

# EXA-PROG

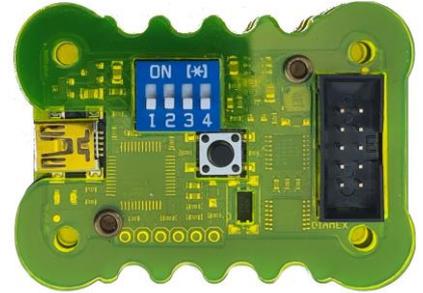
USB-Programmieradapter für AVR-ISP, AVR-UPDI, STM32, LPC, ESP8266, ESP32

---

## EXA = EXAMIN

EXA-PROG – abgeleitet vom lateinischen „Examin“ = überprüfen. EXA-PROG hat eine eingebaute Funktion zur Selbstüberprüfung aller Signale auf den Programmierleitungen mit dem dazugehörigen Exa-Tool.

EXA-Prog steht für die neue Generation der ISP-Programmer, die nicht nur auf einen speziellen Microcontroller-Typ beschränkt sind, sondern mehrere unterschiedliche Controller-Architekturen und Programmierschnittstellen unterstützt.



### Diese Microcontroller können mit Exa-Prog programmiert werden:

- ATMEL/Microchip AVR, MegaAVR, TinyAVR-Controller mit ISP-Schnittstelle (MISO, MOSI, SCK, RESET) oder alternativ (PDI, PDO, SCK, RESET).
- ATMEL/Microchip TinyAVR, AVR-DA-Controller mit UPDI-Schnittstelle. Mit Hochvolt-Funktion für Controller mit deaktiviertem UPDI-Pin.
- STM32 32-Bit ARM-Cortex-M-Controller, die über einen internen UART-Bootloader verfügen.
- NXP LPC 32-Bit ARM-Cortex-M-Controller, die über einen internen UART-Bootloader verfügen.
- ESP8266, ESP32, ESP32-S2 Controller und Module über den internen UART-Bootloader.

### Fähigkeiten:

- Funktionswahl über DIP-Schalter.
- Signalpegel umschaltbar.
- Integrierter Hochvolt-Generator für UPDI-Programmierung.
- Automatische Bitratenanpassung im AVR-ISP-Modus
- Taktgenerator für AVR-Controller mit ver“fuse“tem Oszillator.
- Taster zum Reset des angeschlossenen Microcontrollers.
- 10-poliger Standard-ISP-Anschluss. Optionales Zubehör: 10-pol auf 6-pol Adapter, ESP01-Adapter.
- Mini-USB-Anschluss zur Stromversorgung und Verbindung mit dem PC.
- Firmware updatefähig über USB.
- Windows-PC-Tool zum Test der Signalpegel am Programmieranschluss.

### Technische Daten:

- Stromaufnahme ohne angeschlossenen Microcontroller: ca. 30mA
- Signalpegel am Programmieranschluss: 5V (USB-Spannung) oder 3,3V
- Stromversorgung für externe Schaltung: Max. 300mA (3,3V), max. 500mA (5V)
- UPDI-Hochspannung: ca. 12,3V
- Gehäusegröße ca. 55x40x10mm, Gewicht ca. 25g

### Unterstützte Software:

- AVR/Microchip-Studio ab v7.x (nur AVR-ISP: STK500).
- AVRDUDE ab v6.3 (AVR-ISP: STK500, AVR-UPDI: jtag2updi).
- Entwicklungsumgebungen, die auf AVRDUDE als Programmierinterface basieren, z.B. Arduino, Bascom.
- ESPTOOL (esptool.py, esptool.exe) und alle Programme, die ESPTOOL als Programmierinterface benutzen.

Hinweis: In der aktuellen Firmware-Version kann EXA-PROG nicht unter ATMEL/MICROCHIP-Studio zur Programmierung von AVR-Controllern mit UPDI-Schnittstelle benutzt werden. Sie können das Studio jedoch als Editor und Compiler nutzen, die erzeugten HEX-Dateien müssen dann extern mit AVRDUDE in den Microcontroller übertragen werden. Wie das funktioniert, finden Sie in einem Link am Ende dieser Anleitung.

# EXA-PROG

USB-Programmieradapter für AVR-ISP, AVR-UPDI, STM32, LPC, ESP8266, ESP32

---

## INSTALLATION

Verbinden Sie EXA-PROG über ein Mini-USB-Kabel mit einem beliebigen USB-2.x oder USB 3.x Port Ihres PC oder Notebook.

### Windows 7 und 8.x

EXA-PROG wird über einen virtuellen COM-Port angesprochen. Unter Windows 7 und 8.x muss hierzu die Treiberdatei installiert werden. Im Geräte manager von Windows den entsprechenden Eintrag anklicken, im Reiter "Treiber" auf "Treiber aktualisieren". Dann die Datei **prog-isp.inf** aus dem EXA-PROG-Programm paket auswählen.

### Windows 8.x

Sollten Sie noch Windows 8.x benutzen, müssen Sie eventuell das „Erzwingen der Treibersignatur deaktivieren“. Wie das funktioniert, können sie leicht mit einer Google-Suche herausfinden. Falls möglich, sollten Sie an einen Umstieg auf Windows 10 denken.

### Windows 10

Unter Windows 10 ist keine Installation eines Treibers erforderlich. Hier meldet sich jede Komponente als „Seriell es USB-Gerät (COMx)“ an.

