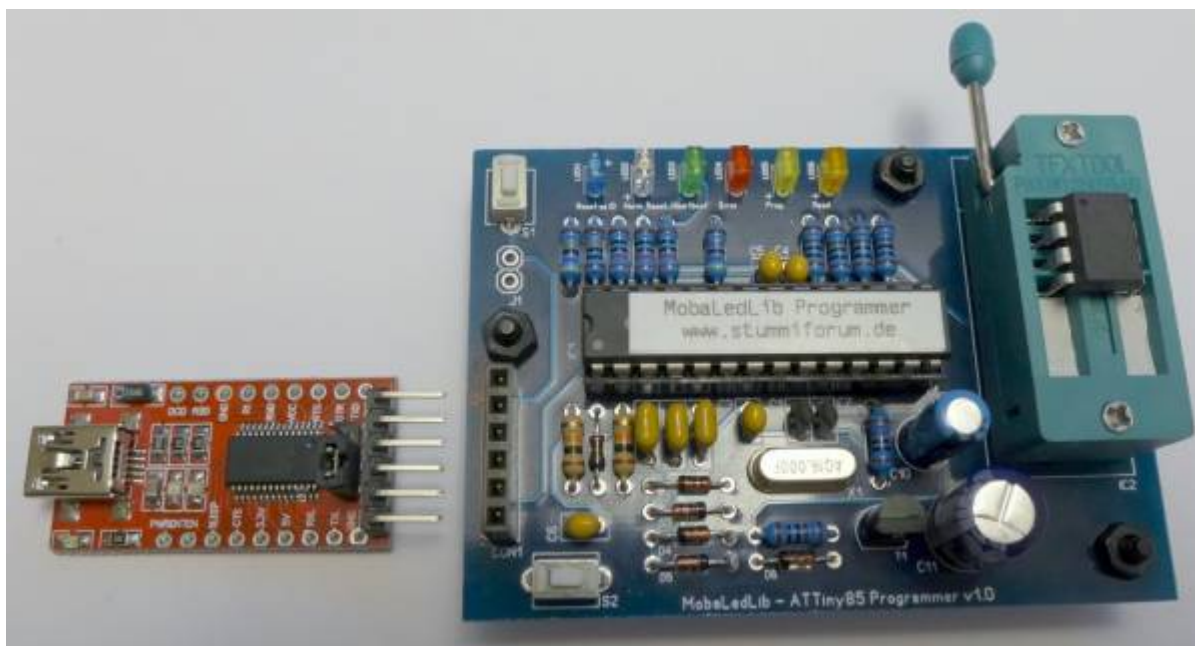


Foto: Hardi

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.

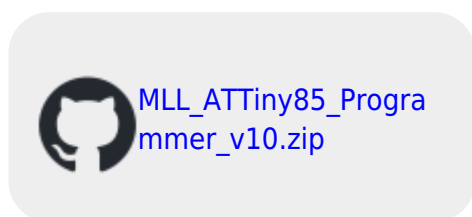
Aber keine Sorge, zum Glück haben wir dafür eine Lösung. Man kann den ATmega328p mit bereits installiertem Bootloader kaufen, etwas teurer, oder ihn selbst in den Mikroprozessor brennen. (Preisunterschied ca. € 2,-) In der Teileliste habe ich die Version mit Bootloader. Sich selbst hochzuladen, ist gar 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 [JLPCB](#), stehen die Gerber-Dateien zur Verfügung:



Schaltplan

S1 - Reset Pin (Arduino IDE - Serial Monitor - 19200 baud)

Short press: Check the actual reset fuse
 > 0.5 secs: Activate normal reset function
 Hold 2 seconds: Program reset pin as output
 Hold 4 seconds: Program default fuses

J1 - Automatic

When the Automatic Jumper is active, the fuses are automatically set so that programming via Arduino IDE is possible without having to reactivate the Reset pin beforehand. After programming, the old fuse value for the Reset pin becomes active again.

LED's

- LED1 - BLue - Reset as IO pin
- LED2 - White - Normal Reset
- LED3 - Green - Heartbeat
- LED4 - Red - Error
- LED5 - Yellow - Prog.
- LED6 - Orange - Res.

Component list (reichelt)

- C1 = 22pF (KERKO 22P)
- C2 = 22pF (KERKO 22P)
- C3 = 100nF (Z5U-2.5 100N)
- C4 = 100nF (Z5U-2.5 100N)
- C5 = 100nF (Z5U-2.5 100N)
- C6 = 100nF (Z5U-2.5 100N)
- C7 = 220nF (Z5U-5 220N)
- C8 = 220nF (Z5U-5 220N)
- C9 = 220nF (Z5U-5 220N)
- C10 = 10uF/35V (GA-A 10U 35)
- C11 = 470uF/16V (M-A 470U 16)

CON1 = 6 Pin Female Header (MPE 094-1-006)
 FTDI - USB to Serial Adapter Module (AliExpress - 32826575637)

D1 = 1N4148 (1N 4148)
 D2 = 1N4148 (1N 4148)
 D3 = 1N4148 (1N 4148)
 D4 = 1N4148 (1N 4148)
 D5 = 1N4148 (1N 4148)
 D6 = Zener 12V (ZF 12)

IC1 = ATmega328p (without: ATMEGA 328P-PU / with: ARD ATMEGA 328)
 IC2 = 14 Pin ZIF (Aliexpress - 1005001578675950)

J1 = 2 Pol. Male Header (MPE 087-1-002)
 1x Jumper (MPE 149-1-002-F2)

LED1 = 2x3x4 Blue (Aliexpress - 1005001685575095)
 LED2 = 2x3x4 White (Aliexpress - 1005001685575095)
 LED3 = 2x3x4 Green (Aliexpress - 1005001685575095)
 LED4 = 2x3x4 Red (Aliexpress - 1005001685575095)
 LED5 = 2x3x4 Yellow (Aliexpress - 1005001685575095)
 LED6 = 2x3x4 Orange (Aliexpress - 1005001685575095)

R1 = 47K (METALL 47,0K)
 R2 = 220 (METALL 220)
 R3 = 220 (METALL 220)
 R4 = 220 (METALL 220)
 R5 = 10K (1/4W 10K)
 R6 = 1K (METALL 1,00K)
 R7 = 1K (METALL 1,00K)
 R8 = 1K (METALL 1,00K)
 R9 = 1K (METALL 1,00K)
 R10 = 10K (METALL 10,0K)
 R11 = 10K (1/4W 10K)
 R12 = 100K 1% (METALL 100K)
 R13 = 470K 1% (METALL 470K)
 R14 = 1K (METALL 1,00K)

S1 = Push Button (Aliexpress - 32798465973)
 S2 = Push Button (Aliexpress - 32798465973)

T1 = BC547B (BC 547B)

X1 = 16 Mhz (16,0000-HC49U-S)

Modifications		Date	Name	Description	Page no.
Date	Name	v1.0	9 mt 2021	Tmaa	MLL - ATtiny85 Programmer
					Pages
https://www.stummforum.de (c) 2018 Hardi - Design HW & SW					Schematic no.
					1

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

Last update: **2023/12/06 10:15**

