



TECHNISCHE DATEN

- MIT KAPAZITIVEN TOUCH
- 4 SERIELLE SCHNITTSTELLEN USB, RS232, SPI, I²C
- 8 DIGITALE, FREI DEFINIERBARE I/Os EINGEBAUT
- 2 ANALOGE EINGÄNGE / 1 ANALOGER AUSGANG
- 8 EINGEBAUTE FONTS
- PIXELGENAUE POSITIONIERUNG BEI ALLEN FUNKTIONEN
- BILDSCHIRMSCHONER MODI
- BIS ZU 256 BILDER INTERN SPEICHERBAR
- BIS ZU 256 MAKROS PROGRAMMIERBAR
- TEXT UND GRAFIK MISCHEN, BLINKATTRIBUTE: EIN/AUS/INVERS
- VERÄNDERBARE DISPLAYHELLIGKEIT (SOFTWARE)

BESTELLBEZEICHNUNGEN

2,9" OLED MIT USB UND TOUCHPANEL
...MIT SCHAUBKLEMME UND STECKER
1,7" OLED MIT USB UND TOUCHPANEL
...MIT SCHRAUBKLEMME UND STECKER

EA PLUGL128-6GTC
EA PLUGL128-6GTCZ
EA PLUGS102-6GTC
EA PLUGS102-6GTCZ

ZUBEHÖR

USB KABEL MINI (~ 1m)
IDC KABEL 25cm 1 STECKER (PLUGL128-6)
... MIT 2 STECKERN
IDC KABEL 25cm 1 STECKER (PLUGS102-6)
... MIT 2 STECKERN
INDIVIDUELLE LÄNGEN AUF ANFRAGE

EA KUSB-MINI
EA KB-126
EA KB-226
EA KB-120
EA KB-220

Inhaltsverzeichnis

Revisionsübersicht	3
Allgemein	4
Hardware	5
Pinbelegung ZIF Stecker	6
Pinbelegung Schraubklemme	7
Pinbelegung Wannenstecker	8
Serielle Schnittstellen	9
USB	9
RS232	10
SPI	12
PC	13
I/O	14
PWM	17
Externer Lautsprecher	18
Software	20
Ebenen	21
Blinkmodus	23
Zeichensätze	24
Touch	31
Makro Programmierung	32
Protokoll/Datenübertragung	33
Befehlsübersicht	39
Terminalfenster	43
Displayfunktionen	44
Clipboard	45
Geraden-/Bereichsfunktionen	46
Textfunktionen	48
Bildfunktionen	49
Bargraphfunktionen	50
Blinkbereiche	52
Menübox/ Touchmenü	53
Makro	55
Allgemein	57
I/O/ Digital/ PWM	59
Analogeingang/ Analogausgang	60
Touch	62
Antworten/ Rückmeldungen	66
Befehlsbeispiele	68
Displayfunktionen	69
Geraden-/Bereichsfunktionen	70
Textfunktionen	71
Bildfunktionen	72
Bargraphfunktionen	74
Menübox/ Touchmenü	75
Touch	76
KitEditor	77
Elektrische Spezifikation	78
Maßzeichnung EA PLUGL128-6	80
Maßzeichnung EA PLUGS102-6	81

REVISIONSÜBERSICHT

EA PLUG-Serie Firmware

Date	Version	Info
06.02.2019	Version 1.1	Bugs: Formatted strings are displayed too large by one zoom level Partly missing reset of protocol send buffer length Internal: Change flash routine for testmacro
07.12.2018	Version 1.0	First release

Datenblatt

Date	Version	Info
05.2019		First release Adding connector and cable accessories

ALLGEMEIN

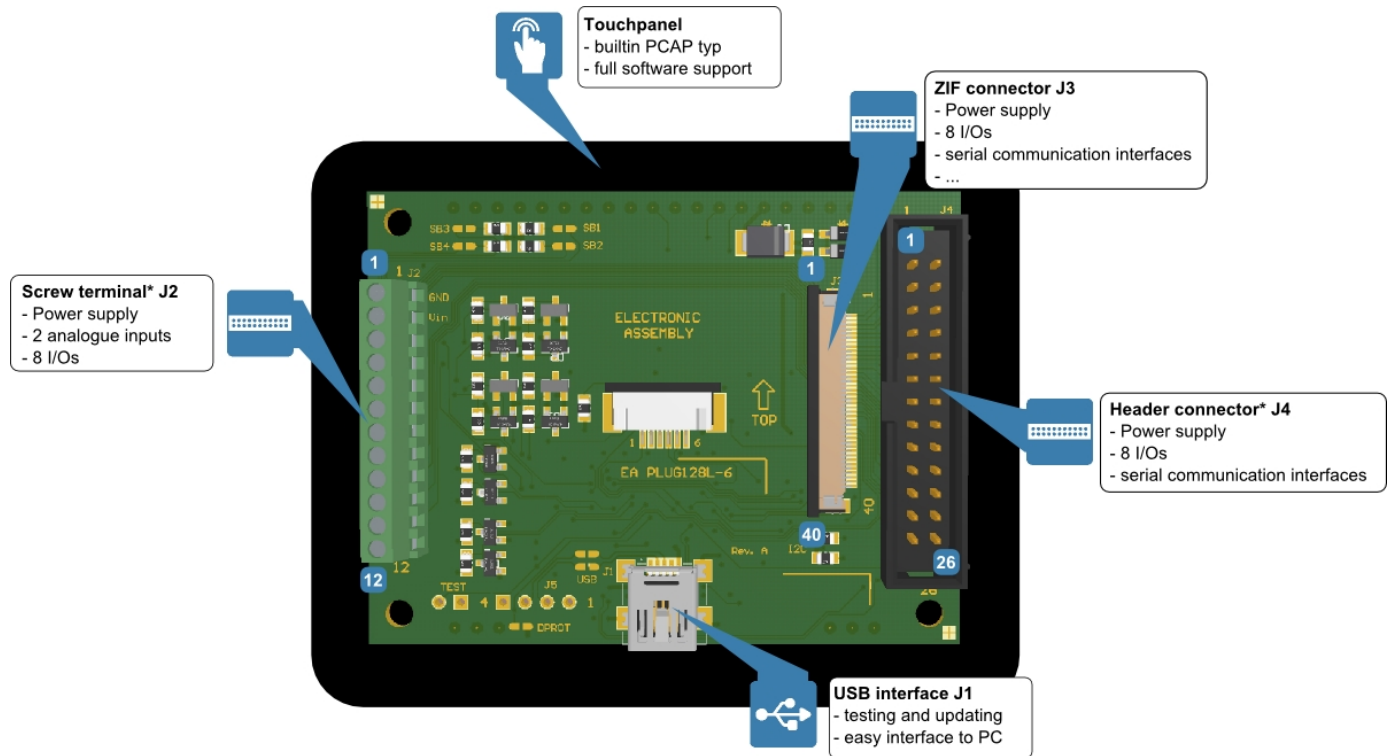
Die EA PLUG-Serie ermöglicht mit dem integrierten Befehlsatz eine einfache grafische Darstellung.

Neben diversen eingebauten Schriften, welche pixelgenau verwendet werden können, bietet es zudem eine ganze Reihe ausgefeilter Grafikfunktionen. Das Display ist mit 3,3V...5V sofort betriebsbereit. Die Ansteuerung erfolgt über eine der vier eingebauten seriellen Schnittstellen RS232, SPI, PC oder USB.

Die Programmierung erfolgt über hochsprachenähnliche Grafikbefehle; die zeitraubende Programmierung von Zeichensätzen und Grafikroutinen entfällt hier völlig. Die simple Verwendung dieses Displays samt Touchpanel verkürzt die Entwicklungszeit drastisch.

HARDWARE

Die EA PLUG-Serie besteht aus einem OLED Display und einem kapazitiven Touchpanel. Das Modul lässt sich schnell und einfach über eine USB Schnittstelle in Betrieb nehmen. Es ist für eine Betriebsspannung von 3,3 bis 5V ausgelegt. Die Datenübertragung erfolgt seriell im RS232 Format, per SPI, I²C oder direkt mit USB. Für kleine Steuerungsaufgaben stehen dem Modul zusätzlich 8 frei verwendbare I/Os, 2 Analoingänge, sowie ein PWM- und ein Analogausgang zur Verfügung.



*Stecker sind nur in der Hardwarevariante Z bestückt (z.B. EA PLUGL128G-6TCZ)

PINBELEGUNG ZIF STECKER (J3)

Pin	Symbol	I/O	Beschreibung	
1	GND		Ground 0V	
2	V _{in}		Spannungsversorgung 3.3~5V	
3	RES	I	Reset	interner Pull-Up (10kΩ), low active
4	CS	I	SPI: Chip Select	low active
5	MOSI	I	SPI: MOSI	
6	MISO	O	SPI: MISO	
7	CLK	I	SPI: Clock	
8	RxD	I	RS232: Receive Data	3,3V Pegel, interner Pull-Up (1MΩ)
9	TxD	O	RS232: Transmit Data	3,3V Pegel
10	DE	O	RS485: Transmit enable	3,3V Pegel
11	SDA	I/O	I ² C: Serial Data	interner Pull-Up: (4,7kΩ)
12	SCL	I	I ² C: Serial CLK	interner Pull-Up: (4,7kΩ)
13	SBUF Test	O I	Low: Sendepuffer enthält Daten PowerOn Low: Testmodus aktiviert	interner Pull-Up (10kΩ)
14	DAC	O	Analogausgang	0~3.1V
15	AIN1	I	Analogeingang 1	0~3.1V
16	AIN2	I	Analogeingang 2	0~3.1V
17	I/O 1	I/O	Digitaler Ein- oder Ausgang	High-Power Ausgang
18	I/O 2	I/O	Digitaler Ein- oder Ausgang	High-Power Ausgang
19	I/O 3	I/O	Digitaler Ein- oder Ausgang	High-Power Ausgang
20	I/O 4	I/O	Digitaler Ein- oder Ausgang	High-Power Ausgang
21	I/O 5	I/O	Digitaler Ein- oder Ausgang / PWM	Low-Power Ausgang
22	I/O 6	I/O	Digitaler Ein- oder Ausgang	Low-Power Ausgang
23	I/O 7	I/O	Digitaler Ein- oder Ausgang	Low-Power Ausgang
24	I/O 8	I/O	Digitaler Ein- oder Ausgang	Low-Power Ausgang
25	V _{out}		Spannungsausgang (max. 3.1V)	für externe Peripherie (max. 100mA)
26	GND		Ground 0V	
27	d.n.c		intern verbunden	
28	d.n.c		intern verbunden	
29	SPEAK	O	Lautsprecherausgang	
30	SPEAK POW	O	Lautsprecherausgang (Power)	
31	d.n.c		intern verbunden	
32	d.n.c		intern verbunden	
33	n.c.			
34	n.c.			
35	n.c.			
36	n.c.			
37	n.c.			
38	n.c.			
39	d.n.c		intern verbunden	
40	GND			

PINBELEGUNG SCHRAUBKLEMME¹⁾ (J2)

Pin	Symbol	I/O	Beschreibung
1	GND		Ground 0V
2	V _{in}		Spannungsversorgung 3.3~5V
3	AIN1	I	Analogeingang 1 0~3.1V
4	AIN2	I	Analogeingang 2 0~3.1V
5	I/O 1	I/O	Digitaler Ein- oder Ausgang High Power Ausgang
6	I/O 2	I/O	Digitaler Ein- oder Ausgang High Power Ausgang
7	I/O 3	I/O	Digitaler Ein- oder Ausgang High Power Ausgang
8	I/O 4	I/O	Digitaler Ein- oder Ausgang High Power Ausgang
9	I/O 5	I/O	Digitaler Ein- oder Ausgang / PWM Ausgang Low Power Ausgang
10	I/O 6	I/O	Digitaler Ein- oder Ausgang Low Power Ausgang
11	I/O 7	I/O	Digitaler Ein- oder Ausgang Low Power Ausgang
12	I/O 8	I/O	Digitaler Ein- oder Ausgang Low Power Ausgang

1) Nur in der Hardwarevariante Z bestückt (EA PLUGL128G-6TCZ oder EA PLUGS102-6GTCZ)

PINBELEGUNG WANNENSTECKER¹⁾ (J4)

Pin	Symbol	I/O	Beschreibung		
1	GND		Ground 0V		
2	V _{in}		Spannungsversorgung 3.3~5V		
3	$\overline{\text{RES}}$	I	Reset	interner Pull-Up (10k Ω), low aktiv	
4	$\overline{\text{CS}}$	I	SPI: Chip Select	low aktiv	
5	MOSI	I	SPI: MOSI		
6	MISO		SPI: MISO		
7	CLK	I	SPI: CLK		
8	RxD	I	RS232: Receive Data	3,3V Pegel, interner Pull-Up (1M Ω)	
9	TxD	O	RS232: Transmit Data	3,3V Pegel	
10	DE	O	RS485: Transmit enable	3,3V Pegel	
11	SDA	I/O	I ² C: Serial Data	interner Pull-Up: (4,7k Ω)	
12	SCL	I	I ² C: Serial Clock	interner Pull-Up: (4,7k Ω)	
13	SBUF Test	O I	Low: Sendepuffer enthält Daten PowerOn Low: Testmodus aktiviert	interner Pull-Up (10k Ω)	
14	DAC	O	Analogausgang	0~3.1V	
15	AIN1	I	Analogeingang 1	0~3.1V	
16	AIN2	I	Analogeingang 2	0~3.1V	
17	I/O 1	I/O	Digitaler Ein- oder Ausgang	High Power Ausgang	
18	I/O 2	I/O	Digitaler Ein- oder Ausgang	High Power Ausgang	
19	I/O 3	I/O	Digitaler Ein- oder Ausgang	High Power Ausgang	
20	I/O 4	I/O	Digitaler Ein- oder Ausgang	High Power Ausgang	
21 ²⁾	I/O 5	I/O	Digitaler Ein- oder Ausgang / PWM Ausgang	Low Power Ausgang	nur EA PLUGL128-6
22 ²⁾	I/O 6	I/O	Digitaler Ein- oder Ausgang	Low Power Ausgang	
23 ²⁾	I/O 7	I/O	Digitaler Ein- oder Ausgang	Low Power Ausgang	
24 ²⁾	I/O 8	I/O	Digitaler Ein- oder Ausgang	Low Power Ausgang	
25 ²⁾	V _{out}		Spannungsausgang (max. 3.1V)	für externe Peripherie (max. 100mA)	
26 ²⁾	GND		Ground 0V		

- 1) Nur in der Hardwarevariante Z bestückt (EA PLUGL128G-6TCZ oder EA PLUGS102-6GTCZ)
2) Nur für EA PLUGL128-6

SERIELLE SCHNITTSTELLEN

Das Modul verfügt über 4 serielle Schnittstellen, darunter RS232, SPI, PC und ein USB Port. Alle Schnittstellen sind aktiviert. Alle empfangenen Daten werden in einem Empfangspuffer zusammengefasst. Soll eine Schnittstelle exklusiv verwendet werden, kann dies über einen [Protokollbefehl](#) eingestellt werden.

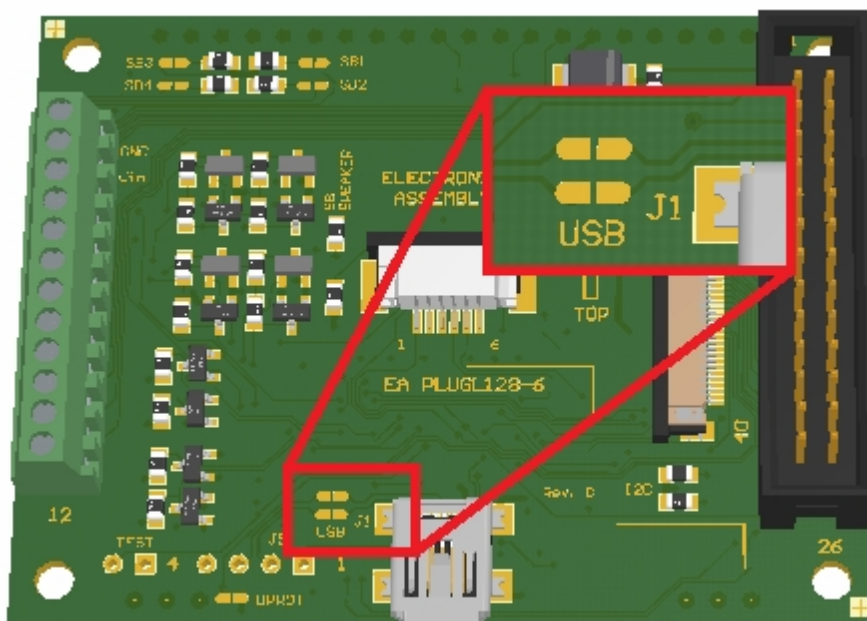
USB

Der **Universal Serial Bus** ist ein serielles Bussystem zur Verbindung mit einem Computer oder anderem Gerät. Er basiert auf einer differentiellen Datenübertragung. Die Bustopologie ist eine strikte Master-Slave-Kommunikation (Ausnahme: On the Go Geräte). Im Fall der EA PLUG-Serie muss immer der PC/Master die Kommunikation leiten. Das Modul verfügt über eine CDC Geräteklasse und meldet sich damit als virtuelle serielle COM-Schnittstelle am PC an:

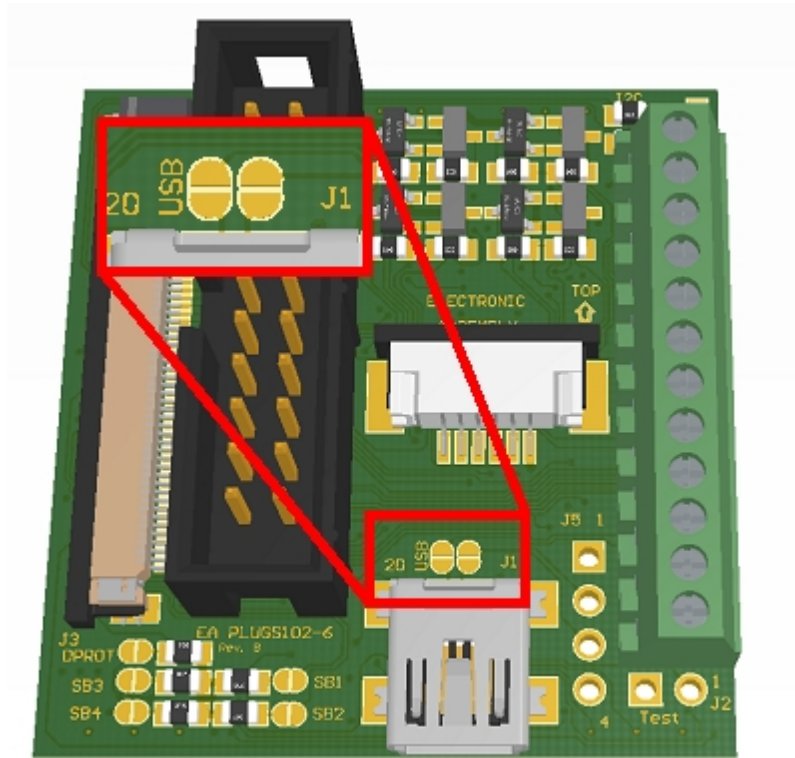
Beschreibung	Wert
Geräteklasse	2
USB Vendor ID	0x2DA9
USB Product ID	0x0CE4
Gerätebeschreibung	EA PLUG

Um das Modul zu programmieren, Einstellungen vorzunehmen oder für erste Tests empfehlen wir die USB-Schnittstelle. Sie ist einfach anzuschließen, schnell und es müssen keine Schnittstellenparameter angepasst werden. Der Windowstreiber kann direkt von unserer Homepage heruntergeladen werden.