Welcher COM-Port für Ihre verwendete Komponente gültig ist, können Sie am besten testen, indem Sie das EXA-PROG-Tool starten und die Liste der COM-Ports anklicken bevor und nachdem Sie die Komponente angesteckt haben. Der COM-Port der hinzugekommen ist, gehört zu Ihrem EXA-PROG.

EXA-PROG verfügt über kein eigenes Programm zur Programmierung von Microcontrollern. Die Firmware des EXA-PROG wurde kompatibel zu vielen vorhandenen Programmier tools der Hersteller oder zu frei verfügbaren Tools geschrieben.

Im Auslieferungszustand ist EXA-PROG auf den Modus AVR-ISP mit 3,3V Spannung eingestellt (alle DIP-Schalter auf OFF).

Im AVR-ISP-Modus ist EXA-PROG kompatibel zu **STK500** und funktioniert mit AVR/ATMEL/MICROCHIP-Studio sowie AVRDUDE (Arduino, Bascom).

⇒ **Hinweis: Unter Windows 10 bitte nur ATMEL/MICROCHIP-Studio ab 7.x benutzen, ältere Versionen funktionieren nicht.**

Im UPDI-Modus emuliert EXA-PROG einen **JTAG2UPDI**-Programmer. Dieser kann derzeit nur mit **AVRDUDE** benutzt werden (siehe Link am Ende dieser Anleitung).

Der ESP-Modus ist kompatibel zu **ESPTOOL** (esptool.py, esptool.exe).

Die Programmier-Modi STM32 und NXP/LPC sind weitgehend für die Benutzung eigener Programmier tools vorgesehen.

Achten Sie bitte immer darauf, dass die DIP-Schalter passend zu Ihrem zu programmierenden Microcontroller eingestellt sind. Unbedingt auch auf die Programmierspannung achten, AVR-Controller lassen sich mit 3,3V und 5V betreiben, nahezu alle 32-Bit Microcontroller dürfen NUR mit 3,3V betrieben werden.

Schalten Sie die DIP-Schalter nicht während der Programmierung um. Eine Umschaltung führt immer zu einem Reset der Programmers, dies wird durch ein kurzes Flackern der roten LED angezeigt. Ein konstantes Leuchten der grünen LED zeigt die Betriebsbereitschaft des Programmers an.

Drücken Sie nicht den RESET-Taster während eines Programmier vorganges. Die Programmierung wird abgebrochen und die Daten im Flash-Speicher des Microcontrollers können ungültig sein.

# EXA-PROG

USB-Programmieradapter für AVR-ISP, AVR-UPDI, STM32, LPC, ESP8266, ESP32

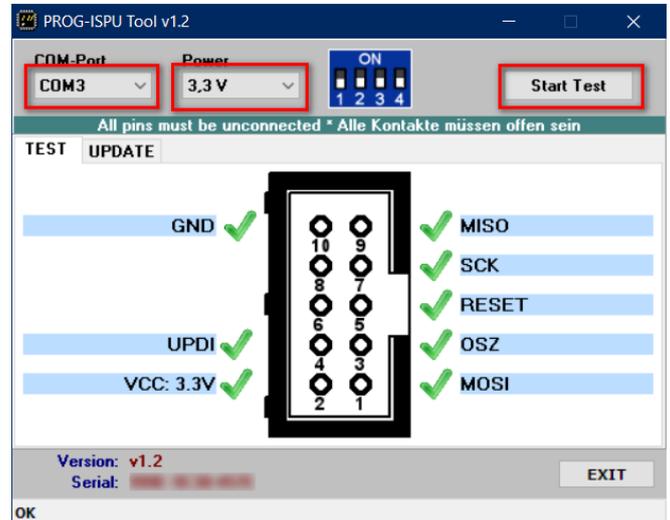
## TESTFUNKTION

Wenn Sie Probleme haben, einen angeschlossenen Microcontroller zu programmieren, können Sie folgendermaßen überprüfen, ob sich der Fehler beim Programmieradapter befindet. EXA-PROG verfügt über eine integrierte Test-Hardware, mit der die Signalpegel an der Programmier-Schnittstelle überprüft werden können.

Starten Sie bitte hierzu das EXA-PROG-Tool unter Windows (Dateiname: Exa\_Tool\_xxx.exe; xxx = Version) aus dem EXA-PROG Programmpaket.

1. Reiter „TEST“ anwählen.
2. COM-Port des EXA-PROG einstellen. Wenn Sie nicht genau wissen, welchen COM-Port Ihr EXA-PROG belegt, können Sie diesen einfach ermitteln, indem Sie die COM-Port Liste vor und nach dem Anstecken des Programmers an den PC vergleichen.
3. Spannung wählen (Power).
4. DIP-Schalter am Programmer nach dem angezeigten Muster einstellen.
5. Entfernen Sie alle Leitungen vom Programmieranschluss des EXA-PROG.
6. Button „Start Test“ drücken. Es werden nun nacheinander alle Ports angesteuert und die Signalpegel gemessen.

⇒ Bitte für den Test den Programmieranschluss am Programmer unbedingt offen lassen. Es dürfen keine Verbindungen zu einem Microcontroller oder einer externen Schaltung vorhanden sein.



Über den Reiter „UPDATE“ können Sie auch ein Update der Firmware Ihres EXA-PROG vornehmen, falls dies erforderlich ist.

### **POSITIV:**

Sind alle Signalpegel in Ordnung, liegt das Programmier-Problem wahrscheinlich an der Verbindung zum Microcontroller oder an einer falsch eingestellten Programmier-Software.

