

## PROG-S2E

### Folgende Microcontroller können mit PROG-S2E programmiert werden:

- ATMEL/Microchip AVR, MegaAVR, TinyAVR-Controller mit ISP-Schnittstelle (MISO, MOSI, SCK, RESET) oder alternativ (PDI, PDO, SCK, RESET).
- 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-C3/S2/S3 Controller und Module über den internen UART-Bootloader.

### Fähigkeiten:

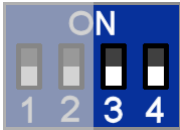
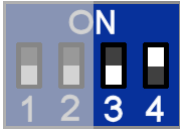
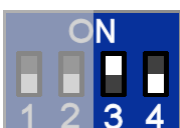
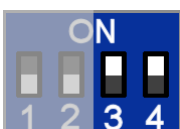
- Funktionswahl über DIP-Schalter.
- Signalpegel 3,3V und 5V umschaltbar.
- Stromversorgung für externe Schaltung bis maximal 300mA.
- Taktgenerator für AVR-Controller mit ver“fuse“tem Oszillator.
- 6- und 10-poliger Standard-ISP-Anschluss.
- Mini-USB-Anschluss zur Stromversorgung und Verbindung mit dem PC.
- Firmware updatefähig über USB.

### Unterstützte Software:

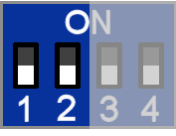
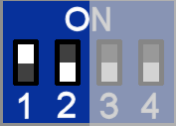
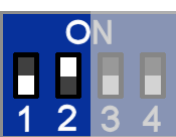

- AVR/Microchip-Studio ab v7.x (nur AVR-ISP: STK500).
- AVRDUDE ab v6.3 (AVR-ISP: STK500).
- Entwicklungsumgebungen, die auf AVRDUDE als Programmierinterface basieren, z.B. Arduino, Bascom.
- ESPTOOL (esptool.py, esptool.exe) und alle Programme, die ESPTOOL als Programmierinterface benutzen.

## DIP-SCHALTER

### SCHALTER 3 + 4: Betriebsmodus

	<p><b>3 = OFF, 4 = OFF</b></p> <p><b>Programmer für alle 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. PROG-S2 emuliert einen STK500-Programmer und ist hierdurch kompatibel zu AVR/ATMEL-Studio und AVRDUDE.</p> <p>Die Programmierspannung kann je nach angeschlossenem AVR-Controller auf 3,3V oder 5V eingestellt werden.</p>
	<p><b>3 = OFF, 4 = ON</b></p> <p><b>Programmer für alle STM32 Cortex-M Controller mit UART-Bootloader</b></p> <p>Die Programmierung der STM32-Controller geschieht über die 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.</p> <p>Bitte die Programmierspannung auf 3,3V stellen!</p>
	<p><b>3 = ON, 4 = OFF</b></p> <p><b>Programmer für alle 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. PROG-S2 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.</p> <p>Bitte die Programmierspannung auf 3,3V stellen!</p>
	<p><b>3 = ON, 4 = ON</b></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 CH_PD (GPIO0) und RESET (EN) benötigt. Es sind keine Jumper, Schalter oder Taster auf der Schaltung erforderlich.</p> <p>Bitte die Programmierspannung auf 3,3V stellen!</p>

## SCHALTER 1 + 2: Stromversorgung extern und Spannungspegel

 	<p><b>1 = ON/OFF, 2 = OFF (externe Spannung aus)</b></p> <p>An PIN 2 des 6- und 10-poligen Programmieranschlusses liegt keine Spannung an. Die Versorgung des angeschlossenen Microcontrollers muss über eine externe Stromversorgung geschehen.</p> <p><b>Bitte unbedingt die externe Spannung an PIN2 (max. 5V) anlegen!</b> <b>Die Höhe der externen Spannung bestimmt den Pegel auf den Programmierleitungen!</b></p>
	<p><b>1 = OFF (3,3V), 2 = ON (externe Spannung ein)</b></p> <p>Höhe der Spannung auf den Datenleitungen und extern = 3,3 Volt</p> <p>Eine externe Schaltung bzw. ein angeschlossener Controller kann vom PROG-S2E mit Strom versorgt werden.</p>
	<p><b>1 = ON (5V), 2 = ON (externe Spannung ein)</b></p> <p>Höhe der Spannung auf den Datenleitungen und extern = 5 Volt</p> <p>Eine externe Schaltung bzw. ein angeschlossener Controller kann vom PROG-S2E mit Strom versorgt werden.</p>

### Hinweise:

- Die Höhe der externen Spannung bei Einstellung 5 Volt ist abhängig von der Spannung an der USB-Buchse, sie kann zwischen 4,3V und 5,3V liegen (abhängig vom PC oder USB-HUB).
- Die maximale Belastung am Programmieranschluss ist 500mA bei 5 Volt bzw. 100mA bei 3,3 Volt. Wird ein größerer Strom benötigt, bitte die angeschlossene Schaltung mit externer Spannung versorgen und DIP-Schalter 2 in Stellung OFF.
- Wenn sich DIP-Schalter 2 in Stellung ON befindet, bitte keine externe Spannung an den Programmieranschluss anlegen.
- Wenn sich DIP-Schalter 2 in Stellung OFF befindet, muss die externe Spannung (max. 5V) an den Programmieranschluss (PIN2) angelegt werden um die Höhe der Signalpegel auf den Programmierleitungen festzulegen.
- Wenn PIN2 des 6- oder 10-poligen Programmieranschlusses unbeschaltet ist, muss Schalter 2 auf ON stehen.
- Bitte unbedingt immer GND-Signal zwischen Programmer und Microcontroller verbinden.