Soll die USB Verbindung nicht über den Mini-USB Stecker (J1) erfolgen sondern über den Primary connector (J5), müssen zwei Lötbrücken geschlossen werden:



Lötbrücken für den Primary connector (EA PLUG128-6)



Lötbrücken für Primary connector (EA PLUGS102-6)

Die Pinbelegung des Primary connector(J5) ist kompatibel zur Pinbelegung des PC Mainboards (Polarität beachten).

Pin	Symbol	Beschreibung
1	VBUS	+5V
2	D-	Data -
3	D+	Data +
4	GND	Ground 0V

Hinweis:

Das [Protokoll](#) ist bei USB immer zu verwenden. Es ist nicht möglich die USB-Schnittstelle zu nutzen und das Protokoll zu deaktivieren. Die Lötbrücke DPROT darf nicht geschlossen werden. Die hohe Geschwindigkeit von USB führt zu Pufferüberläufen, die durch das Protokoll verhindert werden.

RS232

RS232 ist ein Standard für eine serielle Schnittstelle. Die Übertragung erfolgt seriell asynchron. Die Daten werden also in einen Bitstrom gewandelt und übertragen. Es existiert keine Taktleitung, jeder Busteilnehmer muss also mit der selben Übertragungsrates (sogenannte Baudrate) arbeiten. RS232 ist eine Spannungsschnittstelle, die Dateninformationen werden durch Spannungspegel übertragen. In der PC-Welt und Industriesteuerungen sind Pegel von +12V bzw. - 12V als Standard definiert. Innerhalb von Platinen bzw. in Mikrocontrollersteuerungen wird mit 0V bzw. VDD (im Fall der EA PLUG-Serie **3,1V**) gearbeitet. Um die Signalpegel anzupassen gibt es einige Möglichkeiten in Form von Levelshiftern (z.B. ICL232, MAX202). RS232 besteht aus "hörenden" und "sprechenden" Leitungen, die zwischen den beiden Teilnehmern gekreuzt werden. In der EA PLUG-Serie ist das Datenformat fest auf 8-N-1 festgelegt:



Die EA PLUG-Serie kann mit folgenden Baudraten arbeiten:

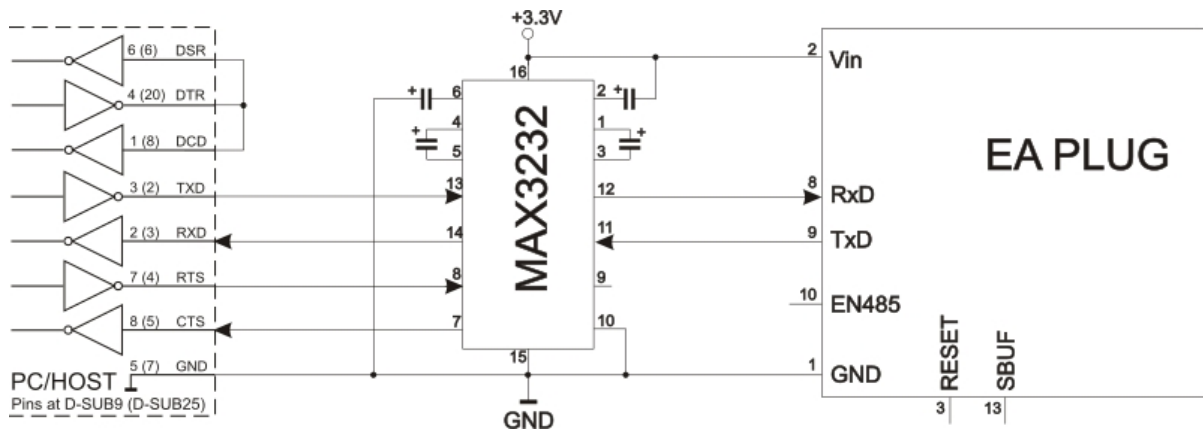
Baud	Fehler
2400	+0.16
4800	+0.16
9600	+0.16
19200	+0.16
38400	+0.16
57600	+0.16
115200	+0.16

Der Parameter (Baudrate) wird mit dem Befehl [#+R](#) gesetzt.

Applikationsbeispiele:

RS232 with Pegelwandler:

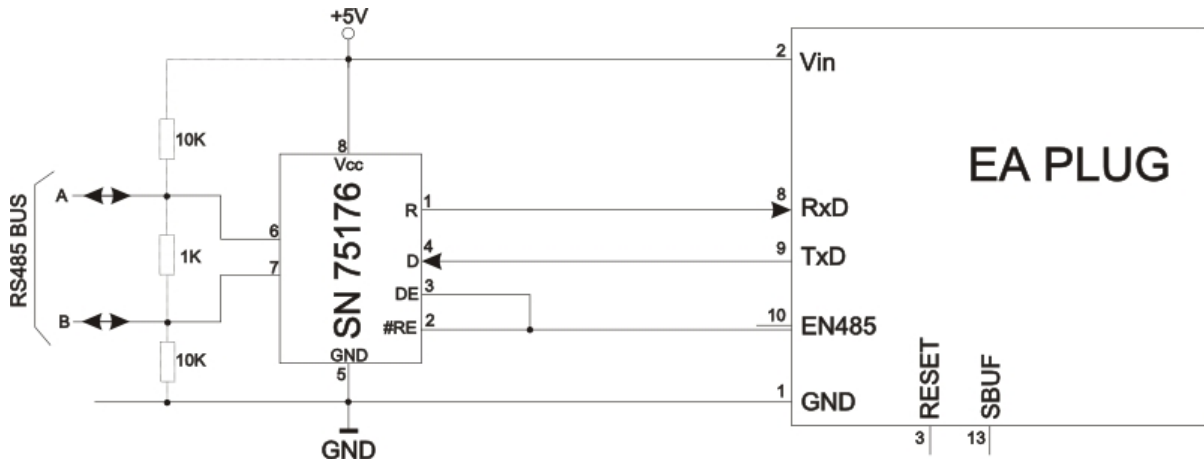
Die EA PLUG-Serie ist für den direkten Anschluss an eine RS-232 Schnittstelle mit CMOS Pegeln (im Fall der EA PLUG-Serie **3,1V**) geeignet. Steht jedoch nur eine Schnittstelle mit $\pm 12V$ Pegeln zur Verfügung, so ist ein externer Pegelwandler erforderlich (z.B. ICL3232, MAX3232).



RS232 V24 - Verbindung zu einem PC (EA PLUG-Serie als Slave)

RS485/RS422 Schnittstelle:

Mit einem externen Umsetzer (z.B. SN75176) kann die EA PLUG-Serie an einen 2-Draht RS-485 Bus angeschlossen werden. Somit können große Entfernungen bis zu 1200m (Ferndisplay) realisiert werden. Der Betrieb von mehreren EA PLUGs an einem RS-485 Bus ist durch Einstellen von Adressen möglich (Befehl [#+R](#)).



RS485 - Verbindung zu einer SPS (EA PLUG-Serie als Slave)

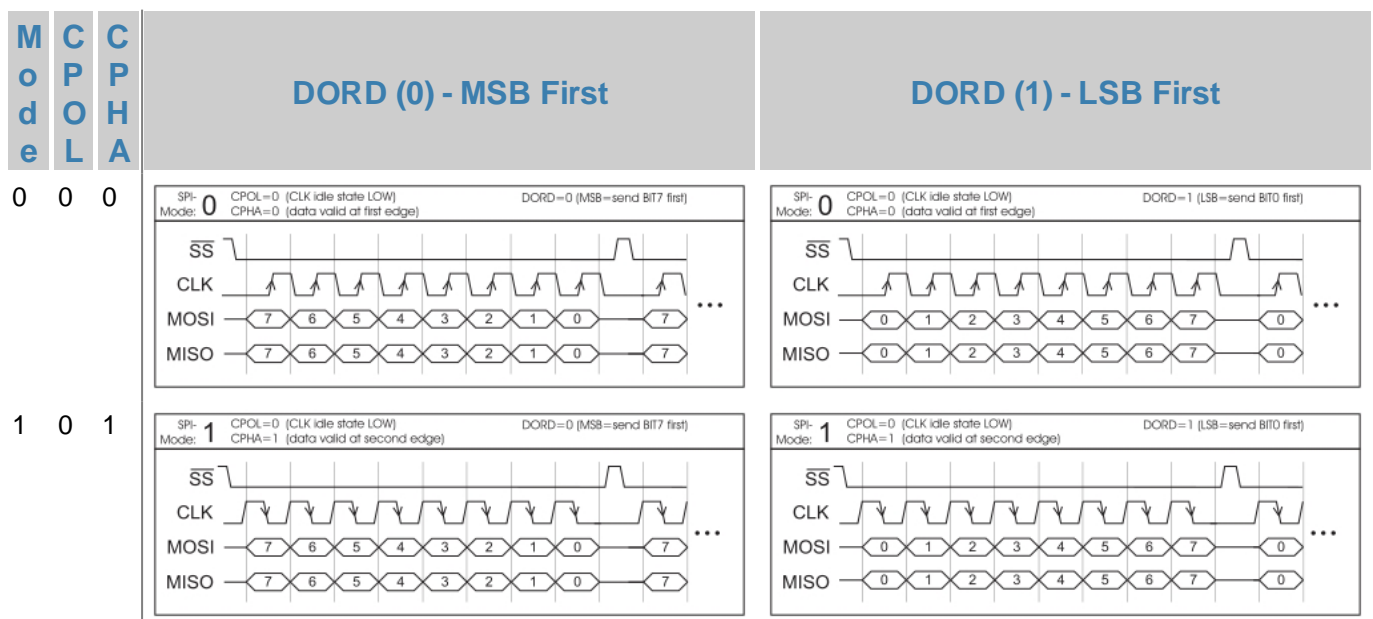
SPI

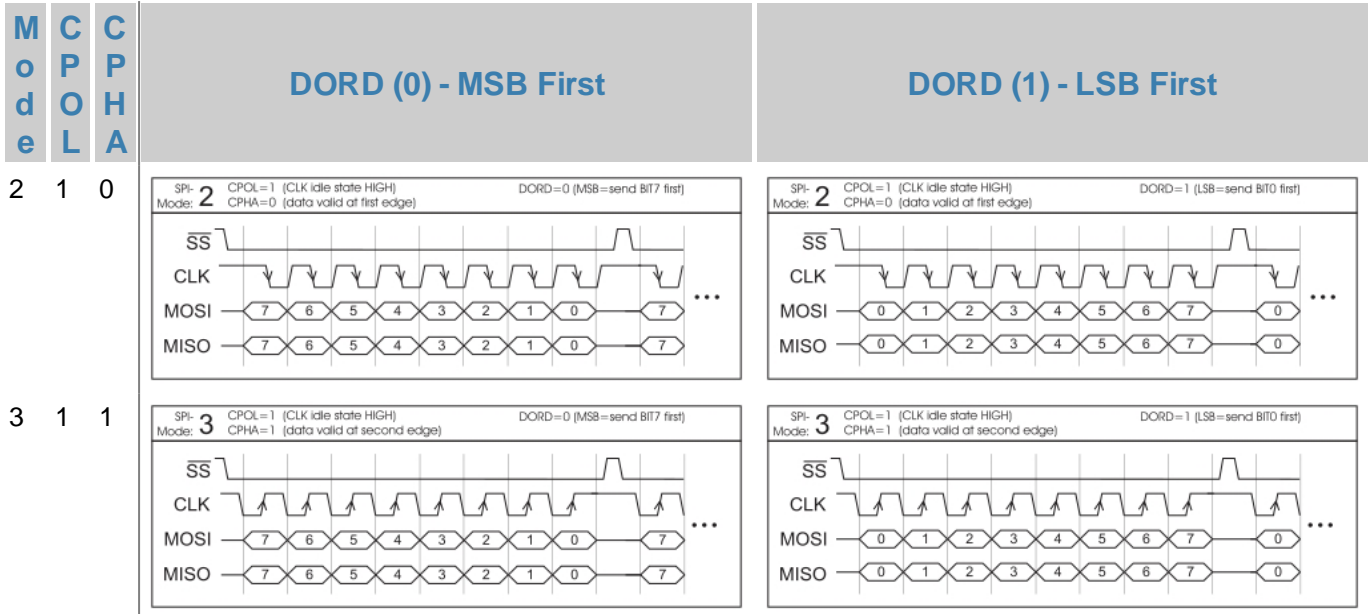
Das **S**erial **P**eripheral **I**nterface ist ein Bussystem für eine serielle synchrone Datenübertragung zwischen verschiedenen ICs. Der Bus besteht aus folgenden Leitungen:

- MOSI (**M**aster **O**ut → **S**lave **I**n) auch SDO (Serial Data Out) oder DO
- MISO (**M**aster **I**n ← **S**lave **O**ut) auch SDI (Serial Data In) oder DI
- SCK (**S**erial **C**lock) - Schiebetakt
- SS (**S**lave **S**elect → Adressierung des Partners) auch CS (Chip Select)

SPI arbeitet mit einem bidirektionalem Übertragungsprinzip, es werden also zeitgleich Daten zwischen den Partner ausgetauscht. Jede Kommunikation wird vom Master mit Hilfe der SCK-Leitung bestimmt.

Das Protokoll für die Datenübertragung ist bei SPI nicht festgelegt, daher gibt es verschiedene Einstellmöglichkeiten. Diese werden durch die Parameter Clock Polarity, Clock Phase sowie Data Order festgelegt. Voreingestellt ist der SPI-Mode 3 mit DORD=0. Über den Befehl `#+S` kann er auf einen der folgenden Modi 0..3 umgeschaltet werden. Alternativ kann der Befehl direkt in das [PowerOn macro](#) geschrieben werden.





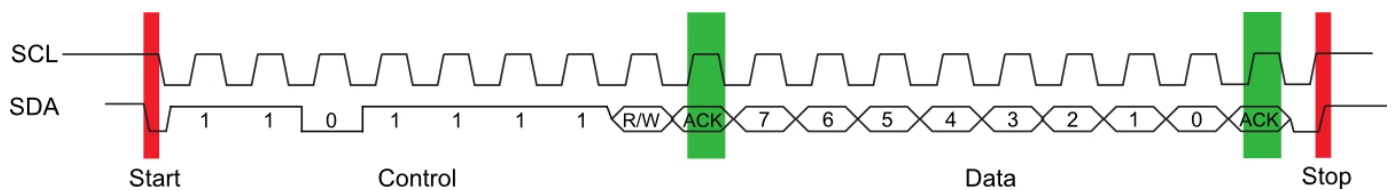
Die maximale Frequenz des Moduls ist 1MHz. Das Modul benötigt eine bestimmte Zeit um die Daten bereit zu stellen; deshalb muss vor den zu lesenden Byte mindestens **50µs** gewartet werden (keine Aktivität auf der SCK-Leitung).

I²C

I²C steht für **I**nter-**I**ntegrated **C**ircuit und ist ein von Phillips entwickelter serieller Datenbus. Der als Master-Slave-Bus konzipierte Bus benötigt 2 Signalleitungen:

- SCL (**S**erial **C**lock **L**ine)
- SDA (**S**erial **D**ata **L**ine)

Die elektrische Spezifikation sieht vor, dass beide Leitungen mit einem Pull-Up-Widerstand an VDD abgeschlossen werden, denn sämtliche an dem Bus angeschlossene Geräte haben Open-Collector-Ausgänge. Der Bustakt wird immer durch den Master vorgegeben, der die gesamte Kommunikation bestimmt:



Nach der Startbedingung folgt in einem Übertragungsprotokoll immer die Slaveadresse. Hierbei ist das Bit 0 das sogenannte R/W-Bit und bestimmt ob vom Slave gelesen (1) oder Daten übermittelt (0) werden sollen. Der Datenaustausch erfolgt bis der Master die Stopbedingung ausführt. Genauere Informationen sind in der I²C Spezifikation zu finden. Die voreingestellte I²C-Bus Adresse ist 0xDE (je nach Schreibweise auch 0x6F). Über den Befehl **#+** kann auf eine x-beliebige andere Adresse umgestellt werden. Alternativ kann der Befehl direkt in das [PowerOn macro](#) geschrieben werden.

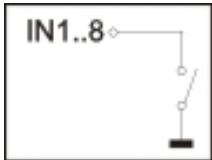
Die maximale Frequenz des Moduls beträgt 400kHz. Das Modul benötigt eine bestimmte Zeit um die Daten bereit zu stellen; deswegen muss vor den zu lesenden Byte mindestens **50µs** gewartet werden (keine Aktivität auf der SCL-Leitung).

DIGITALE EIN- UND AUSGÄNGE

Das Modul verfügt über 8 digital I/O's (CMOS Pegel, nicht potentialfrei). Sie können in beliebiger Anzahl umdefiniert werden.

Anmerkung: I/O 5 kann als PWM Ausgang verwendet werden.

Eingänge (I/O 1-8)



Im Auslieferungszustand sind alle Leitungen als Eingänge eingestellt. Jeder Eingang hat einen internen Pullup, somit ist es möglich Taster und Schalter direkt nach GND anzuschließen. Die Eingänge können mit dem Befehl [#YR](#) abgefragt und ausgewertet werden.