### **NEGATIV:**

Sollte der Test negativ ausfallen, überprüfen Sie zunächst, ob die Spannung richtig eingestellt ist und ob die DIP-Schalter auf dem EXA-PROG mit der angezeigten Schalterstellung übereinstimmen. Sind wirklich keine externen Bauteile mit dem Programmieranschluss verbunden? Überprüfen Sie auch die USB-Spannung mit einem Multimeter indem Sie DIP-Schalter 1 auf ON stellen und die Spannung zwischen PIN2 (Plus) und PIN10 (Minus) am Programmieranschluss messen. Diese muss zwischen 4,5V und 5,2V betragen.

Wenn sich trotz aller Tests und weiterer Überprüfungen herausstellt, dass Ihr EXA-PROG defekt ist, wenden Sie sich bitte an den Händler, bei dem Sie den Programmer gekauft haben. Geben Sie bitte immer die genaue Fehlerbeschreibung an, eventuell auch mit einem Bildschirmfoto des EXA-PROG-Tools.

### **PRÄVENTIV:**

Leider kommt es immer wieder vor, dass bei Experimenten mit der Programmierung von Microcontrollern der Programmieradapter zerstört wird. Das passiert übrigens auch den absoluten Experten, deshalb haben die meistens Entwickler einen zweiten Programmer in Reserve, damit das aktuelle Projekt nicht ausgebremst wird. Um einem Defekt des Programmers vorzubeugen, sollten Sie folgende Dinge beachten:

- Schließen Sie immer zuerst die GND-Leitung (PIN10 des Programmieranschlusses) mit der Schaltung an, in der sich Ihr zu programmierender Microcontroller befindet. So findet ein Potentialausgleich zwischen Programmer und Schaltung statt.
- Schließen Sie keine Ausgangssignale am Programmieranschluss (z.B. PIN1, PIN3, PIN5, PIN7) direkt an GND oder an die Stromversorgung an. Die Ausgangstreiber können zerstört werden.
- Achten Sie darauf, dass die Pins am Programmieranschluss mit keiner höheren als der eingestellten Spannung am DIP-Schalter 1 in Verbindung kommen.

# EXA-PROG

USB-Programmieradapter für AVR-ISP, AVR-UPDI, STM32, LPC, ESP8266, ESP32

## LEUCHTDIODEN

<b>LED Grün</b>	Leuchtet konstant, wenn der eingestellte Betriebsmodus bereit ist. Erlischt kurz bei Druck auf den RESET-Taster.
<b>LED Rot</b>	Flackert bei Datenübertragungen über RXD/TXD und während der Programmierung. Flackert kurz, wenn der Programmier-Modus per DIP-Schalter gewechselt wird.

## RESET-TASTER

Der Taster auf dem EXA-PROG löst einen manuellen Reset des angeschlossenen Microcontrollers aus. Dies wird häufig benötigt, wenn ein Startvorgang (Boot) eines frisch programmierten Controllers mehrfach getestet werden soll. Die Funktion des Tasters ist abhängig vom gewählten Programmier-Modus.

### AVR-ISP, AVR-UPDI:

PIN5 (RESET) wird für die Dauer des Tastendrucks auf LOW-Pegel gelegt.

Die grüne LED erlischt so lange, wie der Taster gedrückt ist.

### STM32:

Ein kurzer Tastendruck (< 1 Sekunde) löst einen RESET des angeschlossenen Controllers aus. An PIN5 (RESET) wird nach Loslassen des Tasters ein 50ms langer Reset-Impuls erzeugt. PIN3 (BOOT) bleibt auf LOW.

Ein langer Tastendruck (>1 Sekunde) versetzt den angeschlossenen Microcontroller in den Boot-Modus. PIN3 (BOOT) wird auf HIGH-Pegel gelegt, ein 50ms langer Reset-Impuls wird erzeugt und danach geht PIN3 (BOOT) wieder auf LOW.

Die grüne LED flackert kurz als Zeichen, dass der Reset ausgelöst wurde.

### NXP/LPC:

Ein kurzer Tastendruck (< 1 Sekunde) löst einen RESET des angeschlossenen Controllers aus. An PIN5 (RESET) wird nach Loslassen des Tasters ein 50ms langer LOW-Impuls erzeugt. PIN3 (BOOT) bleibt auf HIGH.

Ein langer Tastendruck (>1 Sekunde) versetzt den angeschlossenen Microcontroller in den Boot-Modus. PIN3 (BOOT) wird auf LOW-Pegel gelegt, ein 50ms langer Reset-Impuls wird erzeugt und danach geht PIN3 (BOOT) wieder auf HIGH.

Die grüne LED flackert kurz als Zeichen, dass der Reset ausgelöst wurde.

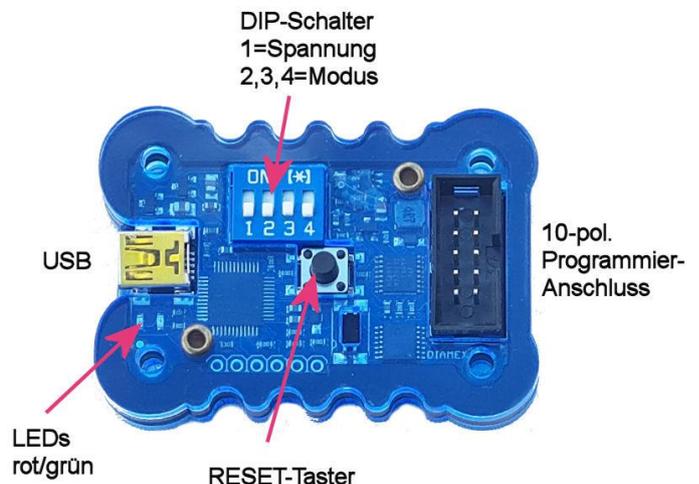
### ESP:

Ein kurzer Tastendruck (< 1 Sekunde) löst einen RESET des angeschlossenen Controllers aus. An PIN5 (RESET) wird nach Loslassen des Tasters ein 50ms langer LOW-Impuls erzeugt. PIN3 (GPIO0) bleibt auf HIGH.