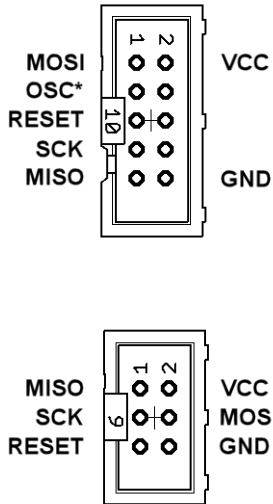
## LEUCHTDIODEN

<b>LED Grün</b>	Leuchtet konstant, wenn eine USB-Verbindung zum PC besteht.
<b>LED Rot</b>	Flackert bei Datenübertragungen über RXD/TXD und beim Programmieren von Microcontrollern.

## Anschluss eines AVR-Controllers

☞ **DIP-SCHALTER: 3 = OFF, 4 = OFF**

Verbinden Sie die Leitungen des Programmieranschlusses auf dem PROG-S2E direkt mit den Pins am Controller:



10-pol. Stiftleiste	6-pol. Stiftleiste	AVR-Controller
PIN1 (MOSI)	PIN4 (MOSI)	MOSI oder PDI
PIN5 (RESET)	PIN5 (RESET)	RESET
PIN7 (SCK)	PIN3 (SCK)	SCK
PIN9 (MISO)	PIN1 (MISO)	MISO oder PDO
Optional: PIN3 (OSC)		XTAL1 (XTALIN)
PIN10 (GND)	PIN6 (GND)	GND
Optional: PIN2 (VCC)	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.

### 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).

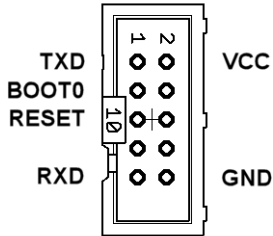
PROG-S2E unterstützt die adaptive SPI-Bitratenanpassung. Eine zu hoch eingestellte SPI-Bitrate führt in der Regel dazu, dass der angeschlossene AVR-Controller nicht erkannt wird. Die Regel besagt, dass die SPI-Bitrate viermal so hoch sein muss, wie die Taktrate des Controllers. Sollte mit der eingestellten Bitrate keine Verbindung hergestellt werden können, schaltet PROG-S2E 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? 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 MHz.
- PROG-S2E kann nicht Controller mit PDI-, TPI- oder UPDI-Schnittstelle programmieren.
- 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 PROG-S2E überschreitet.

## Anschluss eines STM32-Controllers

☞ **DIP-SCHALTER: 3 = OFF, 4 = ON**

Verbinden Sie die Leitungen des Programmieranschlusses auf dem PROG-S2E 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.

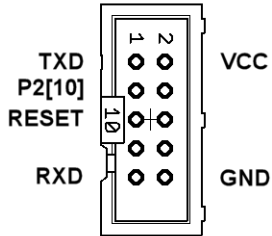
Bitte die Programmierspannung auf 3,3V stellen!

### 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..100 kOhm auf GND.
- Zur Programmierung des STM32 muss dieser mit 3,3 Volt versorgt werden, wenn die Stromaufnahme nicht höher als 100mA ist, kann der Controller inklusive Schaltung auch aus dem PROG-S3E 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).
- 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.

## Anschluss eines NXP/LPC-Controllers

☞ **DIP-SCHALTER: 3 = ON, 4 = OFF**



Verbinden Sie die Leitungen des Programmieranschlusses auf dem PROG-S2E 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 in das Datenblatt um die zu Ihrem Controller passenden Pin-Nummern zu ermitteln.

Bitte die Programmierspannung auf 3,3V stellen!

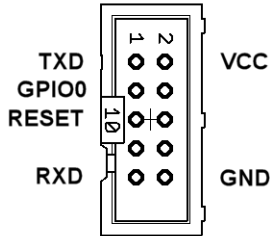
### HINWEISE, TIPPS:

- 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.
- 8-Bit-Controller (LPC9xx) und NXP-Controller ohne ARM-Cortex-Kern werden von diesem Programmierer nicht unterstützt.

## Anschluss eines ESP-Controllers

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

Verbinden Sie die Leitungen des Programmieranschlusses auf dem PROG-S2E direkt mit den Pins am Controller bzw. an einem ESP8266 oder ESP32-Modul:



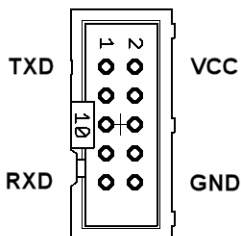
10-pol. Stiftleiste	ESP8266	ESP32
PIN1 (TXD)	RXD0	RXD0
PIN3 (GPIO0)	CH_PD (GPIO0)	GPIO0
PIN5 (RESET)	RST	EN
PIN9 (RXD)	TXD0	TXD0
PIN10 (GND)	GND	GND
Optional: PIN2 (VCC)	VCC	VCC

Bitte die Programmierspannung auf 3,3V stellen!

### 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 PROG-S2E gesteuert wird.
- Zur Programmierung des ESP8266 oder ESP32 muss dieser mit 3,3 Volt versorgt werden. Wenn die Stromaufnahme nicht höher als 100mA ist, kann der Controller inklusive Schaltung auch aus dem PROG-S2E versorgt werden.
- Bei der Benutzung von ESPTOOL werden die Steuerleitungen GPIO0 und RESET automatisch angesteuert. Eine manuelle Aktivierung des Bootloaders ist dadurch nicht erforderlich.

## PROG-S2E als USB/Seriell-Wandler



PROG-S2E 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.

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

## 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 PROG-S2E entstehen könnten.

## LINKS

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

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

Flash-Magic für LPC-Controller:  
<http://www.flashmagictool.com>

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

## 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