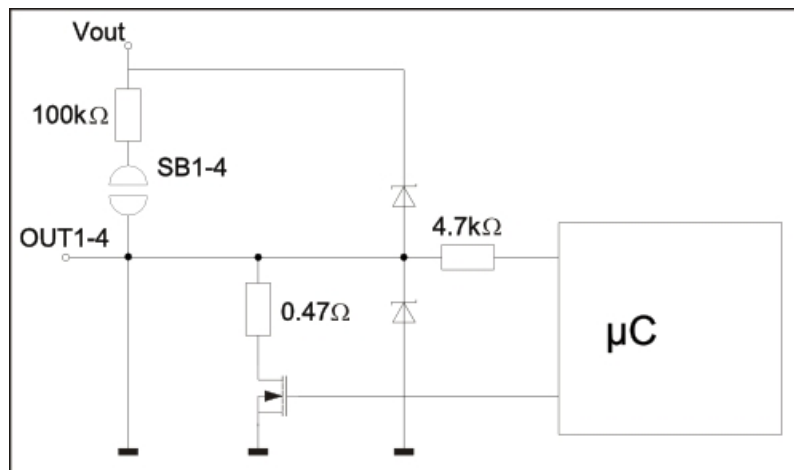
Zusätzlich ist es möglich, bei Änderungen an den Eingängen ein Bit- / Portmakro automatisch aufzurufen. Die automatische Portabfrage lässt sich mit dem Befehl [#YA](#) aktivieren und deaktivieren. Bei jeder Änderung des Eingangsports werden zuerst die Bitmakros und dann das Portmakro ausgeführt. Ist kein Makro definiert so wird der neue Portzustand in den Sendepuffer gestellt (<ESC>[P](#)).

Anmerkung: Die Logik ist für langsame Vorgänge ausgelegt; d.h. mehr als 3 Änderungen pro Sekunde können nicht mehr sinnvoll ausgeführt werden.

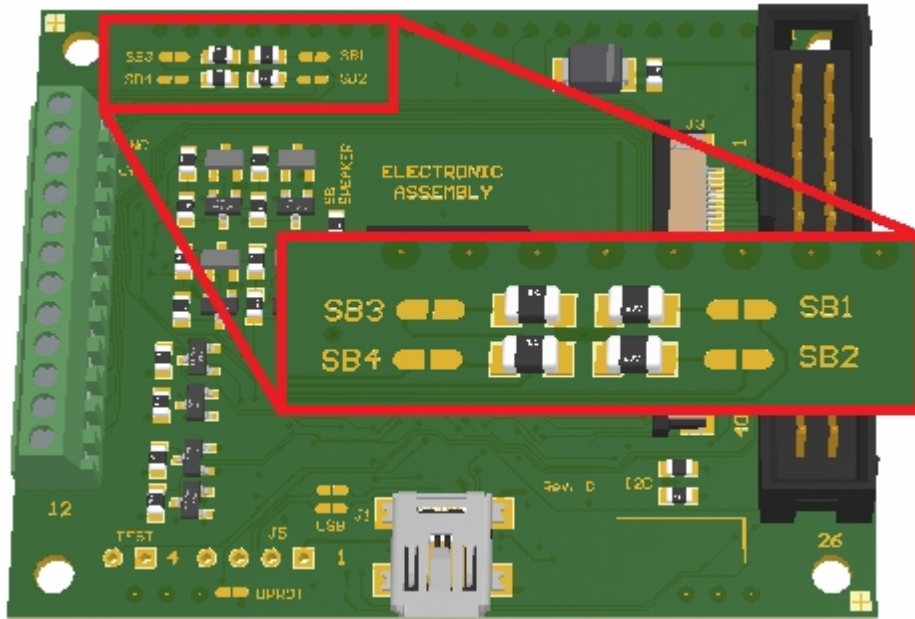
Ausgänge (I/O 1-4 / High power)

Über den Befehl [#YM](#) können ein oder mehrere Eingänge als Ausgang umdefiniert werden. Jeder Ausgang kann per Befehl [#YW](#) individuell angesteuert werden. Diese Port Pins besitzen bereits einen internen MOSFET (max. 360mA)

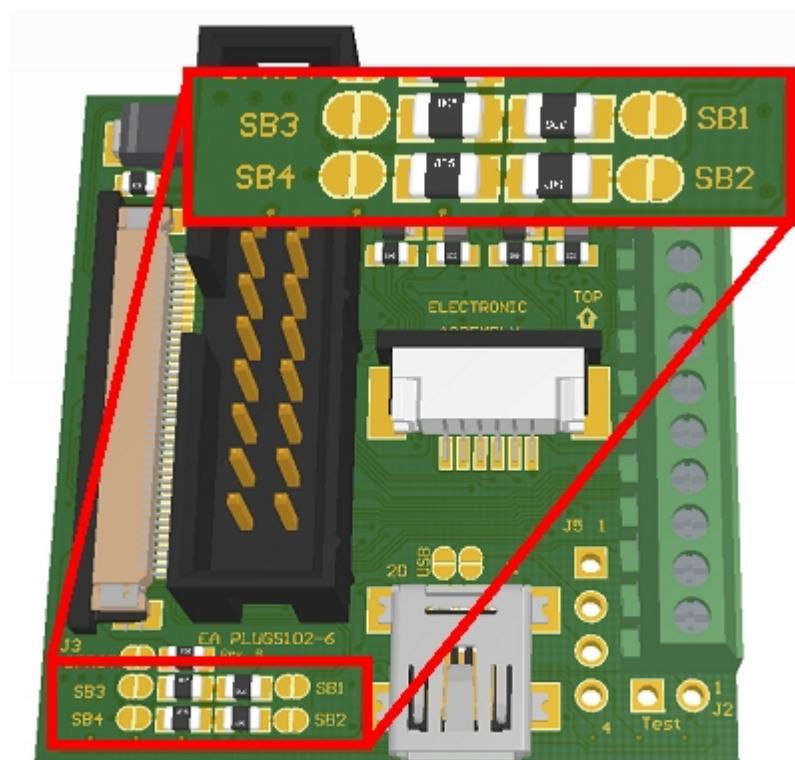
Anmerkung: Möchte man die internen pull up Widerstände (100kΩ) verwenden muss man die Lötbrücken (SB1-SB4) schließen.



I/O port Beschaltung (Port 1-4)



Lötbrücken für I/O ports 1-4 (EA PLUG128-6)

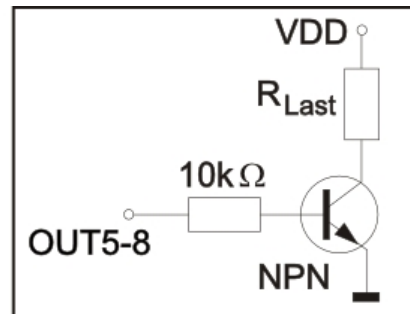
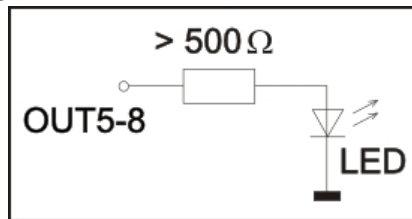


Lötbrücken für I/O ports 1-4 (EA PLUGS102-6)

Ausgänge (I/O 5-8 / Low power)

Über den Befehl **#YM** können ein oder mehrere Eingänge als Ausgang umdefiniert werden. Jeder Ausgang kann per Befehl **#YW** individuell angesteuert werden.

Pro Leitung kann ein Strom von maximal 5mA geschaltet werden. Es ist somit möglich, mit einem Ausgang direkt eine LED (low current) zu schalten. Für höhere Ströme muss ein externer Transistor verwendet werden.



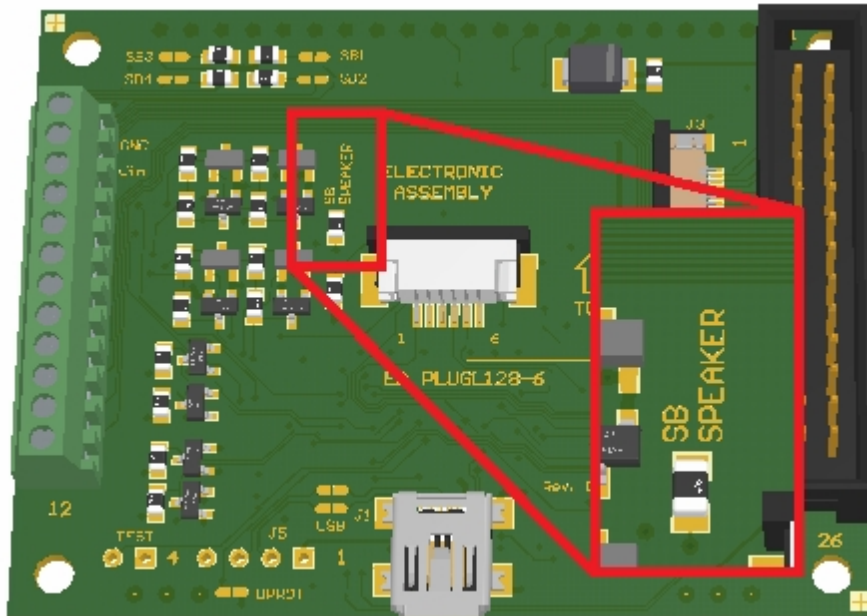
Applikationsbeispiel (Port 5-8)

PWM

Das Modul verfügt über die Möglichkeit mittels PWM-Signal (Pulsweitenmodulation) externe Komponenten anzusteuern. Der Ausgangspin des PWM Signals ist I/O Port 5. Dabei wird bei konstanter Frequenz (einstellbar von 2 Hz bis 24 kHz #YO) das Tastverhältnis eines rechteckigen Impulses geändert. Durch die Modulation ändert sich das Verhältnis zwischen An- und Ausschaltzeit und somit die Charakteristik des Ausgangssignals. Auf diese Art können elektromechanische Bauteile wie z.B. Motoren angesteuert werden oder auch eine quasi-analog Spannung erzeugt werden. Die Variation der Tastverhältnisse sorgt dann für eine geringe Motordrehzahl/Spannung bei kurzer Anschaltzeit oder eine hohe Motordrehzahl/Spannung bei langer Anschaltzeit. Der Ausgangspegel liegt bei 0V und V_{in} .

EXTERNER LAUTSPRECHER

Die EA PLUG-Serie wird mit einem Lautsprecher ausgeliefert. Ist dieser zu leise kann man stattdessen einen eigenen Lautsprecher anschließen. Hierzu werden Pin 29 oder Pin 30 (Power) verwendet (siehe hierzu nachfolgende Applikationsbeispiele). Der interne Lautsprecher wird durch Entfernen des 0Ω Widerstandes (SB Speaker) deaktiviert.

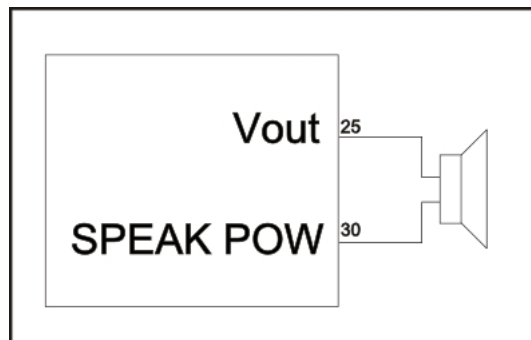


Applikationsbeispiele:

Anschluss eines externen Lautsprechers:

Ein externer Lautsprecher kann direkt an die beiden Pins V_{out} und SPEAK POW angeschlossen werden. Folgende Maximalwerte dürfen nicht überschritten werden.

Wert	min.	typ.	max.	Einheit
Stromaufnahme	-	-	400	mA
Leistungsaufnahme	-	-	1	W
Innenwiderstand Lautsprecher	6	8	32	Ω

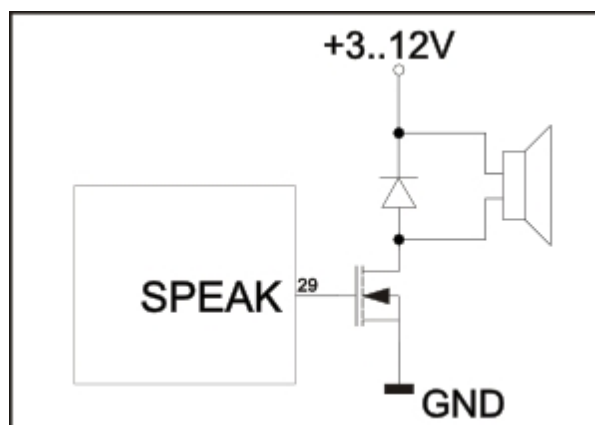


Anschluss eines externen Lautsprechers

Anschluss eines externen Lautsprechers mit mehr Leistung:

Möchte man die Lautstärke weiter erhöhen benötigt man eine zusätzliche Ansteuerschaltung mit eigener Spannungsversorgung.

Wert	min.	typ.	max.	Einheit
Ausgangsspannung V_{speak} (Pin 29)	-	-	3,1	V
Duty cycle	-	50	-	%
Frequenz	-	4	-	kHz



Anschluss eines Lautsprechers mit eigener Ansteuerschaltung

SOFTWARE

Die Programmierung des Displays erfolgt über Befehle wie z.B. Zeichne ein Rechteck von (0,0) nach(64,15). Es ist keine zusätzliche Software oder Treiber erforderlich. Zeichenketten lassen sich pixelgenau platzieren. Blinkattribute können beliebig oft vergeben werden - auch für Grafiken. Das Mischen von Text und Grafik ist jederzeit möglich. Es können bis zu 16 verschiedene Zeichensätze verwendet werden. Jeder Zeichensatz kann wiederum 2- bis 4-fach gezoomt werden. Mit dem größten Zeichensatz lassen sich somit bildschirmfüllende Worte und Zahlen darstellen.

EBENEN

Die EA PLUG-Serie besitzt zwei verschiedene Ebenen:

- Terminal-Ebene
- Grafik-Ebene

Die Terminal-Ebene kann für die ersten Schritte oder zum Debuggen verwendet werden. Beim Starten des Moduls ist die Terminal-Ebene aktiv und der Cursor blinkt in der ersten Zeile. Alle empfangenen Zeichen werden auf dem Modul im ASCII Format angezeigt (Ausnahmen: CR, LF, FF, ESC, '#'). Voraussetzung dafür ist ein [funktionierender Protokollrahmen](#) oder ein [abgeschaltetes Protokoll](#). Der Zeilenvorschub erfolgt automatisch oder durch das Zeichen 'LF'. Ist die letzte Zeile voll, scrollt der Terminalinhalt nach oben. Beim Zeichen 'FF' wird das Terminal gelöscht.

Das Zeichen '#' wird als Escape-Zeichen verwendet und ist somit nicht direkt im Terminal darstellbar. Soll das Zeichen '#' im Terminal ausgegeben werden, so muss es doppelt gesendet werden '##'.

Das Terminal besitzt eine eigene Ebene zur Darstellung und ist somit völlig unabhängig von den Graphikausgaben. Wird z.B. der Graphikbildschirm mit [#DL](#) gelöscht, so beeinflusst das nicht den Inhalt des Terminalfensters.

Der Terminalfont ist fest im ROM vorhanden und kann auch für Grafikausgaben 'ESC Z...' verwendet werden (FONT nr=0 einstellen).

+ Lower Upper	\$0 (0)	\$1 (1)	\$2 (2)	\$3 (3)	\$4 (4)	\$5 (5)	\$6 (6)	\$7 (7)	\$8 (8)	\$9 (9)	\$A (10)	\$B (11)	\$C (12)	\$D (13)	\$E (14)	\$F (15)
\$00 (dez: 0)	o	o	o	o	o	o	o	o	o	o	o	o	o	o	o	o
\$10 (dez: 16)	0	1	2	3	4	5	6	7	8	9	:	;	<	=	>	?
\$20 (dez: 32)		!	"	#	\$	%	&	'	()	*	+	,	-	.	/
\$30 (dez: 48)	0	1	2	3	4	5	6	7	8	9	:	;	<	=	>	?
\$40 (dez: 64)	@	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O
\$50 (dez: 80)	P	Q	R	S	T	U	V	W	X	Y	Z	[\]	^	_
\$60 (dez: 96)	~	a	b	c	d	e	f	g	h	i	j	k	l	m	n	o
\$70 (dez: 112)	p	q	r	s	t	u	v	w	x	y	z	{		}	~	Δ
\$80 (dez: 128)	ø	ü	é	â	ä	å	ç	è	ë	ê	ï	î	ï	ä	å	
\$90 (dez: 144)	é	æ	œ	ô	ö	ò	û	ü	ö	ü	ç	£	¥	β	f	
\$A0 (dez: 160)	á	í	ó	ú	ñ	ñ	ø	ø	¿	¿	½	¼	i	«	»	
\$B0 (dez: 176)	::	::	::													
\$C0 (dez: 192)	L	L	T	T	-	+	F	H	H	H	H	H	H	H	H	H
\$D0 (dez: 208)	μ	π	π	μ	E	F	π	#	#	J	r	■	■	■	■	■
\$E0 (dez: 224)	α	β	Γ	π	Σ	σ	μ	τ	ϕ	θ	Ω	δ	φ	Ε	Π	
\$F0 (dez: 240)	≡	±	≥	≤	ρ	J	÷	≈	°	*	.	√	n	2	3	-

Terminal font

Alle anderen Befehle werden in der Grafik-Ebene angezeigt. Jeder Befehl beginnt mit ESCAPE oder HASH gefolgt von einem oder zwei Befehlsbuchstaben und deren Parameter. Es gibt zwei

Möglichkeiten Befehle an das Modul zu übertragen:

ASCII-Modus

- Das Escape-Zeichen entspricht dem Zeichen '#' (hex: \$23, dec: 35).
- Die Befehlsbuchstaben folgen direkt im Anschluss an das '#' Zeichen.
- Die Parameter werden im Klartext (mehrere ASCII Ziffern) mit einem nachfolgenden Trennzeichen (z.B. das Komma ',') gesendet, auch hinter dem letzten Parameter z.B.: #GD0,0,159,103,
- Zeichenketten (Texte) werden direkt ohne Anführungsstrichen geschrieben und mit CR (hex: \$0D) oder LF (hex: \$0A) abgeschlossen.

Binär-Modus

- Das Escape-Zeichen entspricht dem Zeichen ESC (hex: \$1B, dec: 27).
- Die Befehlsbuchstaben werden direkt gesendet.
- Die Koordinaten x und y und alle anderen Parameter werden als 8-Bit Binärwert (1 Byte) gesendet.
- Zeichenketten (text) werden mit CR (hex: \$0D), LF (hex: \$0A) oder NUL (hex: \$00) abgeschlossen.