Ein langer Tastendruck (>1 Sekunde) versetzt den angeschlossenen Microcontroller in den Boot-Modus. PIN3 (GPIO0) wird auf LOW-Pegel gelegt, ein 50ms langer Reset-Impuls wird erzeugt und danach geht PIN3 (GPIO0) wieder auf HIGH.

Die grüne LED flackert kurz als Zeichen, dass der Reset ausgelöst wurde.

⇒ Drücken Sie niemals den RESET-Taster während eines Programmiervorganges. Die Programmierung wird abgebrochen und die Daten im Flash-Speicher den Microcontrollers können ungültig sein.

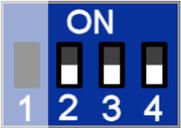
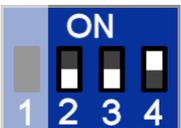
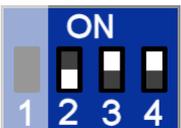
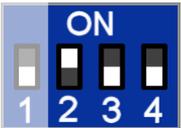
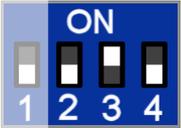


# EXA-PROG

USB-Programmieradapter für AVR-ISP, AVR-UPDI, STM32, LPC, ESP8266, ESP32

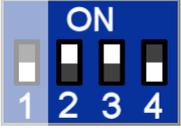
## DIP-SCHALTER

### SCHALTER 2, 3, 4: Betriebsmodus

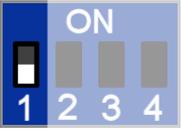
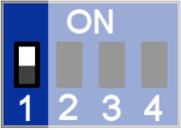
AVR-ISP	
	<p>2 = OFF, 3 = OFF, 4 = OFF</p> <p><b>Programmer für Atmel-AVR-Controller mit ISP-Schnittstelle</b></p> <p>Die Programmierung dieser Controller geschieht über die ISP-Schnittstelle, diese ist bei den meisten AVR-Controllern gleichbedeutend mit den Pins für den SPI-Bus (Achtung! Es gibt einige Controller, die separate PDI/PDO-Pins besitzen). Zusätzlich zu SCK, MISO, MOSI wird noch die Reset-Leitung benötigt. EXA-PROG emuliert einen STK500-Programmer und ist hierdurch kompatibel zu ATMEL/MICROCHIP-Studio und AVRDUDE. Die Programmierspannung kann je nach angeschlossenem AVR-Controller auf 3,3V oder 5V eingestellt werden.</p>
AVR-UPDI	
	<p>2 = OFF, 3 = OFF, 4 = ON</p> <p><b>Programmer für Atmel-AVR-Controller mit UPDI-Schnittstelle</b></p> <p>Die Programmierung dieser Controller geschieht über die UPDI-Schnittstelle. EXA-PROG emuliert einen jtag2updi-Programmer und ist derzeit nur kompatibel zu AVRDUDE. Die Programmierspannung kann je nach angeschlossenem AVR-Controller auf 3,3V oder 5V eingestellt werden.</p>
	<p>2 = OFF, 3 = ON, 4 = ON</p> <p><b>Programmer für Atmel-AVR-Controller mit UPDI-Schnittstelle (HV-Modus)</b></p> <p>Modus zum Reaktivieren der UPDI-Schnittstelle, wenn bei ATTiny-Controllern die Funktion des UPDI-Pins in den Fuses geändert wurde. Bitte nicht bei Controllern mit dediziertem UPDI-Pin (z.B. AVR32/64/128DA) verwenden.</p>
STM32	
	<p>2 = ON, 3 = OFF, 4 = OFF</p> <p><b>Programmer für STM32 Cortex-M Controller mit UART-Bootloader</b></p> <p>Die Programmierung der STM32-Controller geschieht über den integrierten Bootloader durch eine der seriellen Schnittstellen (siehe Datenblatt des zu programmierenden Controllers). Anschluss findet der Programmer an den Controller über die RX und TX-Anschlüsse. Zur automatischen Aktivierung des integrierten Bootloaders der Controller werden zusätzlich die Anschlüsse BOOT0 und RESET benötigt, hierdurch sind keinerlei Jumper oder Schalter an der Schaltung erforderlich. <b>Bitte DIP1 = OFF, Programmierspannung auf 3,3V stellen!</b></p>
NXP/LPC	
	<p>2 = OFF, 3 = ON, 4 = OFF</p> <p><b>Programmer für NXP/LPC Cortex-M Controller mit UART-Bootloader</b></p> <p>Die Programmierung der LPC-Controller geschieht über die integrierten Bootloader durch die serielle Schnittstelle UART0. Anschluss findet der Programmer an den Controller über die RX und TX-Anschlüsse. Zur automatischen Aktivierung des integrierten Bootloaders der Controller werden zusätzlich die Anschlüsse P2[10] und RESET benötigt, hierdurch sind keinerlei Jumper oder Schalter an der Schaltung erforderlich. EXA-PROG in Verbindung mit dem Tool „Flash-Magic“ schaltet den Controller automatisch in den Bootloader-Modus, programmiert den Flash-Speicher und startet danach das soeben geladene Anwenderprogramm. <b>Bitte DIP1 = OFF, Programmierspannung auf 3,3V stellen!</b></p>

# EXA-PROG

USB-Programmieradapter für AVR-ISP, AVR-UPDI, STM32, LPC, ESP8266, ESP32

ESP	
	<p>2 = ON, 3 = ON, 4 = OFF</p> <p><b>Programmer für ESP8266 und ESP32 mit ESPTOOL</b></p> <p>Die Programmierung des ESP8266 und ESP32 geschieht über den integrierten Bootloader durch die Leitungen TXD0 und RXD0. Zur automatischen Aktivierung des Bootloaders werden zusätzlich die Leitungen GPIO0 und RESET (EN) benötigt.</p> <p><b>Bitte DIP1 = OFF, Programmierspannung auf 3,3V stellen!</b></p>

## SCHALTER 1: Spannungspegel

3,3V	
	<p>1 = OFF</p> <p>An PIN 2 des 10-poligen Programmieranschlusses liegt eine Spannung von 3,3V an. Alle Signalpegel auf den Programmierleitungen haben eine maximale Spannung von 3,3V.</p>
5V	
	<p>1 = ON</p> <p>An PIN 2 des 10-poligen Programmieranschlusses liegt eine Spannung von ca. 5V an. Alle Signalpegel auf den Programmierleitungen haben eine maximale Spannung von ca. 5V.</p> <p>Beachten Sie bitte, dass der genaue Spannungswert abweichen kann und abhängig von der Spannung am USB-Anschluss ist. Dieser kann je nach Belastung schwanken und zwischen 4,5V und 5,2V betragen. Dies ist aber in der Regel bei der Programmierung von Microcontrollern unerheblich.</p>

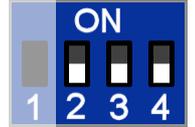
### Hinweise:

- Die maximale Belastung am Programmieranschluss ist 500mA bei 5 Volt und 300mA bei 3,3 Volt. Wird ein größerer Strom benötigt, bitte die angeschlossene Schaltung mit externer Spannung versorgen und PIN2 am Programmieranschluss nicht beschalten.
  - Die externe Schaltung muss immer mit derselben Spannung betrieben werden, wie die Spannung, die am Programmer mit DIP-Schalter 1 eingestellt ist.
  - GND muss immer mit der Schaltung verbunden werden, in der sich der zu programmierende Controller befindet.
- ⇒ **Achten Sie bitte darauf, dass nahezu alle 32-Bit Microcontroller nur mit 3,3V betrieben werden dürfen und hierzu DIP-Schalter 1 auf OFF gestellt werden muss. Die Controller können zerstört werden, wenn sie mit 5V betrieben werden.**

# EXA-PROG

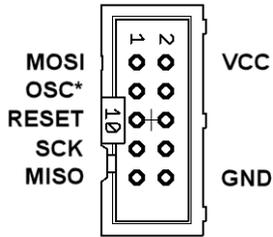
USB-Programmieradapter für AVR-ISP, AVR-UPDI, STM32, LPC, ESP8266, ESP32

## Anschluss eines AVR-Controllers mit ISP-Schnittstelle



☞ **DIP-SCHALTER: 2 = OFF, 3 = OFF, 4 = OFF**

Verbinden Sie die Leitungen des Programmieranschlusses auf dem EXA-PROG direkt mit den Pins am Controller:



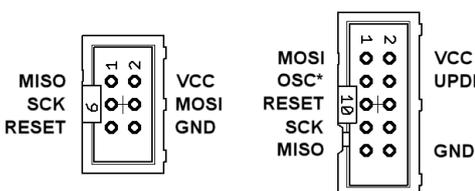
10-pol. Stiftleiste	AVR-Controller
PIN1 (MOSI)	MOSI oder PDI
PIN5 (RESET)	RESET
PIN7 (SCK)	SCK
PIN9 (MISO)	MISO oder PDO
Optional: PIN3 (OSC)	XTAL1 (XTALIN)
PIN10 (GND)	GND
Optional: PIN2 (VCC)	VCC

Die Pin-Nummern der AVR-Controller variieren abhängig vom Typ und der Gehäuseform. Schauen Sie bitte ins Datenblatt um die zu Ihrem Controller passenden Pin-Nummern zu ermitteln.

DIP-Schalter 1: OFF = 3,3V, ON = 5V. Stellen Sie hier unbedingt dieselbe Spannung ein, mit der Ihre Schaltung betrieben wird. Wird der Microcontroller in einem externen Sockel programmiert, können beide Spannungswerte eingestellt werden.