Im Binär-Modus dürfen keine Trennzeichen z.B. Leerzeichen oder Kommas verwendet werden. Die Befehle benötigen kein Abschlussbyte wie z.B. Carriage Return (außer Zeichenkette: \$00).

DISPLAY BLINKMODUS

Nach dem Einschalten oder dem Befehl 'ESC DG 0' ist das PLUG im Blinkmodus. Zwei Bildinhalte werden abwechselnd in einem einstellbaren Zeitraum nacheinander angezeigt.

Blinkattribute werden mit den Befehlen [#ZB](#), [#UB](#), [#GB](#) eingestellt:

- n1=0: Kein blinken
- n1=1: An/Aus blinken
- n1=2: Invertierend blinken
- n1=3: Aus/An blinken (phasenverschoben)

Innerhalb von Zeichenketten ([#ZL](#), [#ZC](#), [#ZR](#)) kann das Blinken lokal aktiviert werden:

- Texte zwischen zwei '~' (\$7E) blinken An/Aus.
- Texte zwischen zwei '&' (\$26) blinken Aus/An phasenverschoben.
- Texte zwischen zwei '@' (\$40) blinken invertierend..

Außerdem kann man den Blinkbereichsfunktionen 'ESC Q...' rechteckigen Bereichen nachträglich ein Blinkattribut zugewiesen oder entfernt werden.

ZEICHENSÄTZE

Es sind standardmäßig, außer dem 8x8 Terminalfont (Font-Nr. 0), noch 3 monospaced, 3 proportionale Zeichensätze und 1 großer Ziffernfont integriert. Die proportionalen Zeichensätze ergeben ein schöneres Schriftbild, gleichzeitig benötigen sie weniger Platz auf dem Bildschirm (z.B. schmales 'i' und breiten 'W'). Jedes Zeichen kann pixelgenau platziert werden und in der Höhe und Breite von 1- bis 4-fach vergrößert werden.

Texte lassen sich linksbündig, rechtsbündig und zentriert ausgeben. Auch eine 90°, 180° und 270° Drehung ist möglich. Mit Hilfe des KitEditors ([EA LCD-Tools](#)) ist es möglich, weitere Fonts zu integrieren (bis zu 15).

Vorgeladene Zeichensätze

+ Lower Upper	\$0 (0)	\$1 (1)	\$2 (2)	\$3 (3)	\$4 (4)	\$5 (5)	\$6 (6)	\$7 (7)	\$8 (8)	\$9 (9)	\$A (10)	\$B (11)	\$C (12)	\$D (13)	\$E (14)	\$F (15)
\$20 (dez: 32)		!	"	#	\$	%	&	'	()	*	+	,	-	.	/
\$30 (dez: 48)	0	1	2	3	4	5	6	7	8	9	:	;	<	=	>	?
\$40 (dez: 64)	@	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O
\$50 (dez: 80)	P	Q	R	S	T	U	V	W	X	Y	Z	[\]	^	_
\$60 (dez: 96)	`	a	b	c	d	e	f	g	h	i	j	k	l	m	n	o
\$70 (dez: 112)	p	q	r	s	t	u	v	w	x	y	z	{		}	~	
\$80 (dez: 128)																
\$90 (dez: 144)																

Font 1: 4x6 monospaced

+ Lower Upper	\$0 (0)	\$1 (1)	\$2 (2)	\$3 (3)	\$4 (4)	\$5 (5)	\$6 (6)	\$7 (7)	\$8 (8)	\$9 (9)	\$A (10)	\$B (11)	\$C (12)	\$D (13)	\$E (14)	\$F (15)
\$20 (dez: 32)		!	"	#	\$	%	&	'	<	>	*	+	,	-	.	/
\$30 (dez: 48)	0	1	2	3	4	5	6	7	8	9	:	;	<	=	>	?
\$40 (dez: 64)	@	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O
\$50 (dez: 80)	P	Q	R	S	T	U	V	W	X	Y	Z	[\]	^	_
\$60 (dez: 96)	`	a	b	c	d	e	f	g	h	i	j	k	l	m	n	o
\$70 (dez: 112)	p	q	r	s	t	u	v	w	x	y	z	{		}	~	
\$80 (dez: 128)										¡	¢	£	¤	¥	¦	§
\$90 (dez: 144)	¨	©	ª	«	¬	­	®	¯	°	±	²	³	´	µ	¶	·
\$A0 (dez: 160)	¸	¹	º	»	¼	½	¾	¿								
\$B0 (dez: 176)																
\$C0 (dez: 192)																
\$D0 (dez: 208)																
\$E0 (dez: 224)																
\$F0 (dez: 240)																

Font 2: 6x8 monospaced

+ Lower Upper	\$0 (0)	\$1 (1)	\$2 (2)	\$3 (3)	\$4 (4)	\$5 (5)	\$6 (6)	\$7 (7)	\$8 (8)	\$9 (9)	\$A (10)	\$B (11)	\$C (12)	\$D (13)	\$E (14)	\$F (15)
\$20 (dez: 32)		!	"	#	\$	%	&	'	()	*	+	,	-	.	/
\$30 (dez: 48)	0	1	2	3	4	5	6	7	8	9	:	;	<	=	>	?
\$40 (dez: 64)	a	b	c	d	e	f	g	h	i	j	k	l	m	n	o	
\$50 (dez: 80)	p	q	r	s	t	u	v	w	x	y	z	[\]	^	_
\$60 (dez: 96)	`	a	b	c	d	e	f	g	h	i	j	k	l	m	n	o
\$70 (dez: 112)	p	q	r	s	t	u	v	w	x	y	z	{		}	~	Δ
\$80 (dez: 128)	€	ü	é	â	ä	à	ã	ç	ê	ë	è	ï	í	ì	ñ	â
\$90 (dez: 144)	É	æ	Æ	ô	ö	ò	û	ù	ÿ	Ö	Ü	¢	£	¥	ß	f
\$A0 (dez: 160)	á	í	ó	ú	ñ	Ñ	ª	º	¿	ª	¼	½	¾	¡	«	»
\$B0 (dez: 176)																
\$C0 (dez: 192)																
\$D0 (dez: 208)																
\$E0 (dez: 224)	α	β	Γ	π	Σ	σ	μ	τ	ϕ	θ	η	δ	φ	φ	ε	η
\$F0 (dez: 240)	≡	±	≥	≤	Γ	J	÷	≈	°	•	•	√	n	z	ε	-

Font 3: 7x12 monospaced

+ Lower Upper	\$0 (0)	\$1 (1)	\$2 (2)	\$3 (3)	\$4 (4)	\$5 (5)	\$6 (6)	\$7 (7)	\$8 (8)	\$9 (9)	\$A (10)	\$B (11)	\$C (12)	\$D (13)	\$E (14)	\$F (15)
\$20 (dez: 32)		!	"	#	\$	%	&	'	()	*	+	,	-	.	/
\$30 (dez: 48)	0	1	2	3	4	5	6	7	8	9	:	;	<	=	>	?
\$40 (dez: 64)	@	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O
\$50 (dez: 80)	P	Q	R	S	T	U	V	W	X	Y	Z	[\]	^	_
\$60 (dez: 96)	`	a	b	c	d	e	f	g	h	i	j	k	l	m	n	o
\$70 (dez: 112)	p	q	r	s	t	u	v	w	x	y	z	{		}	~	Δ
\$80 (dez: 128)	€	ü	é	â	ä	à	ã	ç	ê	ë	è	ï	î	í	Ä	Å
\$90 (dez: 144)	É	æ	Æ	ô	ö	ò	û	ù	ÿ	Ö	Ü					
\$A0 (dez: 160)	á	í	ó	ú	ñ	Ñ	ä	ö								
\$B0 (dez: 176)																
\$C0 (dez: 192)																
\$D0 (dez: 208)																
\$E0 (dez: 224)		ß														
\$F0 (dez: 240)									°							

Font 4: GENEVA10 proportional

+ Lower Upper	\$0 (0)	\$1 (1)	\$2 (2)	\$3 (3)	\$4 (4)	\$5 (5)	\$6 (6)	\$7 (7)	\$8 (8)	\$9 (9)	\$A (10)	\$B (11)	\$C (12)	\$D (13)	\$E (14)	\$F (15)
\$20 (dez: 32)		!	"	#	\$	%	&	'	()	*	+	,	-	.	/
\$30 (dez: 48)	0	1	2	3	4	5	6	7	8	9	:	;	<	=	>	?
\$40 (dez: 64)	@	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O
\$50 (dez: 80)	P	Q	R	S	T	U	V	W	X	Y	Z	[\]	^	_
\$60 (dez: 96)	`	a	b	c	d	e	f	g	h	i	j	k	l	m	n	o
\$70 (dez: 112)	p	q	r	s	t	u	v	w	x	y	z	{		}	~	Δ
\$80 (dez: 128)	€	ü	é	â	ä	à	å	ç	ê	ë	è	ì	î	ï	Ä	Å
\$90 (dez: 144)	É	æ	Æ	ô	ö	ò	û	ù	ÿ	Ö	Ü					
\$A0 (dez: 160)	á	í	ó	ú	ñ	Ñ	ä	ö								
\$B0 (dez: 176)																
\$C0 (dez: 192)																
\$D0 (dez: 208)																
\$E0 (dez: 224)		ß														
\$F0 (dez: 240)									°							

Font 5: CHICAGO14 proportional

+ Lower Upper	\$0 (0)	\$1 (1)	\$2 (2)	\$3 (3)	\$4 (4)	\$5 (5)	\$6 (6)	\$7 (7)	\$8 (8)	\$9 (9)	\$A (10)	\$B (11)	\$C (12)	\$D (13)	\$E (14)	\$F (15)
\$20 (dez: 32)		!	"	#	\$	%	&	'	()	*	+	,	-	.	/
\$30 (dez: 48)	0	1	2	3	4	5	6	7	8	9	:	;	<	=	>	?
\$40 (dez: 64)	@	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O
\$50 (dez: 80)	P	Q	R	S	T	U	V	W	X	Y	Z	[\]	^	_
\$60 (dez: 96)	'	a	b	c	d	e	f	g	h	i	j	k	l	m	n	o
\$70 (dez: 112)	p	q	r	s	t	u	v	w	x	y	z	{		}	~	Δ
\$80 (dez: 128)	€	ü	é	â	ä	à	â	ç	ê	ë	è	ï	î	ì	Ä	Å
\$90 (dez: 144)	É	æ	Æ	ô	ö	ò	û	ù	ÿ	Ö	Ü					
\$A0 (dez: 160)	á	í	ó	ú	ñ	Ñ	ä	ö								
\$B0 (dez: 176)																
\$C0 (dez: 192)																
\$D0 (dez: 208)																
\$E0 (dez: 224)		β														
\$F0 (dez: 240)									◦							

Font 6: Swiss30 Bold proportional

+ Lower Upper	\$0 (0)	\$1 (1)	\$2 (2)	\$3 (3)	\$4 (4)	\$5 (5)	\$6 (6)	\$7 (7)	\$8 (8)	\$9 (9)	\$A (10)	\$B (11)	\$C (12)	\$D (13)	\$E (14)	\$F (15)
\$20 (dez: 32)												+	-	.		
\$30 (dez: 48)	0	1	2	3	4	5	6	7	8	9	:					

Font 7: grosse Ziffern BigZif57

Ladbare Zeichensätze

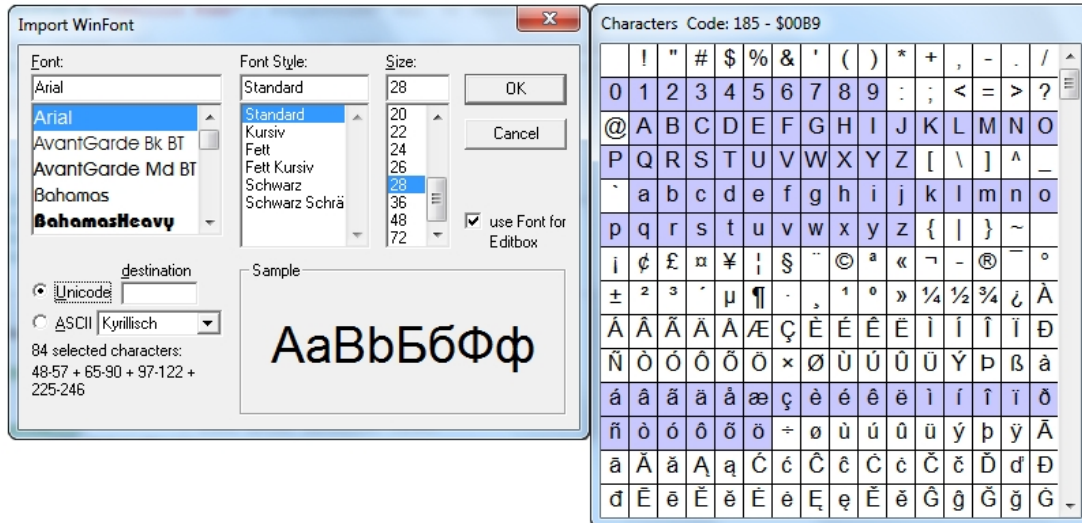
Der [KitEditor](#) kann verwendet werden, um weitere Zeichensätze zu integrieren.

- Compilanweisung [WinFont](#):

Damit ist es möglich, TrueType-Fonts in verschiedenen Größen zu rastern und einzubinden. Sie können entweder den kompletten Zeichensatz (ASCII) einbinden oder sie wählen aus dem gesamten Unicode-Zeichensatz bestimmte Zeichen aus. Ein Doppelklick im KitEditor auf den Fontnamen öffnet dazu die Font-Auswahlbox. Um die Verwendung dieser Zeichensätze zu vereinfachen gibt es die komfortable Möglichkeit einer Zeichen-Auswahlbox. Wird im KitEditor ein String ausgegeben (z.B. #ZL 5,5,"Hello") kann mit einem Doppelklick auf den String dieser

geöffnet werden. Dies ist vor Allem bei kyrillischen, asiatischen oder Symbolschriftarten zu empfehlen. Der KitEditor setzt daraufhin automatisch den richtigen ASCII-Code ein. Alternativ zu den Anführungsstrichen können geschweifte Klammern genutzt werden (z.B. #ZL 5,5, {48656C6C6F}).

- Compileranweisung **Font**:
Alle *FXT Zeichensätze können verwendet werden.



Importieren von WinFonts

TOUCH PANEL

Die Version -xxxTC wird mit einem kapazitiven Touchpanel ausgeliefert. Bis zu 40 Touchbereiche (Tasten, Schalter, Menüs, Bargrapheingaben), können gleichzeitig und pixelgenau definiert werden. Die EA PLUG-Serie unterstützt die Darstellung mit komfortablen Befehlen. Beim Berühren der Touch-Tasten können diese automatisch invertiert werden und ein Summer (Pin 29/30) signalisiert die Berührung) Der zuvor definierte Return-Code der Taste wird über die Schnittstelle gesendet oder es wird statt dessen ein internes Touch Makro mit der Nummer des Return-Codes gestartet ([siehe Makro Programmierung](#))

[Rahmen und Tastenformen](#)

Mit den Befehlen Rahmen/Rahmenbox zeichnen sowie beim Zeichnen von Touchtasten kann ein Rahmentyp eingestellt werden. Es stehen dabei 16 Rahmentypen zur Verfügung (0=keinen Rahmen zeichnen). Die Rahmengröße muss mindestens 16x16 Pixel betragen.



[Bitmaps als Tasten](#)

Außer den Rahmentypen (sind in der Größe frei skalierbar) gibt es noch die Möglichkeit beliebige Bitmaps (jeweils 2 Stück für nicht-gedrückt und gedrückt) als Touch-Tasten oder -Schalter zu verwenden. Über die LCD-Tools ([KitEditor](#)) können eigene Buttons als Bilder eingebunden werden (Compileranweisung **PICTURE**). Ein Button besteht immer aus zwei gleich großen monochromen Windows-BMPs (ein Bitmap für die normale Darstellung der Touchtaste und ein Bitmap für die gedrückte Touchtaste). Die aktive Fläche der Touchtaste ergibt sich automatisch aus der Größe der Button-Bitmaps.

[Schalter in Gruppen \(Radio group\)](#)

Touch-Schalter ändern ihren Zustand bei jeder Berührung von EIN in AUS und umgekehrt. Mehrere Touchschalter können zu einer Gruppe zusammengefasst werden (Befehl [#AR](#)). Wird nun ein Touch-Schalter innerhalb einer Gruppe 'nr' eingeschaltet, dann werden automatisch alle andern Touch-Schalter dieser Gruppe ausgeschaltet. Es ist also automatisch immer nur ein Schalter gesetzt.

MAKRO PROGRAMMIERUNG

Einzelne oder mehrere Befehlsfolgen können als sog. Makros zusammengefasst und im EEPROM fest abgespeichert werden. Diese können dann mit den Befehlen Makro ausführen gestartet werden. Der [KitEditor](#) wird dazu verwendet solche Makros in den internen Speicher zu schreiben. Es gibt verschiedene Makrotypen (Compileranweisungen sind grün geschrieben):

- **Normal Makro Macro:**
Start per Befehl 'ESC MN xx' über serielle Schnittstelle oder von einem anderen Makro aus. Es können auch mehrere hintereinander liegende Makros automatisch zyklisch aufgerufen werden (Movie, sich drehende Sanduhr, mehrseitiger Hilfetext). Diese automatischen Makros werden solange abgearbeitet bis ein Befehl über die Schnittstelle empfangen wird, oder ein Touchmakro mit entsprechendem Return-Code ausgelöst wird.
Außerdem werden diese Makros von Makro-Prozessen in definierten Intervallen aufgerufen. Makro- Prozesse werden nicht durch Empfang von Befehlen von der Schnittstelle oder von ausgelösten Touchmakros unterbrochen.
- **Touch Makro TouchMacro:**
Start beim Berühren/Loslassen eines Touchfeldes (nur bei Versionen mit Touch Panel TP) oder per Befehl [#MT](#).
- **Menü Makro (1 to 255) MenuMacro:**
Start bei Auswahl eines Menüeintrages oder per Befehl [#MM](#).
- **Bit Makro BitMacro:**
Start bei Anlegen/Änderung einer Spannung an einzelnen Eingängen IN1..8 (Bitweise) oder per Befehl [#MB](#). Die Bit-Makros 1..8 reagieren auf fallende Flanke, Bit-Makros 9..16 auf die steigende Flanke der Eingänge 1..8. Mit dem Befehl [#YD](#) kann die Zuordnung der Eingänge zu den Bitmakros umdefiniert werden.
- **Port Makro PortMacro:**
Start bei Anlegen/Änderung einer Spannung an den 8 Eingängen IN 1..8 (binär kombiniert) oder per Befehl [#MP](#).
- **Analog Makro AnalogMacro:**
Automatischer Start bei Änderung des Analogwertes AIN1 oder AIN2 oder per Befehl [#MV](#).
- **Power-on-Makro PowerOnMacro:**
Start nach dem Einschalten. Hier kann man z.B. den Cursor abschalten und einen Startbildschirm definieren.
- **Reset-Makro ResetMacro:**
Start nach einem externen Reset (Low Pegel an Pin 3).
- **Watchdog-Makro WatchdogMacro:**
Start nach einem Fehlerfall (z.B. Absturz).

Achtung: Wird im PowerOn-, Reset-, oder Watchdog-Makro eine Endlosschleife programmiert, ist das Display nicht mehr ansprechbar. In diesem Fall muss die Ausführung des Power-On Makros unterdrückt werden.. Dies erreicht man mit dem Protokoll Befehl [break command](#).

PROTOKOLL/DATENÜBERTRAGUNG

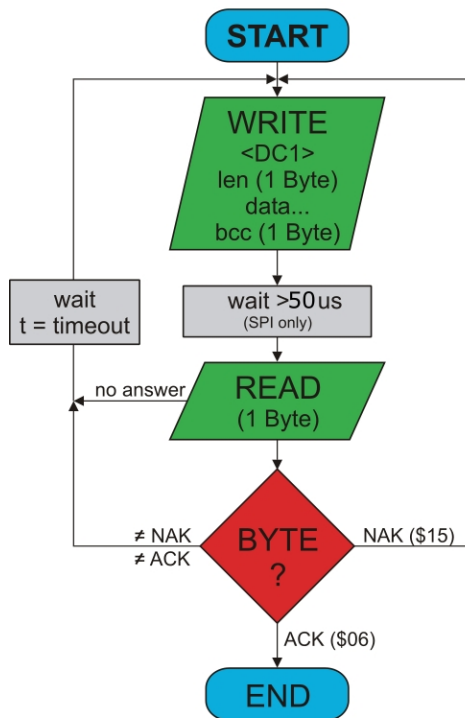
Egal über welche der 4 seriellen Schnittstellen die Daten von der übergeordneten Steuerung übertragen werden ist das Übertragungsprotokoll identisch. Die Hardwarebeschaltung ist unterschiedlich und kann unter dem Punkt [Serielle Schnittstellen](#) nachgelesen werden.

Die Datenübertragung ist jeweils eingebettet in einen festen Rahmen mit Prüfsumme. Die EA PLUG-Serie quittiert dieses Paket mit dem Zeichen <ACK> (=0x06) bei erfolgreichem Empfang oder <NAK> (=0x15) bei fehlerhafter Prüfsumme oder Empfangspufferüberlauf. In Falle eines <NAK> wird das komplette Paket verworfen und muss nochmal gesendet werden. Ein <ACK> bestätigt lediglich die korrekte Übertragung. Ein Syntax-Check erfolgt nicht.

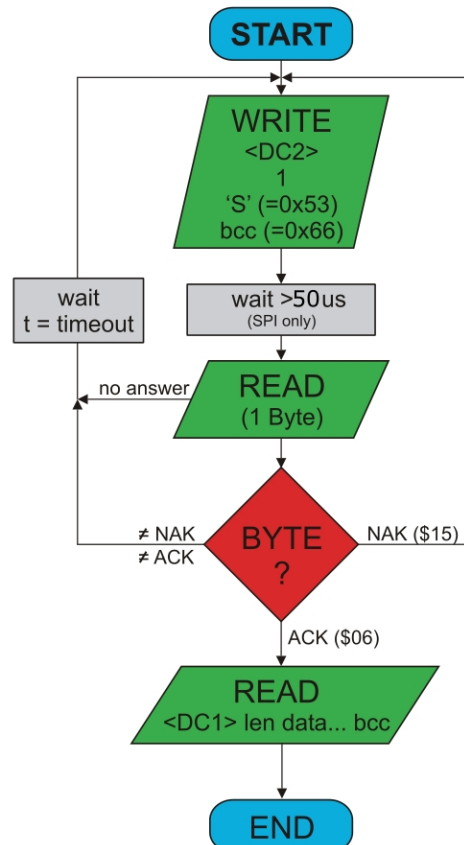
Hinweis: Das <ACK> muss aktiv eingelesen werden (SPI und PC). Empfängt der Hostrechner keine Quittierung, so ist mindestens ein Byte verloren gegangen. In diesem Fall muss die eingestellte Timeoutzeit abgewartet werden, bevor das Paket komplett wiederholt wird.

Die Anzahl der Nutzdaten pro Paket kann max. 128 Byte betragen. Befehle die größer sind (z.B. Bilder 'ESC UL..') müssen auf mehrere Pakete aufgeteilt werden. Die Nutzdaten in den einzelnen Paketen werden nach korrektem Empfang vom Displaymodul wieder zusammengefügt.

Send

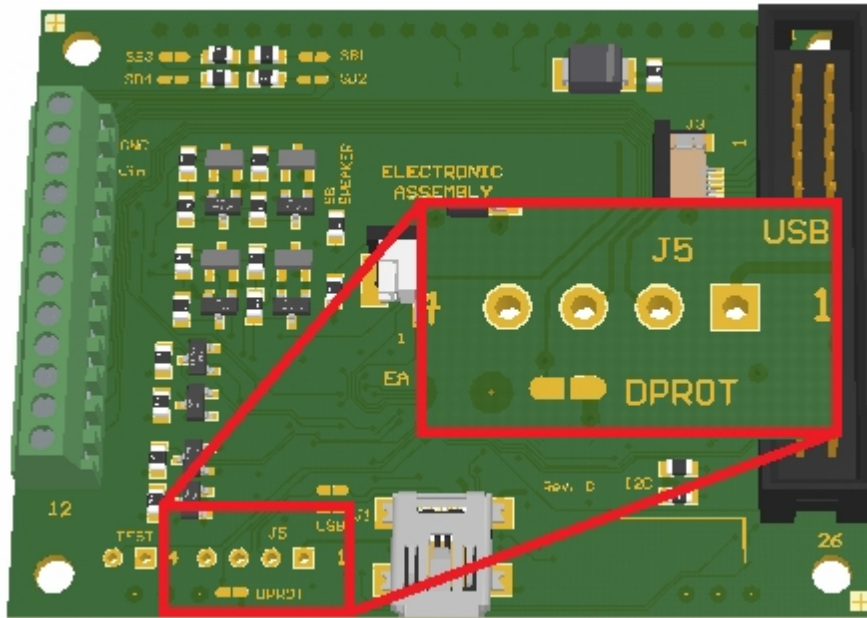


Receive

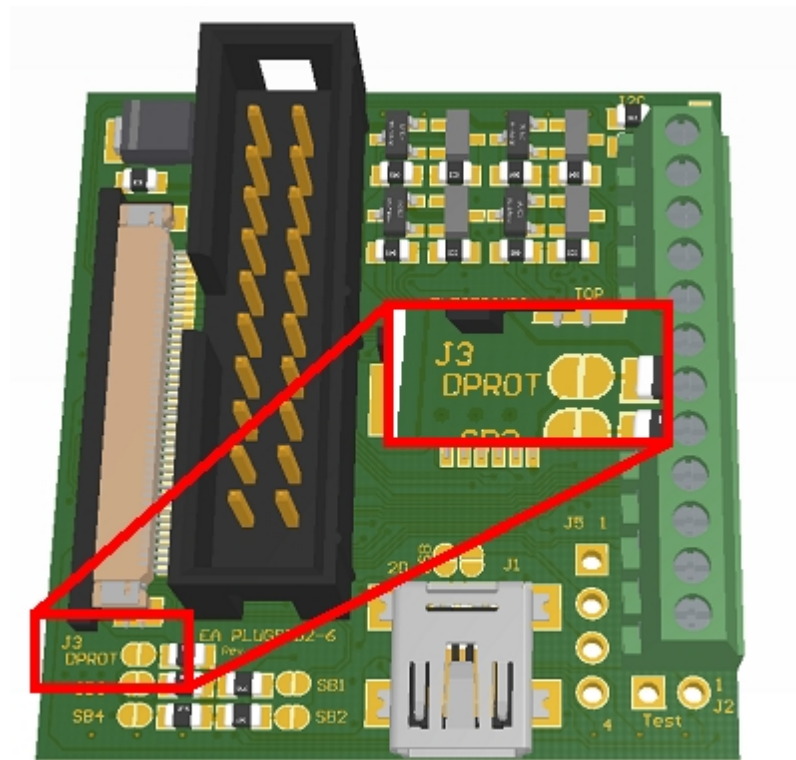


Deaktivieren des Small Protokolls

Für Testzwecke kann das Small Protokoll durch Schließen der Lötbrücke DPROT deaktiviert werden. Es wird dringend empfohlen das Protokoll im Normalbetrieb zu aktivieren. Empfangspufferüberläufe würden ansonsten nicht detektiert.



Lötbrücke zum Deaktivieren des Small Protokolls (EA PLUG128-6)



Lötbrücke zum Deaktivieren des Small Protokolls (EA PLUG128-6)

Anmerkung: Deaktivieren Sie das Protokoll nicht, wenn sie USB als Kommunikationsschnittstelle verwenden.

Small Protokoll

Eingerahmt von <DC1>, der Anzahl der Daten "len" und der Prüfsumme "bcc" werden die jeweiligen Nutzdaten übertragen. Als Antwort sendet das Display <ACK> zurück.

1. Übertragung von Befehlen oder Daten zur EA PLUG-Serie

Dieser Protokoll-Befehl überträgt Daten zum Display. Es können mehrere Grafikbefehle in ein Protokollpaket verpackt werden. Sind die Daten größer als die maximale Paketgröße können die Daten auf mehrere Pakete aufgeteilt werden. Das Modul fügt die einzelnen Datenpakete wieder zusammen.

Modul empfängt	DC1 0x11	Länge (8 Bit) 0xXX	Daten..... 0x....	bcc (8 Bit) 0xXX
Modul sendet	ACK 0x06			

2. Inhalt des Sendepuffers anfordern

Fallen Daten im Modul an, werden diese im Sendepuffer des Moduls gespeichert. Über die seriellen Schnittstellen können die Daten angefordert werden. Ob Daten verfügbar sind kann über den Pin 13 SBUF herausgefunden werden, oder aber die übergeordnete Steuerung fragt zyklisch die Daten ab (polling).

Modul empfängt	DC2 0x12	Länge (8 Bit) 0x01	'S' 0x53	bcc (8 Bit) 0x66
Modul sendet	ACK 0x06			
Modul sendet	DC1 0x11	Länge (8 Bit) 0xXX	Daten..... 0x....	bcc (8 Bit) 0xXX

3. Letztes Datenpaket wiederholen

Ist ein Empfangenes Packet des Moduls fehlerhaft (falsche Länge oder Prüfsumme) kann es erneut angefordert werden:

Modul empfängt	DC2 0x12	Länge (8 Bit) 0x01	'R' 0x52	bcc (8 Bit) 0x65
Modul sendet	ACK 0x06			
Modul sendet	DC1 0x11	Länge (8 Bit) 0xXX	Daten..... 0x....	bcc (8 Bit) 0xXX

4. Pufferinformationen anfordern

Mit diesem Befehl wird nachgefragt, ob Nutzdaten zur Abholung bereit stehen (= Pin 13 SBUF) und wie voll der Empfangspuffer des Displays bereits ist.

Modul empfängt	DC2 0x12	Länge (8 Bit) 0x01	'I' 0x49	bcc (8 Bit) 0x5C	
Modul sendet	ACK 0x06				
Modul sendet	DC2 0x12	Länge (8 Bit) 0x02	Sendepuffer bytes ready (8 Bit) 0xXX	Empfangspuffer bytes free (8 Bit) 0xXX	bcc (8 Bit) 0xXX

5. Protokollinformationen

Hierüber lässt sich die maximale Paketgröße welche das Display senden darf begrenzen. Voreingestellt ist eine Paketgröße mit bis zu 64 Byte Nutzdaten. Weiterhin lässt sich der Timeout in 1/100s einstellen. Der Timeout spricht an, wenn einzelne Bytes verloren gegangen sind. Danach muss das gesamte Paket nochmals übertragen werden.

Modul empfängt Defaultwerte	DC2 0x12	Länge (8 Bit) 0x03	'D' 0x44	Paketgröße Sendepuffer (8 Bit) 0x80	Time-out (8 Bit) in 1/100s 0xC8 (=2 Sekunden)	bcc (8 Bit) 0x20
Modul sendet	ACK 0x06					

6. Protokollinformationen

Abfrage der Protokolleinstellungen (siehe 5.).

Modul empfängt	DC2 0x12	Länge (8 Bit) 0x01	'P' 0x50			bcc (8 Bit) 0x63
Modul sendet	ACK 0x06					
Modul sendet	DC2 0x12	Länge (8 Bit) 0x03	Maximale Paketgröße Sendepuffer (8 Bit) 0x80	Paketgröße Sendepuffer (8 Bit) 0xXX	Time-out (8 Bit) in ms 0xXX	bcc (8 Bit) 0xXX

7. RS485 Adresse selektieren/ deselektieren

Mit diesem Befehl lässt sich ein Modul am RS485-Bus selektieren oder deselektieren. Per Default ist ein Modul mit der Adresse 0x7 immer aktiv.

Modul empfängt Defaultwerte	DC2 0x12	Länge (8 Bit) 0x03	'A' 0x41	'S' (=Selektieren) 'D' (=Deselektieren) 0x53 or 0x44	RS485-Adresse 0xXX	bcc (8 Bit) 0xXX
Modul sendet	ACK 0x06 ----	→ Selektieren → Deselektieren				

8. RS485 Verzögerung Enable Signal

Einige RS485 benötigen eine gewisse Zeit um das Enable-Signal zu verändern und z.B. vom schreibenden in den lesenden Modus umzuschalten. Um eine erfolgreiche Kommunikation mit diesen Geräten zu ermöglichen kann mit diesem Befehl die Umschaltung in den schreibenden Modus verzögert werden.

Modul empfängt Defaultwerte	DC2 0x12	Länge (8 Bit) 0x03	'T' 0x54	Verzögerung in 10 us 0x00 0x00	bcc (8 Bit) 0x69
Modul sendet	ACK 0x06				

9. Schnittstelle exklusiv anfordern

Alle 4 seriellen Schnittstellen werden parallel und gleichwertig behandelt. Um zu gewährleisten, dass eine Abfolge von Protokollpaketen ohne Unterbrechung durchgeführt wird, können die anderen seriellen Schnittstellen deaktiviert werden und die Schnittstelle exklusiv angefordert werden. Dies ist zum Beispiel bei einem Projektupdate über USB sinnvoll.

Modul empfängt	DC2 0x12	Länge (8 Bit) 0x02	'G' 0x47	0x00 = Freigabe 0x01 = Anfordern	bcc (8 Bit) 0xXX
Modul sendet	ACK 0x06				
Modul sendet	DC2 0x12	Länge (8 Bit) 0x01	Aktiv (8 Bit) 0x00 = alle 0x01 = RS232 0x02 = SPI 0x03 = IIC 0x04 = USB		bcc (8 Bit) 0xXX

10. Break-Kommando, Ausführung Unterbrechen/ Beenden

Falls in einem Makro eine Dauerschleife programmiert wurde oder eine Unterbrechung des normalen Ablaufs gewünscht wird, kann mit diesem Befehl gezielt Unterbrochen und Beendet werden. Auch dieser Befehl eignet sich vor allem für Update-Vorgänge.

Modul empfängt Defaultwerte	DC2 0x12	Länge (8 Bit) 0x02	'C' 0x43	break 0x01 = Wait command 0x02 = aktuelles Makrofile 0x04 = Sendepuffer leeren 0x08 = Empfangspuffer leeren 0x10 = Beendet automatischen Port- und Analogscan 0xFF = Alles unterbrechen und beenden	bcc (8 Bit) 0xXX
Modul sendet	ACK 0x06				

11. Software Reset

Das Modul wird mit diesem Protokollbefehl neu gestartet. Je nach Parameter wird nach dem Reset eine andere Startoption gewählt.

Modul empfängt Defaultwerte	DC2 0x12	Länge (8 Bit) 0x02	'B' 0x42	Option 0x00 = Normaler Neustart 0x01 = Neustart im Testmodus 0x02 = Neustart ohne Ausführung des PowerOnMacro's	bcc (8 Bit) 0xXX
Modul sendet	ACK 0x06				

BCC-Berechnung

Für die Berechnung der Prüfsumme wird eine einfache 8-Bit Summenprüfung (Modulo 256) benötigt. Im Folgenden sehen Sie eine typische C-Implementierung.

```
//-----
//function: buffer2bcc()
//input:   ptr data, block length
//output:  Byte bcc
//descr:   calculate bcc for a buffer
//-----
UBYTE buffer2bcc(UBYTE *dat, UBYTE len)
{
    UBYTE bcc = 0;
    while(len--)
        bcc += *dat++;
    return bcc;
}
```

BEFEHLSÜBERSICHT

Die Befehle können zur Laufzeit über die seriellen Schnittstellen übertragen werden oder in sogenannten Macrofiles auf dem internen FLASH-Speicher des Moduls abgelegt sein. In den folgenden Tabellen sind alle Befehle beschrieben.