### HINWEISE, TIPPS:

- Einige AVR-Controller (z.B. AT90CAN32/64/128, ATmega64/128) haben spezielle Programmierpins PDI, PDO. Bitte benutzen Sie diese Leitungen anstatt MISO/MOSI (siehe Datenblatt).
- EXA-PROG unterstützt die adaptive SPI-Bitratenanpassung. Eine zu hoch eingestellte SPI-Bitrate führt in der Regel dazu, dass der angeschlossene AVR-Controller nicht erkannt wird. Die Regel besagt, dass die SPI-Bitrate maximal ein Viertel der Taktrate des Controllers sein darf. Sollte mit der eingestellten Bitrate keine Verbindung hergestellt werden können, schaltet EXA-PROG automatisch auf niedrigere Bitraten um, bis der Controller antwortet.
- Wenn der angeschlossene AVR-Controller trotzdem nicht antworten will, kontrollieren Sie bitte zunächst die Leitungsverbindungen. MISO an MISO und MOSI an MOSI (das wird häufig vertauscht), SCK an SCK und RESET an RESET. Sind auch keine anderen Bauteile an den Programmierleitungen angeschlossen? Ein eventuell vorhandener Kondensator am RESET-Pin muss unbedingt entfernt werden. Hat der angeschlossene Microcontroller eine Stromversorgung? Eventuell fehlt dem AVR-Controller der Systemtakt, weil die Fuses auf externen Takt eingestellt sind. In diesem Fall kann ein Quarz zur Takterzeugung an die XTAL-Pins des Controllers angeschlossen werden oder das Taktsignal von PIN3 (OSC) der 10-poligen Stiftleiste auf XTALIN oder XTAL1 (siehe Datenblatt des Controllers) gelegt werden. Die Standard-Taktfrequenz an PIN3 des ISP-Anschlusses beträgt ca. 1,3 MHz.
- EXA-PROG unterstützt nicht ATXmega-Controller mit PDI-Schnittstelle und ATTiny-Controller mit TPI-Schnittstelle.
- Verbinden Sie nur PIN2 mit der zu programmierenden Schaltung, wenn diese nicht über eine eigene Stromversorgung verfügt und die zu erwartende Stromaufnahme nicht die maximale Grenze des EXA-PROG überschreitet.
- Für Boards mit 6-poligem ISP-Anschluss benutzen Sie bitte den optional erhältlichen Adapter oder verbinden Ihre Schaltung mit passendem Jumperkabeln mit dem 10-poligen ISP-Anschluss.



Adapter 10-pol. auf 6-pol. ISP

(Sonderzubehör, nicht im Lieferumfang des EXA-PROG enthalten)

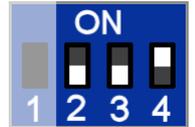


# EXA-PROG

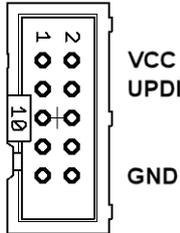
USB-Programmieradapter für AVR-ISP, AVR-UPDI, STM32, LPC, ESP8266, ESP32

## Anschluss eines AVR-Controllers mit UPDI-Schnittstelle

☞ **DIP-SCHALTER: 2 = OFF, 3 = OFF, 4 = ON**



Verbinden Sie die Leitungen des Programmieranschlusses auf dem EXA-PROG direkt mit den Pins am Controller:



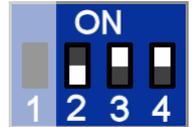
10-pol. Stiftleiste	AVR-Controller
PIN4 (UPDI)	UPDI (RESET, PORT)
PIN10 (GND)	GND
Optional: PIN2 (VCC)	VCC

Die Pin-Nummern der AVR-Controller variieren abhängig vom Typ und der Gehäuseform. Schauen Sie bitte ins Datenblatt um die zu Ihrem Controller passenden Pin-Nummern zu ermitteln.

DIP-Schalter 1: OFF = 3,3V, ON = 5V. Stellen Sie hier unbedingt dieselbe Spannung ein, mit der Ihre Schaltung betrieben wird. Wird der Microcontroller in einem externen Sockel programmiert, können beide Spannungswerte eingestellt werden.

## UPDI-HV-MODUS

☞ **DIP-SCHALTER: 2 = OFF, 3 = ON, 4 = ON**

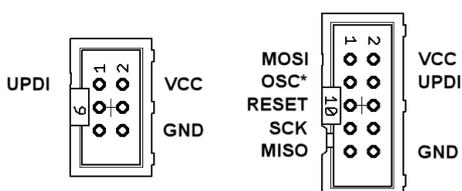


Bei ATTiny-Controllern, die nicht über einen separaten UPDI-Pin verfügen (z.B. Tiny412, Tiny1614, Tiny3216), kann der Pin über die Fuses auf eine alternative Funktion umgeschaltet werden. Danach ist keine Programmierung im Standard-UPDI-Modus mehr möglich. Wählen Sie in diesem Fall den UPDI-Hochvolt-Modus, in dem der UPDI-Anschluss des Controllers über einen kurzen 12 Volt Impuls für die Programmierung reaktiviert wird. Wenn der Controller in einer Schaltung programmiert werden soll, achten Sie unbedingt darauf, dass sich keine Bauteile am UPDI-Pin befinden dürfen.

⇒ Bitte diesen Modus nicht für Controller mit dediziertem UPDI-Pin (z.B. AVR32/64/128DA) verwenden.

### HINWEISE, TIPPS:

- Verbinden Sie nur PIN2 mit der zu programmierenden Schaltung, wenn diese nicht über eine eigene Stromversorgung verfügt und die zu erwartende Stromaufnahme nicht die maximale Grenze des EXA-PROG überschreitet.
- Die Software-Schnittstelle des EXA-PROG im UPDI-Modus emuliert ein jtag2updi-Interface. Dieses wird vom AT-MEL/MICROCHIP-Studio nicht unterstützt. Verwenden Sie stattdessen zur Programmierung das Tool AVRDUDE. Dieses wird in verschiedenen Entwicklungsumgebungen wie Arduino oder Bascom eingesetzt. Viele Infos hierzu finden Sie im Internet und über den Link am Ende dieser Anleitung.
- Für Boards mit 6-poligem UPDI-Anschluss benutzen Sie bitte den optional erhältlichen Adapter oder verbinden Ihre Schaltung mit passendem Jumperkabeln mit dem 10-poligen ISP-Anschluss.