[Alle Befehle auf einen 'Blick](#)

Terminalfenster #T

#TP	Cursor positionieren
#TC	Cursor ein/aus
#TS	Cursorposition sichern
#TR	Cursorposition wiederherstellen
#TA	Terminal aus
#TE	Terminal ein
#TV	Version ausgeben
#TJ	Projektname anzeigen
#TI	Informationen anzeigen

Displayfunktionen #D

#DO	Display Orientierung
#DR	Reset Display
#DL	Displayinhalt löschen
#DI	Displayinhalt invertieren
#DS	Displayinhalt füllen
#DA	Display ausschalten
#DE	Display einschalten
#DC	Display Clipboard
#DN	Display Normaldarstellung
#DZ	Bildschirmschoner einstellen
#DW	Einstellungen für Modus 1 (Helligkeit)
#DX	Einstellungen für Modus 2 (Animiertes Bild/ Zufallsmuster)
#DV	Einstellungen für Modus 3 (Invertieren)
#DY	Bildschirmschoner (Re)-Trigger

Clipboard #C

#CB	Displayinhalt sichern
#CS	Bereich sichern
#CR	Bereich wiederherstellen
#CK	Bereich kopieren

Geradenfunktionen #G

#GZ	Punktgröße/ Liniendicke
#GV	Verknüpfungsmodus
#GB	Blinkmodus
#GP	Punkt zeichnen
#GD	Gerade zeichnen
#GW	Gerade weiter zeichnen
#GR	Rechteck zeichnen

Bereichsfunktionen #R

#RL	Bereich löschen
#RI	Bereich invertieren
#RS	Bereich füllen
#RM	Bereich mit Füllmuster
#RO	Box zeichnen
#RR	Rahmen zeichnen
#RT	Rahmenbox zeichnen

Textfunktionen #Z

#ZF	Font einstellen
#ZZ	Font-Zoomfaktor
#ZY	Zusätzlicher Zeilenabstand
#ZJ	Leerzeichenbreite einstellen
#ZW	Text-Winkel
#ZV	Text-Verknüpfungsmodus
#ZB	Text-Blinkattribut
#ZL	Zeichenkette ausgeben (linksbündig)
#ZC	Zeichenkette ausgeben (zentriert)
#ZR	Zeichenkette ausgeben (rechtsbündig)
#ZT	Zeichenkette für Terminal

Bildfunktionen #U

#UZ	Bild-Zoomfaktor
#UW	Bild-Winkel
#UV	Bild-Verknüpfungsmodus
#UB	Bild-Blinkattribut
#UC	Bild aus Clipboard
#UI	Internes Bild laden
#UL	Bild laden
#UH	Hardcopy senden

Bargraphfunktionen #B

#BR	Bargraph definieren (rechts)
#BL	Bargraph definieren (links)
#BO	Bargraph definieren (oben)
#BU	Bargraph definieren (unten)
#BD	Bargraph löschen
#BA	Bargraph aktualisieren
#BZ	Bargraph neu zeichnen
#BS	Bargraphwert senden

Blinkbereiche #Q

#QZ	Blinkzeit einstellen
#QL	Blinkattribut löschen
#QI	Invertierender Blinkbereich
#QM	Muster Blinkbereich
#QR	Blinkbereich wiederherstellen
#QE	Phasenverschoben invertierend
#QP	Phasenverschoben mit Muster

Menübox/ Touchmenü #N

#NF	Menü-Font einstellen
#NZ	Menü-Font-Zoomfaktor
#NY	Zusätzlicher Zeilenabstand
#NW	Menü-Winkel
#NT	Touchmenü-Automatik
#ND	Menü definieren und darstellen
#NN	Nächster Eintrag
#NP	Vorheriger Eintrag
#NS	Menüende /Senden
#NM	Menüende/ Makro
#NA	Menüende/ Abbrechen

Makro #M

#MN	Normal Makro ausführen
#MT	Touch Makro ausführen

#MM	Menü Makro ausführen
#MP	Port Makro ausführen
#MB	Bit Makro ausführen
#MV	Analog Makro ausführen
#MG	Makro mit Verzögerung
#ME	Automatische Makros einmal
#MA	Automatische Makros zyklisch
#MJ	Automatische Makros pingpong
#MD	DefineMakroprozess definieren
#MZ	Makroprozess Zeitintervall
#MS	Makroprozesse anhalten

Allgemein #Y, #S, #X, #+

#Y@	Helligkeit speichern
#YH	Helligkeit einstellen
#YN	Helligkeit erhöhen
#YP	Helligkeit reduzieren
#YL	Helligkeit an/aus
#YB	Helligkeit mit Bargraph einstellen
#YS	Summer ein/aus
#SB	Bytes senden
#SV	Version senden
#SJ	Projektname senden
#SI	Interne Infos senden
#X	Warten (Pause)
#+R	RS232 Einstellungen
#+S	SPI Einstellungen
#+I	PC Einstellungen

I/O/ Digital/ PWM

#Y	
#YR	Eingabe Port lesen
#YA	Port Scan an/aus
#YI	Eingabe Port invers
#YD	Bit-Makros für Eingänge umdefinieren
#YM	Ausgabe Ports definieren
#YW	Ausgabe Port schreiben
#YO	PWM Einstellungen

Analogeingang/ Analogausgang #V

#V@	Analogabgleich
#VA	Analog-Abfrage Ein/Aus
#VD	Analogwert senden
#VK	Bereiche/ Grenzen für Analog-Makros
#VB	Bargraph für Analogeingang
#VR	Bargraph aktualisieren
#VF	Benutzerwert Font
#VZ	Benutzerwert Zoom
#VW	Benutzerwert Winkel
#VE	Benutzerwerte/ Skalierung einstellen
#VS	Benutzerwert senden
#VT	Benutzerwert an Terminal
#VG	Benutzerwert anzeigen
#VO	Analogwert ausgeben
#VU	Lookup Tabelle definieren (Rampe)

Touch #A

#AE	Touch-Rahmen Form
#AR	Radiogroup für Schalter
#AF	Beschriftungs Font
#AZ	Beschriftungs-Zoomfaktor
#AY	Zusätzlicher Zeilenabstand
#AW	Beschriftungs-Winkel
#AT	Touch-Taste definieren
#AU	Touch Taste definieren (Bild)
#AK	Touch-Schalter definieren
#AJ	Touch Schalter definieren (Bild)
#AM	Touch-Taste mit Menüfunktion definieren
#AD	Zeichenbereich definieren
#AH	Freien Touchbereich definieren
#AB	Bargraph per Touch einstellbar
#AA	Touch-Abfrage Ein/Aus
#AI	Touch-Tasten Reaktion (automatisch invertieren)
#AS	Touch Tasten Reaktion (Ton)
#AQ	Bargraphwert automatisch in den Sendepuffer stellen
#AN	Touch-Taste invertieren
#AP	Touch-Schalter einstellen
#AX	Touch-Schalter abfragen
#AG	Radiogroup abfragen
#AL	Touch-Bereich löschen (mit Return Code)
#AV	Touch-Bereich löschen (mit Koordinaten)

Antworten

<ESC> <u>A</u>	Button / switch state (value changed)
<ESC> <u>B</u>	Bargraph value (value changed)
<ESC> <u>N</u>	Touch menu (value changed)
<ESC> <u>I</u>	Menu (value changed)
<ESC> <u>P</u>	Port read (value changed)
<ESC> <u>H</u>	Free touch area (value changed)
<ESC> <u>N</u>	Menu
<ESC> <u>B</u>	Bargraph value
<ESC> <u>X</u>	Button / switch
<ESC> <u>G</u>	Radiogroup
<ESC> <u>Y</u>	Port state
<ESC> <u>D</u>	Analog value
<ESC> <u>V</u>	Firmware version
<ESC> <u>J</u>	Project Name
<ESC> <u>I</u>	Internal information
<ESC> <u>U</u>	Hardcopy data
<ESC> <u>W</u>	Formatierte Zeichenkette (ADC/DAC)

TERMINALFENSTER

Im Terminalfenster werden alle empfangenen Daten direkt angezeigt. Dieses Fenster ist nützlich um während der Entwicklungszeit schnell einfache Ausgaben zu erstellen oder Fehlermeldungen zu erhalten.

Befehl	Codes	Anmerkung	
Form feed FF (dec: 12)	^L	Bildschirm wird gelöscht und der Cursor nach Pos. (1,1) gesetzt	
Carriage return CR (dec: 13)	^M	Cursor ganz nach links zum Zeilenanfang	
Line feed LF (dec: 10)	^J	Cursor 1 Zeile tiefer, falls Cursor in letzter Zeile dann wird gescrollt	

Terminalebene Einstellungen

Befehl	Codes	Anmerkung		
Cursor positionieren	ESC T	P Spalte, Zeile	Ursprung links oben ist (1,1)	1,1
Cursor ein/aus		C Sichtbarkeit	Sichtbarkeit =0 (Terminalebene nicht sichtbar); =1 (Terminalebene sichtbar)	1
Cursorposition sichern		S	Die aktuelle Cursorposition wird gesichert	
Cursorposition wiederherstellen		R	Die letzte gesicherte Cursorposition wird wiederhergestellt	
Terminal aus		A	Terminal Anzeige ausschalten; Ausgaben werden verworfen	
Terminal ein		E	Terminal Anzeige einschalten	Ein

Informationen ausgeben

Befehl	Codes	Anmerkung		
Version ausgeben	ESC T	V	Die Versionsnummer wird im Terminal ausgegeben (z.B. "EAPLUG128-6 V1.0 T+")	
Projektname anzeigen		J	Der Makro-Projektname wird im Terminal ausgegeben (z.B. "init / delivery state")	
Informationen anzeigen		I	Das Terminal wird initialisiert und gelöscht, die Software Version, der Makro-Projektname und die CRC-Prüfsummen werden im Terminal ausgegeben.	

DISPLAYFUNKTIONEN (WIRKUNG AUF DAS GESAMTE DISPLAY)

Befehl	Codes			Anmerkung		
Display Orientierung	ESC	D	O	Orientierung	Orientierung =0 (0°); =1 (90°); =2 (180°); =3 (270°)	0
Reset Display			R		Führt einen Reset des Displays durch	

Displayinhalt

Befehl	Codes			Anmerkung		
Displayinhalt löschen	ESC	D	L		Displayinhalt löschen (alle Pixel aus)	
Displayinhalt invertieren			I		Displayinhalt invertieren (invertiert alle Pixel)	
Displayinhalt füllen			S		Displayinhalt füllen (alle Pixel ein)	
Display ausschalten			A		Displayinhalt wird unsichtbar bleibt aber erhalten; Befehle weiterhin möglich	
Display einschalten			E		Displayinhalt wieder sichtbar	On
Display Clipboard			C		Inhalt des Clipboards wird dargestellt. Displayausgaben sind nicht mehr sichtbar	
Display Normaldarstellung			N		Aktuelles Bild wird dargestellt (Normalbetrieb). Alle Ausgaben wieder sichtbar	

Bildschirmschoner

Befehl	Codes			Anmerkung		
Bildschirmschoner einstellen	ESC	D	Z	Modus, Maske	Bildschirmschoner-Modus einstellen: Modus =0 (kein Bildschirmschoner); = 1 (Helligkeit); = 2 (Animiertes Bild/Zufallsmuster); =3 (invertieren) / Retrigger Maske = \$01 (Touch); =\$2 (USB); =\$4 (RS232); =\$8 (SPI); =\$16 (I²C)	0
Einstellungen für Modus 1 (Helligkeit)			W	zzZeit ¹⁾ , Helligkeit ¹ , zzZeit ²⁾ , Helligkeit ²	Bei aktiviertem Bildschirmschoner Modus 1 (#DZ1,...): zzZeit1 und zzZeit2 (16-Bit) stellt die Zeitpunkte in Sekunden ein an denen die Helligkeit (Helligkeit1 and Helligkeit2) aktiv sind. Die Helligkeitsangaben sind jeweils relativ zur aktuellen Helligkeit (0...150%).	
Einstellungen für Modus 2 (Animiertes Bild/ Zufallsmuster)			X	Typ, zzZeit ¹⁾	Bei aktiviertem Bildschirmschoner Modus 2 (#DZ2,...): Typ bestimmt den Animationstyp Typ =0 (Zufallsmuster/Sternenhimmel); =1...255 (Animiertes Bild 1...255) zzZeit (16-Bit): Startzeitpunkt in Sekunden	
Einstellungen für Modus (Invertieren)			V	zzZeit ¹⁾ , zzZeit ²⁾	Bei aktiviertem Bildschirmschoner Modus 3 (#DZ3,...): zzZeit1 (16Bit): Startzeitpunkt in Sekunden zzZeit2 (16Bit): Invertierungsintervall in Sekunden	
Bildschirmschoner (Re)-Trigger			Y	Option	Option =0 (Trigger/Starte Bildschirmschoner); =1 (Retrigger Bildschirmschoner)	

1) 16-Bit Wertebereich (Bei binärer Übertragung erst Low- dann High-Byte)

CLIPBOARD (ZWISCHENSPEICHER FÜR BILDBEREICHE)

Befehl	Codes			Anmerkung		
Displayinhalt speichern	ESC	C	B		Der gesamte Displayinhalt wird als Bildbereich ins Clipboard kopiert	
Bereich sichern			S	x1, y1, x2, y2	Der Bildbereich von x1, y1 bis nach x2, y2 wird ins Clipboard kopiert	
Bereich wiederherstellen			R		Der Bildbereich im Clipboard wird wieder ins Display kopiert	
Bereich kopieren			K	x, y	Der Bildbereich im Clipboard wird ins Display nach x, y kopiert	

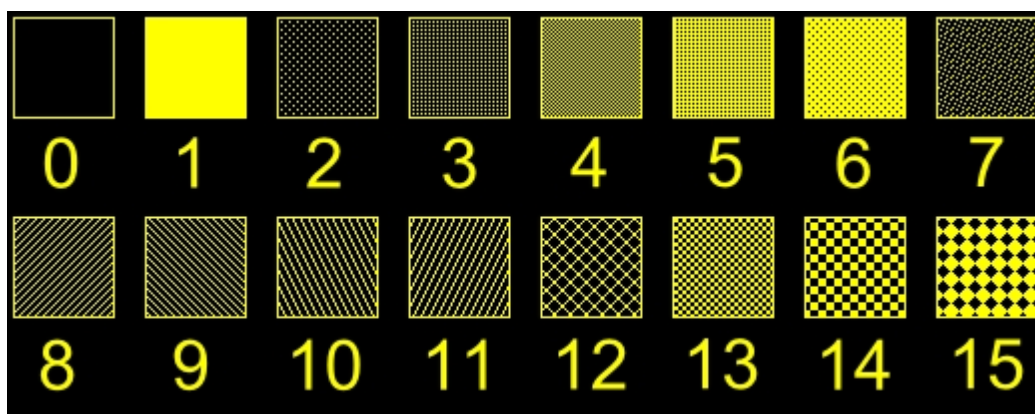
GERADEN- UND BEREICHS-FUNKTIONEN

Geraden und Punkte

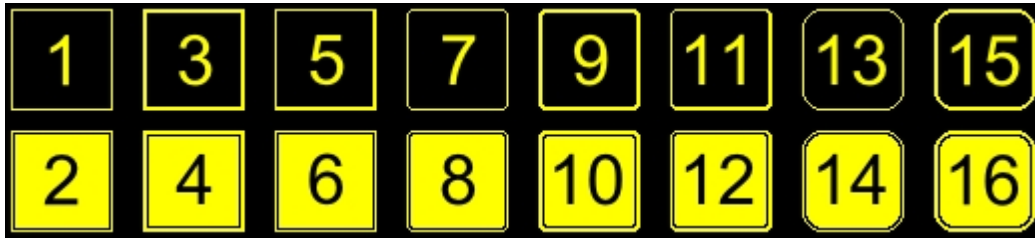
Befehl	Codes		Anmerkung			
Punktgröße/ Liniendicke	ESC	G	Z	n1, n2	n1= x-Punktgröße (1...15); n2 = y-Punktgröße (1...15)	1,1
Verknüpfungsmodus			V	Modus	Zeichen- Modus =1 (setzen); =2 (löschen); =3 (ivers) einstellen	1
Blinkmodus			B	Modus	Blink- Modus =0 (kein blinken); =1 (an/aus); =2 (invertierend blinken); =3 (aus/an phasenverschoben) einstellen	0
Punkt zeichnen	ESC	G	P	x, y	Ein Punkt an die Koordinaten x, y setzen	
Gerade zeichnen			D	x1, y1, x2, y2	Eine Gerade von x1, y1 nach x2, y2 zeichnen	
Gerade weiter zeichnen			W	x, y	Eine Gerade vom letzten Endpunkt bis x, y zeichnen	0,0
Rechteck zeichnen			R	x1, y1, x2, y2	Vier Geraden als Rechteck von x1, y1 nach x2, y2 zeichnen	

Bereichsfunktionen

Befehl	Codes		Anmerkung			
Bereich löschen	ESC	R	L	x1, y1, x2, y2	Einen Bereich von x1, y1 nach x2, y2 löschen (alle Pixel aus)	
Bereich invertieren			I	x1, y1, x2, y2	Einen Bereich von x1, y1 nach x2, y2 invertieren (alle Pixel invertieren)	
Bereich füllen			S	x1, y1, x2, y2	Einen Bereich von x1, y1 nach x2, y2 füllen (alle Pixel ein)	
Bereich mit Füllmuster			M	x1, y1, x2, y2, Muster	Einen Bereich von x1, y1 nach x2, y2 mit Muster zeichnen (immer setzen)	
Box zeichnen			O	x1, y1, x2, y2, Muster	Ein Rechteck von x1, y1 nach x2, y2 mit Muster zeichnen (immer Replace)	
Rahmen zeichnen			R	x1, y1, x2, y2, Rahmen	Einen Rahmen von x1, y1 nach x2, y2 zeichnen (immer setzen)	
Rahmenbox zeichnen			T	x1, y1, x2, y2, Rahmen	Eine Rahmen-Box von x1, y1 nach x2, y2 zeichnen (immer Replace)	



Füllmuster



Rahmen

TEXTFUNKTIONEN

Einstellungen

Befehl	Codes			Anmerkung		
Font einstellen	ESC	Z	F	n1	Font mit der Nummer n1 = 0...15 einstellen	0
Font-Zoomfaktor			Z	x, y	x = x-Zoomfaktor(1...4); y = y-Zoomfaktor (1...4)	1,1
Zusätzlicher Zeilenabstand			Y	Abstand	Zwischen zwei Textzeilen einen zusätzlichen Zeilen- Abstand = 0...15 Pixel einfügen	0
Leerzeichenbreite einstellen			J	Breite	Einstellen der Leerzeichen- Breite =0 (Originale Breite vom Font); =1 (Gleiche Breite wie Zahlen); >=2 (Breite in Pixel)	0
Text-Winkel			W	Winkel	Text Ausgabe- Winkel = 0 (0°); = 1 (90°); = 2 (180°); = 3 (270°)	0
Text-Verknüpfungsmodus			V	Modus	Verknüpfungs- Modus = 1 (setzen); =2 (löschen); = 3 (invers); =4 (Replace); =5 Invers Replace)	4
Text-Blinkattribut			B	Modus	Blink- Modus =0 (kein blinken); =1 (an/aus); =2 (invertierend blinken); =3 (aus/an phasenverschoben)	0

Zeichenkette

Befehl	Codes			Anmerkung		
Zeichenkette ausgeben L: linksbündig C: zentriert R: rechtsbündig	ESC	Z	L	x, y, Text ..., NUL	Eine Zeichenkette an x, y ausgeben; Zeichenkettenende: ' NUL ' (\$00), 'LF' (\$0A) oder 'CR' (\$0D); Mehrere Zeilen werden durch das Zeichen' ' (\$7C) getrennt; Texte zwischen zwei '~' (\$7E): blinken an/aus; Texte zwischen zwei '&' (\$26): blinken aus/an phasenverschoben; Texte zwischen zwei '@' (\$40): blinken invertierend; Das Backslash Zeichen '\' (\$5C, backslash) hebt die Sonderfunktion der Zeichen ' ~@\' auf; z.B. "name\@test.de" => "name@test.de"	
C						
R						
Zeichenkette für Terminal	ESC	Z	T	Text ...	Befehl um eine Zeichenkette (Text ...) in einem Makro an das Terminal ausgeben zu können	

BILDFUNKTIONEN

Einstellungen

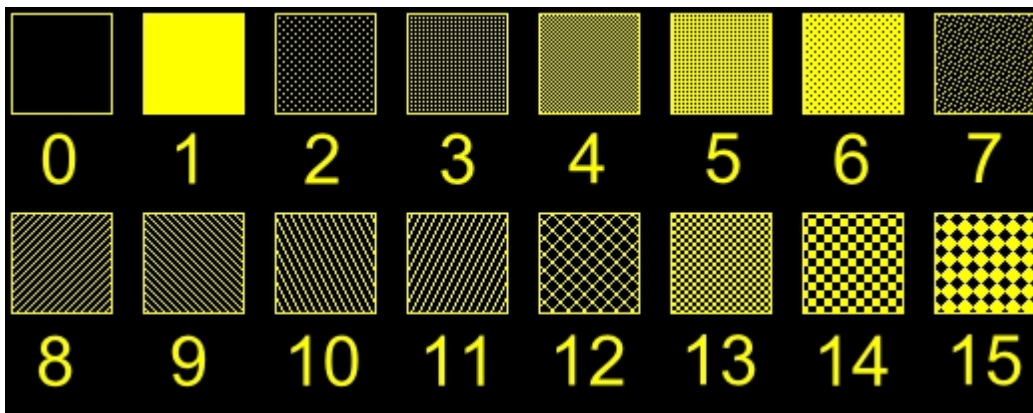
Befehl	Codes		Anmerkungen			
Bild-Zoomfaktor	ESC	U	Z	x, y	x = x-Zoomfaktor (1...4); y = y-Zoomfaktor (1...4)	1,1
Bild-Winkel			W	Winkel	Ausgabe- Winkel = 0 (0°); = 1 (90°); = 2 (180°); = 3 (270°) des Bildes	0
Bild-Verknüpfungsmodus			V	Modus	Bild-Verknüpfungs- Modus = 1 (setzen); =2 (löschen); = 3 (invers); =4 (Replace); =5 (Invers replace)	4
Blink-Blinkattribut			B	Modus	Blink- Modus =0 (kein blinken); =1 (an/aus); =2 (invertierend blinken); =3 (aus/an phasenverschoben)	0

Bilder

Befehl	Codes		Anmerkungen			
Bild aus Clipboard	ESC	U	C	x, y	Der aktuelle Clipboardinhalt wird mit allen Bildattributen nach x, y geladen	
Internes Bild laden			I	x, y, n1	Internes Bild mit der Nummer n1 =0...255 aus dem FLASH nach x, y laden	
Bild laden			L	x, y, BLH daten ...	Ein Bild nach x, y laden; daten... = Bild imBLH-Format	
Hardcopy senden			H	x1, y1, x2, y2	Es wird ein Bild angefordert. Zuerst werden die Breite und Höhe in Pixel und dann die eigentlichen Bilddaten gesendet.	

BARGRAPH/ SCHIEBEREGLER

Befehl	Codes				Anmerkung	
Bargraph definieren	ESC	B	R	n1, x1, y1, x2, y2, aw ew Typ, Muster	Bargraph nach L(inks), R(echts), O(ben), U(nten) mit der Nummer n1 =1..32 definieren. x1, y1, x2, y2 sind das umschließende Rechteck des Bargraphs, aw und ew sind die Werte für 0% und 100%. Typ =0 (Balken); =1 (Balken im Rechteck); Muster =Balkenmuster Typ =2 (Strich); =3 (Strich im Rechteck); Muster = Strichbreite	-
			L			
			O			
			U			
Bargraph löschen		B	D	n1, Sichtbarkeit	Die Definition des Bargraph mit der Nummer n1 wird ungültig. War der Bargraph als Eingabe mit Touch definiert, so wird auch dieses Touchfeld gelöscht. Sichtbarkeit =0 (Bargraph weiterhin sichtbar) =1 (Bargraph wird gelöscht)	
Bargraph aktualisieren	B	A	n1, Wert	Bargraph mit der Nummer n1 auf den neuen Benutzer- Wert setzen und zeichnen		
Bargraph neu zeichnen		Z	n1	Den Bargraph mit der Nummer n1 neu zeichnen		
Bargraphwert senden		S	n1	Den aktuellen Wert des Bargraph mit der Nummer n1 senden		



Füllmuster

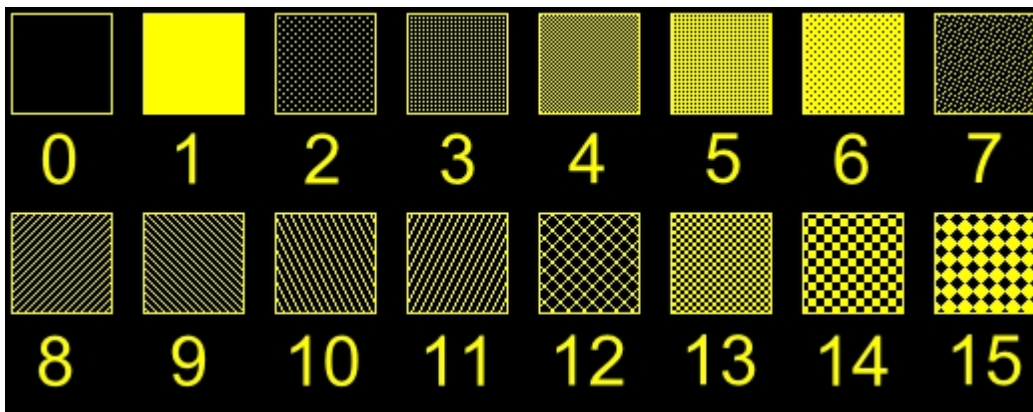
Weitere Bargraph/ Schieberegler Befehle (mit touch)

Befehl	Codes				Anmerkung	
Bargraph per Touch einstellbar	ESC	A	B	nr	Der Bargraph mit der Nummer nr wird zur Eingabe per Touchpanel definiert	
Bargraphwert automatisch in den Sendepuffer stellen			Q	n1	Das automatische Senden eines neuen Bargraphwertes per Toucheingabe wird: n1 =0: deaktiviert n1 =1: neuer Wert wird nach dem Einstellen in den Sendepuffer gestellt n1 =2: jede Änderung landet während des Einstellens im Sendepuffer	1
Helligkeit per Touch einstellbar	ESC	Y	B	nr	Die Helligkeit des Display kann mit dem Bargraph nr eingestellt werden	

Befehl	Codes				Anmerkung	
Bargraph für Analogausgang	ESC	V	B	128, nr	Dem Analogausgang wird der Bargraph mit der Nummer nr =1...20 zugewiesen Bei der Bargraphdefinition sind die Anfangs- und Endwerte (aw, ew) in [mV/20] anzugeben.	

BLINKBEREICHE

Befehl	Codes		Anmerkungen			
Blinkzeit einstellen	ESC	Q	Z	Zeit	Einstellen der Blink-Zeit =1...15 in 1/10s; = 0 (Blinkfunktion deaktivieren)	6
Blinkattribut löschen			L	x1, y1, x2, y2	Löscht das Blinkattribut von x1, y1 bis x2, y2 . Nicht für phasenverobene Blinkbereiche! (Kopiert den Bereich von der Graphik- in die Blinkebene)	
Invertierender Blinkbereich			I	x1, y1, x2, y2	Definiert einen invertierenden Blinkbereich von x1, y1 bis x2, y2 . (Kopiert den Bereich von der Graphik- in die Blinkebene)	
Muster Blinkbereich			M	x1, y1, x2, y2, Muster	Definiert einen Blinkbereich mit Muster (an/aus) von x1, y1 bis x2, y2 (Füllt einen Bereich mit dem Muster in der Blinkebene)	
Blinkbereich wiederherstellen			R	x1, y1, x2, y2	Löscht das phasenverschobene Blinken von x1, y1 bis x2, y2 . Nicht für phasenverobene Blinkbereiche! (Kopiert den Bereich von der Graphik- in die Blinkebene)	
Phasenverschoben invertierend			E	x1, y1, x2, y2	Definiert einen phasenverschobenen invertierenden Blinkbereich von x1, y1 bis x2, y2 . (Kopiert den Bereich von der Graphik- in die Blinkebene)	
Phasenverschoben mit Muster			P	x1, y1, x2, y2, Muster	Definiert einen phasenverschobenen Blinkbereich mit Muster (aus/an) von x1, y1 bis x2, y2 . (Füllt einen Bereich mit dem Muster in der Graphikebene)	



Füllmuster

MENÜBOX/ TOUCHMENÜ

Einstellungen für Menübox/ Touchmenü

Befehl	Codes				Anmerkungen	
Menü-Font einstellen	ESC	N	F	n1	Font mit der Nummer n1 = 0...15 für Menüdarstellung einstellen	0
Menü-Font-Zoomfaktor			Z	x, y	x = x-Zoomfaktor (1...4); y = y-Zoomfaktor (1...4)	1,1
Zusätzlicher Zeilenabstand			Y	Abstand	Zwischen Menüeinträgen zusätzlichen Zeilen- Abstand (0...15 Pixel) einfügen	0
Menü-Winkel			W	Winkel	Menüdarstellung Winkel =0 (0°); =1 (90°)	0
Touchmenü-Automatik			T	Typ	Typ =1: (Touchmenü öffnet automatisch); =0 (Touchmenü öffnet nicht automatisch stattdessen wird die Anforderung 'ESC T0' zum Öffnen an den Hostrechner gesendet. Dieser kann dann mit 'ESC N T 2' das Touchmenü öffnen)	1

Menübox Befehle (Steuerung mit Tasten nicht per Touch)

Befehl	Codes				Anmerkungen	
Menü definieren und darstellen	ESC	N	D	x, y, n1, Text ..., NUL	Ein Menü wird ab der Ecke x, y mit dem aktuellen Menüfont gezeichnet. n1 = aktuell invertierter Eintrag (z.B. 1 = erster Eintrag) Text... = Zeichenkette mit den Menüeinträgen. Die einzelnen Einträge sind durch Zeichen ' ' (\$7C, dez:124) getrennt z.B. "Eintrag1 Eintrag2 Eintrag3" Der Hintergrund des Menüs wird automatisch gesichert. Ist bereits ein Menü definiert, wird dieses automatisch abgebrochen+entfernt.	
Nächster Eintrag			N		Der nächste Eintrag wird invertiert oder bleibt am Ende stehen	
Vorheriger Eintrag			P		Der vorherige Eintrag wird invertiert oder bleibt am Anfang stehen	
Menüende/ Senden			S		Das Menü wird entfernt und durch den ursprünglichen Hintergrund ersetzt. Der aktuelle Eintrag wird als Nummer (1..n) gesendet (0 = kein Menü dargestellt)	
Menüende/ Makro			M	n1	Das Menü wird entfernt und durch den ursprünglichen Hintergrund ersetzt. Für Eintrag 1 wird das Menü-Makro n1 aufgerufen, für Eintrag 2 Menü-Makro n1+1 u.usw.	
Menüende/ Abbrechen	ESC	N	A	Das Menü wird entfernt und durch den ursprünglichen Hintergrund ersetzt		

Weitere Menübefehle (mit Touch)

Befehl	Codes			Anmerkungen
Touch-Taste mit Menüfunktion definieren	ESC	A	M	<p>Der Bereich x1, y1 nach x2, y2 wird als Menü-Taste definiert.</p> <p>Down Code:(1...255) Rückgabe/ Touchmakro beim Drücken.</p> <p>Up Code:(1...255) Rückgabe/ Touchmakro beim Menü-Abbruch.</p> <p>MNU Code:(1...255)Rückgabe/ Menümakro+(Eintragsnummer-1) nach Auswahl eines Menü-Eintrags. (down/up Code = 0: Aktivieren/Abbruch wird nicht gemeldet.)</p> <p>Text:= Zeichenkette mit den Tastentext und den Menüeinträgen. Das erste Zeichen bestimmt die Richtung in der das Menü aufklappt (R=rechts L=links O=oben U=Unten). Das zweite Zeichen bestimmt die Ausrichtung des Touchtasten-Textes (C=zentriert L=linksbündig R=rechtsbündig). Die Menü-Einträge sind durch Zeichen ' ' (\$7C,dez:124) getrennt. z.B. "UCTaste Eintrag1 Eintrag2 Eintrag3" Der Tastentext wird mit dem akt. Touchfont und die Menü-Einträge mit dem akt. Menüfont gezeichnet. Der Hintergrund des Menüs wird automatisch im Clipboard gesichert.</p>

MAKRO

Einzelne oder mehrere Befehlsfolgen können als sogenannte Makros zusammengefasst werden. Diese können dann mit den Befehlen gestartet werden.

Makros aufrufen

Befehl	Codes		Anmerkung		
Normal Makro ausführen	ESC	M	N	n1	Das (Normal-)Makro mit der Nummer n1 (0..255) aufrufen (max. 7 Ebenen)
Touch Makro ausführen			T	n1	Das Touch-Makro mit der Nummer n1 (0..255) aufrufen (max. 7 Ebenen)
Menü Makro ausführen			M	n1	Das Menü-Makro mit der Nummer n1 (0..255) aufrufen (max. 7 Ebenen)
Port Makro ausführen			P	n1	Das Port-Makro mit der Nummer n1 (0..255) aufrufen (max. 7 Ebenen)
Bit Makro ausführen			B	Bittyp	Das Bitmakro mit dem Bittyp (0..255) aufrufen (max. 7 Ebenen)
Analog Makro ausführen			V	Analogtyp	Das Analogmakro mit dem Analogtyp (0..255) aufrufen (max. 7 Ebenen)

Bit Makros	
Bittyp	Start des Makros bei...
1..8	fallende Flanke (I/O 1..8)
9..16	steigende Flanke (I/O 1..8)

Analog Makros		
Analogtyp		Start des Makros bei...
AIN1	AIN2	
0	10	jeder Änderung des Analogwertes
1	11	fallendem Analogwert
2	12	steigendem Analogwert
3	13	kleiner als unterer Grenzwert
4	14	größer als unterer Grenzwert
5	15	kleiner als oberer Grenzwert
6	16	größer als oberer Grenzwert
7	17	außerhalb beider Grenzwerte
8	18	innerhalb beider Grenzwerte
9	19	kleiner als anderer Analogwert

Automatische (Normal-)Makros

Befehl	Codes			Anmerkung		
Makro mit Verzögerung	ESC	M	G	n1, Verzögerung	Das (Normal-)Makro mit der Nummer n1 (0..255) mit Verzögerung (in 1/10s) aufrufen. Die Ausführung wird durch Befehle (z.B. durch Empfang oder Touchmakros) gestoppt.	
Automatische Makros einmal			E	n1, n2, Zeit	Makros n1..n2 automatisch einmal abarbeiten; Zeit = Pause in 1/10s. Die Ausführung wird durch Befehle (z.B. durch Empfang oder Touchmakros) gestoppt.	
Automatische Makros zyklisch			A	n1, n2, Zeit	Makros n1..n2 automatisch zyklisch abarbeiten; Zeit = Pause in 1/10s. Die Ausführung wird durch Befehle (z.B. durch Empfang oder Touchmakros) gestoppt	
Automatische Makros pingpong			J	n1, n2, Zeit	Makros von n1..n2..n1 (PingPong) abarbeiten; Zeit = Pause in 1/10s. Die Ausführung wird durch Befehle (z.B. durch Empfang oder Touchmakros) gestoppt	

Makro Prozesse

Befehl	Codes			Anmerkung		
Makroprozess definieren	ESC	M	D	nr, Typ, n1, n2, Zeit	Ein Makroprozess mit der Nummer nr (1..4) wird definiert (1 = höchste Priorität). Die (Normal-)Makros n2 bis n3 werden nacheinander alle Zeit /10s ausgeführt Typ =1 (einmal); =2 (zyklisch); =3 (ping pong n1..n2..n1)	
Makroprozess Zeitintervall			Z	nr, Zeit	Dem Makroprozess mit der Nummer nr (1..4) wird eine neue Zeit /10s zugeordnet. Ist die Zeit = 0 so wird die Ausführung angehalten.	
Makroprozesse anhalten			S	n1	Alle Makroprozesse werden mit n1=0 gestoppt und mit n1=1 wieder gestartet. z.B. um Einstellungen und Ausgaben über die Schnittstelle ungestört auszuführen.	1

ALLGEMEIN

Helligkeit

Befehl	Codes				Anmerkung	
Helligkeit speichern	ESC	Y	@		Aktuellen Helligkeitswert im FLASH abspeichern	
Helligkeit einstellen			H	Helligkeit	Helligkeit =0%...150% einstellen	100
Helligkeit erhöhen			N		Aktuellen Helligkeitswert erhöhen	
Helligkeit reduzieren			P		Aktuellen Helligkeitswert verringern	
Helligkeit an/aus			L	n1	Helligkeit n1 =0 (aus); =1 (an); =2...255: Display für n1 /10s einschalten.	1
Helligkeit mit Bargraph einstellen			B	nr	Helligkeit mit Bargraph nr einstellen	

Sende-Befehle

Befehl	Codes				Anmerkung	
Bytes senden	ESC	S	B	anz, Daten ...	Es werden anz (=1 to 255) Bytes zum sendepuffer gesendet; Im Quelltext der Makroprogrammierung darf die Anzahl anz nicht angegeben werden, diese wird vom PLUG-Compiler automatisch eingetragen.	
Version senden			V		Version wird als String gesendet (z.B. "EAPLUG128-6 V1.0 T+") <i>Sendepuffer</i>	
Projektname senden			J		Es wird der Makro-Projektname as String gesendet (z.B. "init / delivery state") <i>Sendepuffer</i>	
Interne Infos senden			I		Es werden interne Informationen vom PLUG gesendet (landen im Sendepuffer)	