Adapter 10-pol. auf 6-pol. UPDI  
(Sonderzubehör, nicht im Lieferumfang des EXA-PROG enthalten)

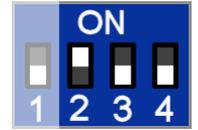


# EXA-PROG

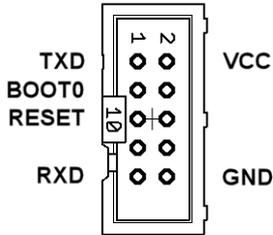
USB-Programmieradapter für AVR-ISP, AVR-UPDI, STM32, LPC, ESP8266, ESP32

## Anschluss eines STM32-Controllers

☞ **DIP-SCHALTER: 2 = ON, 3 = OFF, 4 = OFF**



Verbinden Sie die Leitungen des Programmieranschlusses auf dem EXA-PROG direkt mit den Pins am Controller:



10-pol. Stiftleiste	STM32-Controller	Control
PIN1 (TXD)	USART1-RX (PA10)	
PIN3 (BOOT)	BOOT0	DTR
PIN5 (RESET)	RESET	!RTS
PIN9 (RXD)	USART1-TX (PA9)	
PIN10 (GND)	GND	
Optional: PIN2 (VCC)	VCC	

Die Pin-Nummern des STM32 variieren abhängig vom Typ und der Gehäuseform. Schauen Sie bitte ins Datenblatt des STM32 um die zu Ihrem Controller passenden Pin-Nummern zu ermitteln.

DIP-Schalter 1: Muss auf OFF = 3,3V stehen.

### HINWEISE, TIPPS:

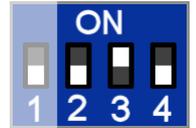
- Damit der Bootloader gestartet werden kann, muss dafür gesorgt werden, dass BOOT1 (PB2) während des RESET auf Massepotential liegt (bei neueren Controllertypen nicht mehr vorhanden). Wenn der Pin nicht anderweitig benutzt wird, können Sie ihn direkt auf GND legen. Ansonsten legen Sie ihn bitte über einen Widerstand 10 kOhm auf GND.
- Zur Programmierung des STM32 muss dieser mit 3,3 Volt versorgt werden, wenn die Stromaufnahme nicht höher als 300mA ist, kann der Controller inklusive Schaltung auch aus dem EXA-PROG versorgt werden.
- Für die Programmierung muss kein Quarz an den STM32 angeschlossen sein, der Bootloader des STM32 benutzt den internen RC-Oszillator (HSI-Clock) zur Erzeugung des Systemtaktes.
- Die Steuerung der Leitungen BOOT0 und RESET funktioniert mit dem Programmierwerkzeug „STM Flash-Loader“ leider nicht. Schließen Sie die Leitungen BOOT0 und RESET nicht an. In diesem Fall müssen Sie sich um die Aktivierung des STM32-Bootloader selber kümmern (BOOT0 auf High, danach RESET kurz auf Low. Dies ist auch über den Reset-Taster auf dem EXA-PROG möglich).
- Wenn Sie Ihr eigenes Programmierwerkzeug entwickeln, können die Leitungen BOOT und RESET über die Signale DTR (BOOT) und RTS (!RESET) gesteuert werden.
- STM8 Controller werden von diesem Programmierer nicht unterstützt.

# EXA-PROG

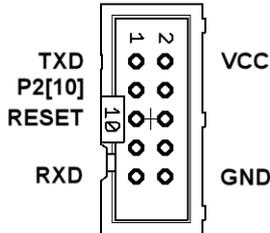
USB-Programmieradapter für AVR-ISP, AVR-UPDI, STM32, LPC, ESP8266, ESP32

## Anschluss eines NXP/LPC-Controllers

☞ **DIP-SCHALTER: 2 = OFF, 3 = ON, 4 = OFF**



Verbinden Sie die Leitungen des Programmieranschlusses auf dem EXA-PROG direkt mit den Pins am Controller:



10-pol. Stiftleiste	NXP-Controller	Control
PIN1 (TXD)	UART0-RXD P0[2]	
PIN3 (BOOT)	P2[10]	!RTS
PIN5 (RESET)	RESET	!DTR
PIN9 (RXD)	UART0-TXD P0[3]	
PIN10 (GND)	GND	
Optional: PIN2 (VCC)	VCC	

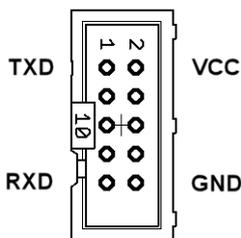
Die Pin-Nummern des NXP-Controllers variieren abhängig von der Gehäuseform. Schauen Sie bitte ins Datenblatt um die zu Ihrem Controller passenden Pin-Nummern zu ermitteln.

DIP-Schalter 1: Muss auf OFF = 3,3V stehen.

### HINWEISE, TIPPS:

- Zur Programmierung des NXP-Controllers muss dieser mit 3,3 Volt versorgt werden.
- Für die Programmierung muss kein Quarz an den NXP-Controller angeschlossen sein, der Bootloader des Controllers benutzt den internen RC-Oszillator zur Erzeugung des Systemtaktes.
- Die Steuerung der Leitungen BOOT und RESET funktioniert mit dem Tool „Flash Magic“ einwandfrei.
- Wenn Sie Ihr eigenes Programmierwerkzeug entwickeln, können die Leitungen BOOT und RESET über die Signale DTR (!BOOT) und RTS (!RESET) gesteuert werden. Dies ist auch manuell über den Reset-Taster auf dem EXA-PROG möglich.
- 8-Bit-Controller (LPC9xx) und NXP-Controller ohne ARM-Cortex-Kern werden von diesem Programmierer nicht unterstützt.

## EXA-PROG als USB/Seriell-Wandler



EXA-PROG kann in den Modi STM32, NXP/LPC und ESP32 als USB/Seriell-Wandler genutzt werden.

Die maximale Übertragungsrate beträgt 1 MBit.

Als Übertragungsparameter werden unterstützt: 7 + 8 Bit, Parity OFF, EVEN, ODD, 1 oder 2 Stoppbits.

TXD (PIN1) wird mit dem RX-Pin und RXD (PIN9) wird mit dem TX-Pin des Microcontrollers verbunden.

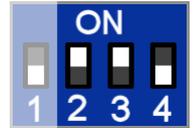
DIP-Schalter 1: OFF = 3,3V, ON = 5V. Stellen Sie hier unbedingt dieselbe Spannung ein, mit der Ihre Schaltung betrieben wird.

# EXA-PROG

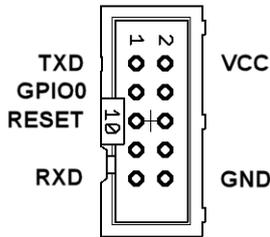
USB-Programmieradapter für AVR-ISP, AVR-UPDI, STM32, LPC, ESP8266, ESP32

## Anschluss eines ESP-Controllers

☞ DIP-SCHALTER: 2 = ON, 3 = ON, 4 = OFF



Verbinden Sie die Leitungen des Programmieranschlusses auf dem EXA-PROG direkt mit den Pins am Controller bzw. an einem ESP8266 oder ESP32-Modul:



10-pol. Stiftleiste	ESP8266	ESP32
PIN1 (TXD)	RXD0	RXD0
PIN3 (GPIO0)	CH_PD (GPIO0)	GPIO0
PIN5 (RESET)	RST	EN
PIN9 (RXD)	TXD0	TXD0
PIN10 (GND)	GND	GND
Optional: PIN2 (VCC)	VCC	VCC

DIP-Schalter 1: Muss auf OFF = 3,3V stehen.

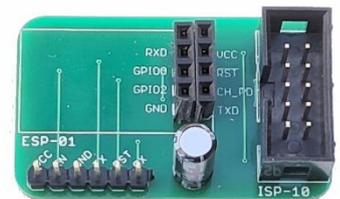
### HINWEISE, TIPPS:

- Der RST-Pin des ESP8266 bzw. EN-Pin des ESP32 sollte mit einem 10 kOhm Widerstand gegen +3,3V gelegt werden. Ein Kondensator nach Masse (wie in vielen Schaltungen empfohlen) darf sich an diesem Pin nicht befinden, da der RESET vom EXA-PROG gesteuert wird.
- Zur Programmierung des ESP8266 oder ESP32 muss dieser mit 3,3 Volt versorgt werden. Wenn die Stromaufnahme nicht höher als 300mA ist, kann der Controller inklusive Schaltung auch aus dem EXA-PROG versorgt werden.
- Bei der Benutzung von ESPTOOL werden die Steuerleitungen GPIO0 und RESET automatisch angesteuert. Eine manuelle Aktivierung des Bootloaders ist dadurch nicht erforderlich, kann jedoch auch über den Reset-Taster auf dem EXA-PROG ausgelöst werden, wenn dies nicht per Software funktioniert.

### Adapter 10-pol.nach ESP

Buchsenleiste für ESP-01-Module, Stiftleisten zum Anschluss von ESP8266 oder ESP32-Controllern über Jumperkabel.

(Sonderzubehör, nicht im Lieferumfang des EXA-PROG enthalten)



# EXA-PROG

USB-Programmieradapter für AVR-ISP, AVR-UPDI, STM32, LPC, ESP8266, ESP32

---

## HINWEISE

© Erwin Reuß; Folker Stange. Nutzung und Weitergabe dieser Informationen auch Auszugsweise nur mit Erlaubnis der Copyright-Inhaber. Alle Markennamen, Warenzeichen und eingetragenen Warenzeichen sind Eigentum Ihrer rechtmäßigen Eigentümer und dienen hier nur der Beschreibung.

## HAFTUNGSHINWEIS

Der Hersteller übernimmt keine Haftung für Schäden die durch Anwendung des EXA-PROG entstehen könnten.

## LINKS

LED-Genial Online-Shop  
<http://www.led-genial.de>

Diamex Online-Shop  
<http://www.diamex.de>

AVR/ATMEL/MICROCHIP-Studio Download  
[http://www.mikrocontroller.net/articles/Atmel\\_Studio](http://www.mikrocontroller.net/articles/Atmel_Studio)

UPDI-Programmierung mit AVRDUDE  
<https://github.com/EITangas/jtag2updi#using-with-avrdude>

ESPTOOL für ESP8266 und ESP32  
<https://github.com/themadinventor/esptool>

### Vertrieb



#### DIAMEX Produktion und Handel GmbH

Innovationspark Wuhlheide  
Köpenicker Straße 325, Haus 41  
12555 Berlin

Telefon: 030-65762631

E-Mail: [info@diamex.de](mailto:info@diamex.de)  
Homepage: <http://www.diamex.de>

### Herstellung



[www.tremex.de](http://www.tremex.de)

Köpenicker Str. 325 12555 Berlin  
Tel. 030-65762631

Hersteller: Tremex GmbH  
DIAMEX × OBD-DIAG × TREMEX  
WEE-Reg.Nr. DE 51673403