Sonstige Befehle

Befehl	Codes				Anmerkung		
Summer ein/aus	ESC	Y	S	n1	Summerausgang (Pin n1 =0 (aus); =1 (ein); =2...255 (Ein für n1 /10s)	0	
Warten (Pause)	ESC	X		Zeit	Zeit /10s abwarten bevor der nächste Befehl ausgeführt wird.		
RS232 Einstellungen	ESC	+	R	bbBaudrate ¹ , RS485, Flash	RS232: Standardeinstellungen: Baudrate = 115200; RS485 = 0-255 (7 ist Standard); Flash =0 (Einstellungen sind nur bis zum nächsten Reset gültig); =1 (Einstellungen dauerhaft abspeichern)		
					Baud	Fehler	
					2400	+0.16	
					4800	+0.16	
					9600	+0.16	
					19200	+0.16	
					38400	+0.16	
57600	+0.16						
115200	+0.16						

Befehl	Codes			Anmerkung		
SPI Einstellungen			S	Modus, Datenreihenfolge, Flash	SPI-Modus[0...3] und Datenreihenfolge (=0 MSB first; =1 LSB first) setzen. Flash =0 (Einstellungen sind nur bis zum nächsten Reset gültig), =1 (Einstellungen dauerhaft abspeichern). SPI Standardeinstellungen: Modus 3 MSB.	
I ² C Einstellungen			I	Adresse, Flash	I ² C Adresse einstellen. Flash =0 (Einstellungen sind nur bis zum nächsten Reset gültig); =1 (Einstellungen dauerhaft abspeichern). Standard I ² C-Adresse: 0xDE (0x6F)	

1) 32-Bit Wertebereich (Bei binärer Übertragung erst Low- dann High-Byte)

I/O/ DIGITAL/ PWM

Befehle	Codes			Anmerkung		
Eingabe Port lesen	ESC	Y	R	n1	n1 =0 (Eingabe-Ports als Binärwert einlesen); =1..8 (Eingabe-Port n1 einlesen)	
Port Scan an/aus			A	n1	Der automatische Scan des Eingabe-Port wird n1 =0 (deaktiviert); =1 (aktiviert)	1
Eingabe Port invers			I	n1	Der Eingabe-Port wird n1 = 0 (normal ausgewerter); =1 invertiert ausgewertet)	0
Bit-Makros für Eingänge undefinieren			D	n1, n2, n3	Eingang n1 =1..8 wird bei fallender Flanke n2 =0 das Bitmakro n3 =0..255 zugewiesen Eingang n1 =1..8 wird bei steigender Flanke n2 =1 das BitMacro n3 =0..255 zugewiesen	
Ausgabe Port definieren			M	Maske	Port binär definieren Maske (1=Ausgang; 0=Eingang)	0
Ausgabe Port schreiben			W	n1,n2	n1 =0: Alle Ausgabe Ports setzen n2 (=Binärwert) n1 =1..8: Ausgabeport zurücksetzen n1 (n2 =0); setzen (n2 =1); invertieren (n2 =2)	
PWM Einstellungen	ESC	Y	O	0, ffFrequenz*, Highphase, Lowphase	PWM ffFrequenz (16-Bit; 2...24kHz) setzen Highphase / Lowphase = Verhältnis High-phase zu Low-phase z.B. 2,1 führt zu 66% high und 33% low	

*16-Bit Wertebereich (Bei binärer Übertragung erst Low- dann High-Byte)

Portmaske (binär):

B7	B6	B5	B4	B3	B2	B1	B0
I/O 8	I/O 7	I/O 6	I/O 5	I/O 4	I/O 3	I/O 2	I/O 1

Wert nach PowerOn

0 0 0 0 0 0 0 0

ANALOGEINGANG/ ANALOGAUSGANG

Befehlsgruppe um den Analogeingang und -ausgang des Moduls zu parametrisieren und auszulesen. Das Modul hat zwei 12-Bit Analogeingänge und einen 8-Bit Analogausgang.

Analogeingang

Befehle	Codes		Anmerkung		
Analogabgleich	ESC	V	@ Kanal, xx1 ¹⁾	Der Abgleich für die Analogkanäle erfolgt folgendermaßen: 1.) Definierte Spannung (1..3.1V) an AIN1 (Kanal 1) oder AIN2 (Kanal 2) anlegen 2.) Befehl unter Angabe des Kanals =1...2 und xx1 =Spannungswert (16-Bit) in [mV] z.B. 3.1V an AIN1; Befehl: '#V@1,3100;	
Analog-Abfrage Ein/Aus			A n1	Der automatische Scan der Analogkanäle wird n1 =0 (deaktiviert); =1 (aktiviert)	0
Analogwert senden			D Kanal	Es wird der Wert in [mV] vom Analog- Kanal =1...2 gesendet (landet im Sendepuffer)	
Bereiche/ Grenzen für Analog-Makros			K Kanal, untere Grenze, obere Grenze, Hyterese	Zwei Grenzen für Analog- Kanal =1...2 einstellen. Mit Hilfe dieser Grenzen können bei Über- und/oder Unterschreiten diverse Analog-Makros automatisch gestartet werden. untere Grenze in [mV/20]; obere Grenze in [mV/20]; Hysterese in [mV]	0,0,0
Bargraph für Analogeingang			B Kanal, nr	Dem Analogkanal Kanal =1...2 wird der Bargraph mit der Nummer nr =1...20 zugewiesen Bei der Bargraphdefinition sind die Anfangs- und Endwerte (aw, ew) in [mV/20] anzugeben.	
Bargraph aktualisieren			R Kanal	Alle definierten Bargraphen für Analog- Kanal =1...2 aktualisieren	
Benutzerwert Font	ESC	V	F Kanal, n1	Font für Analog- Kanal =1..2 mit der Nummer n1 einstellen	0
Benutzerwert Zoom			Z Kanal, x, y	Zoomfaktor für Analog- Kanal =1...2 einstellen; x = X-Zoom(1...4); y = Y-Zoom (1...4)	1,1
Benutzerwert Winkel			W Kanal, Winkel	Analog- Kanal =1..2; Schrift- Winkel =0 (0°); =1 (90°)	0
Benutzerwerte/ Skalierung einstellen			E Kanal, Format String ..., NUL	Benutzerwert für Analog- Kanal =1..2 einstellen. Jeweils 2 Analogwerten (0..3100mV) wird ein Benutzerwert max. Anzeigeumfang 4 1/2 Stellen 19999 + Dezimalpunkt('.') oder ',' + evtl. Vorzeichen '-' zugeordnet. Format String: "mV1=Benutzerwert 1;mV2=Benutzerwert 2"; 'NUL' (\$) = Stringende Beispiel: Anzeige soll bei 2000mV "-123.45" und bei 1000mV "0.00" sein Format String: "2000=-123.45;1000=0"	
Benutzerwert senden			S Kanal	Aktuellen Benutzerwert für Analog- Kanal =1...2 senden (landet im Sendepuffer)	
Benutzerwert an Terminal			T Kanal	Aktuellen Benutzerwert für Analog- Kanal =1...2 zum Terminal ausgeben	

Befehle	Codes		Anmerkung	
Benutzerwert anzeigen		G	Kanal, x, y	Aktuellen Benutzerwert für Analog- Kanal =1...2 rechtsbündig an x, y ausgeben

Analogausgang

Befehle	Codes		Anmerkung			
Benutzerwert Font	ESC	V	F	128, n1	Font für Analogausgang mit der Nummer n1 einstellen	0
Benutzerwert Zoom			Z	128, x, y	Zoomfaktor für Analogausgang einstellen; x = X-Zoom(1...4); y = Y-Zoom (1...4)	1,1
Benutzerwert Winkel			W	128, Winkel	Analogausgang; Schrift- Winkel =0 (0°); =1 (90°)	0
Benutzerwerte/ Skalierung einstellen			E	128, Format String ..., NUL	Benutzerwert für Analogausgang einstellen. Jeweils 2 Analogwerten (0..3100mV) wird ein Benutzerwert max. Anzeigebereich 4 1/2 Stellen 19999 + Dezimalpunkt('.') oder ',' + evtl. Vorzeichen '-' zugeordnet. Format String: "mV1=Benutzerwert 1;mV2=Benutzerwert 2"; 'NUL' ('\$00') = Stringende Beispiel: Anzeige soll bei 2000mV "-123.45" und bei 1000mV "0.00" sein Format String: "2000=-123.45;1000=0"	
Analogwert ausgeben	ESC	V	O	128, Wert, Zeit, Rampe	Analog- Wert (Pin 14/ 8-Bit) ausgeben. Der aktuelle Analogwert erhöht oder verringert sich je nach gewählter Rampe (=0 linear; =1 beschleunigend → Linear; =2 linear → abbremsend; =3 beschleunigend → linear → abbremsend; =4 Benutzerdefiniert (#VU...) in der Zeit /10s	
Lookup Tabelle definieren (Rampe)			U	128, n1,...,n100	Setze 100 Werte für die benutzerdefinierte Rampe (Wertebereich 0...100)	
Bargraph für Analogausgang			B	128, nr	Dem Analogausgang wird der Bargraph mit der Nummer nr =1...20 zugewiesen Bei der Bargraphdefinition sind die Anfangs- und Endwerte (aw, ew) in [mV/20] anzugeben.	
Benutzerwert senden			S	128	Aktuellen Benutzerwert für Analogausgang senden (landet im Sendepuffer)	
Benutzerwert an Terminal			T	128	Aktuellen Benutzerwert für Analogausgang zum Terminal ausgeben	
Benutzerwert anzeigen			G	128, x, y	Aktuellen Benutzerwert für Analogausgang rechtsbündig an x, y ausgeben	

1) 16-Bit Wertebereich (Bei binärer Übertragung erst Low- dann High-Byte)

TOUCH

Einstellungen

Befehle	Codes				Anmerkung	
Touch-Rahmen Form			E	Rahmen	Der Rahmentyp für die Darstellung von Touch-Tasten/Schaltern wird eingestellt.	1
Radiogroup für Schalter	ESC	A	R	nr	Innerhalb einer Gruppe ist immer nur 1 Schalter aktiv, alle anderen werden deaktiviert nr=0 : neu definierte Schalter gehören keiner Gruppe an. nr=1..255 : neu definierte Schalter gehören der Gruppe mit der Nummer nr an. Bei Schalter in einer Gruppe wird nur der Down Code beachtet, der Up Code wird ignoriert.	



Rahmenformen

Touch label font

Befehle	Codes				Anmerkung	
Beschriftungs Font			F	nr	Font mit der Nummer nr (0...15) für Touchtastenbeschriftung einstellen	0
Beschriftungs-Zoomfaktor	ESC	A	Z	x, y	x =X-Zoomfaktor (1...4); y = Y-Zoomfaktor (1...4)	1,1
Zusätzlicher Zeilenabstand			Y	n1	Zwischen zwei Textzeilen n1 (0...15) Pixel als zusätzlichen Zeilenabstand einfügen	0
Beschriftungs-Winkel			W	n1	Text-Ausgabewinkel; n1 =0 (0°); =1 (90°)	0

Taster/Schalter

Befehle	Codes				Anmerkung	
Touch-Taste definieren (Taste is gedrückt solange der Touch berührt wird)	ESC	A	T	x1, y1, x2, y2, Down Code, Up Code, Text ..., NUL	'T': Der Bereich von x1, y1 nach x2, y2 wird als Taste definiert. 'U': Das Bild Nummer. n1 wird nach x1, y1 geladen und als Taste definiert. Down Code :(1...255) Rückgabe/Touchmakro beim Drücken. Up Code : (1...255) Rückgabe/Rouchmakro beim Loslassen der Taste. (down/up code = 0 drücken/loslassen wird nicht gemeldet). Text : Das erste Zeichen bestimmt die Ausrichtung des Textes(C=zentriert, L=linksbündig, R=rechtsbündig). danach folgt eine Zeichenkette die mit dem aktuellen Touch-Font in der Taste platziert wird. Mehrzeilige Texte werden mit dem Zeichen ' ' (\$7C, dec: 124) getrennt; NUL : (\$00) = Zeichenkettenende	
			U	x1, y1, n1, Down Code, Up Code, Text ..., NUL		

Befehle	Codes			Anmerkung
Touch-Taste invertieren			N Code	Die Touch-Taste mit dem zugeordnetem Return- Code wird manuell invertiert.
Touch-Schalter definieren (Zustand der Schalter toggelt nach jeder Berührung)			K x1, y1, x2, y2, Down Code, Up Code, Text ..., NUL	'K': Der Bereich von x1, y1 nach x2, y2 wird als Schalter definiert. 'J': Das Bild Nummer. n1 wird nach x1,y1 geladen und als Schalter definiert.
			J x1, y1, n1, Down Code, Up Code, Text ..., NUL	Down Code :(1...255) Rückgabe/Touchmakro beim Drücken. Up Code : (1...255) Rückgabe/Rouchmakro beim Loslassen der Taste. (down/up code = 0 drücken/loslassen wird nicht gemeldet). Text : Das erste Zeichen bestimmt die Ausrichtung des Textes(C=zentriert, L=linksbündig, R=rechtsbündig). danach folgt eine Zeichenkette die mit dem aktuellen Touch-Font in der Taste platziert wird. Mehrzeilige Texte werden mit dem Zeichen ' ' (\$7C, dec: 124) getrennt; NUL : (\$00) = Zeichenkettenende
Touch-Schalter einstellen			P Code, n1	Zustand des Schalters wird per Befehl geändert (n1=0=aus ; n1=1=ein).

Touch/Zeichenbereich

Befehle	Codes			Anmerkung
Zeichenbereich definieren	ESC	A	D x1, y1, x2, y2, n1	Ein Zeichenbereich wird definiert. Innerhalb der Eck-Koordinaten x1, y1 und x2, y2 kann dann mit der Strichstärke n1 gezeichnet werden.
Freien Touchbereich definieren			H x1, y1, x2, y2	Ein frei benutzbarer Touchbereich wird definiert. Touchaktionen (down, up und drag) innerhalb der Eck-Koordinaten x1, y1 und x2, y2 werden gesendet.

Menüfunktion

Befehle	Codes				Anmerkung	
Touch-Taste mit Menüfunktion definieren	ESC	A	M	x1, y1, x2, y2, Down Code, up Code, MNU Code, Text ..., NUL	<p>Der Bereich x1, y1 nach x2, y2 wird als Menü-Taste definiert.</p> <p>Down Code:(1...255) Rückgabe/Touchmakro beim Drücken.</p> <p>Up Code:(1...255) Rückgabe/Touchmakro beim Menü-Abbruch</p> <p>MNU Code:(1...255) Rückgabe/Menümakro+(EintragsNr-1) nach Auswahl eines Menü-Eintrags. (down/up code = 0: Aktivieren/Abbruch wird nicht gemeldet.)</p> <p>Text:= Zeichenkette mit den Tastentext und den Menüeinträgen. Das erste Zeichen bestimmt die Richtung in der das Menü aufklappt (R=rechts, L=links, O=oben, U=unten). Das zweite Zeichen bestimmt die Ausrichtung des Touch-Tasten-Textes (C=zentriert, L=linksbündig, R=rechtsbündig). Die Menü-Einträge sind durch das Zeichen ' ' (\$7C,dec:124) getrennt (z.B."UCTaste Eintrag1Eintrag2 Eintrag3". Der Tastentext wird mit dem aktuellen Touch-Font und die Menüeinträge mit dem aktuellen menü-Font gezeichnet. Der Hintergrund des Menüs wird automatisch im Clipboard gesichert.</p>	

Globale Einstellungen

Befehle	Codes				Anmerkung	
Touch-Abfrage Ein/Aus	ESC	A	A	n1	Touchabfrage wird n1=0 (deaktiviert); n1=1 (aktiviert)	1
Touch-Tasten Reaktion			I	n1	Automatisches Invertieren beim Berühren der Touch-Taste: n1=0= Aus; n1=1= Ein;	1
			S	n1	Summer piepst kurz beim Berühren einer Touch-Taste: n1=0= Ausf; n1=1= Ein	1

Bargraph/ Slider

Befehle	Codes				Anmerkung	
Bargraph per Touch einstellbar	ESC	A	B	nr	Der Bargraph mit der Nummer nr wird zur Eingabe per Touchpanel definiert	
Bargraphwert automatisch in den Sendepuffer stellen			Q	n1	Das automatische Senden eines neuen Bargraphwertes per Toucheingabe wird: n1=0: deaktiviert n1=1: neuer Wert wird nach dem Einstellen in den Sendepuffer gestellt n1=2: jede Änderung landet während des Einstellens im Sendepuffer	1

Antworten

Befehle	Codes				Anmerkung	
Radiogroup abfragen	ESC	A	G	nr	Der Down Code des aktivierten Schalters aus der Radiogroup mit der Nummer nr wird in den Sendepuffer gestellt	
Touch-Schalter abfragen			X	nr	Zustand des Schalters (Aus=0; Ein=1) wird in den Sendepuffer gestellt	

Touchbereich löschen

Befehle	Codes				Anmerkung	
Touch-Bereich löschen	ESC	A	L	Code, n1	Der Touchbereich mit dem Return- Code (Code =0: alle Touchbereiche) wird aus der Touchabfrage entfernt.. Mit n1 =0, bleibt der Bereich am Display sichtbar; mit n1 =1, wird der Bereich gelöscht.	
			V	x, y, n1	Touchbereich der die Koordinaten x , y umschließt aus der Touchabfrage entfernen.. n1 =0: Bereich bleibt sichtbar; n1 =1: Bereich löschen	

ANTWORTEN/RÜCKMELDUNGEN

Alle Antworten der EA PLUG-Serie werden in einen Sendepuffer gestellt. Über das Small-Protokoll werden diese dann vom Host angefordert. Dies kann per „Polling“ geschehen, oder alternativ dazu zeigt der Pin 20 „SBUF“ mit einem LO-Pegel an, dass Daten zur Abholung bereit stehen.

Selbstständige Antworten (landen im Sendepuffer)

ESC	A	1	Code	Antwort vom Touchpanel wenn eine Taste/Schalter gedrückt wurde. Code = Down or Up Code der Taste/Schalter. Es wird nur gesendet wenn kein Touch-Makro mit der Nr. Code definiert ist!
ESC	B	2	nr, Wert	Nach dem Einstellen eines Bargraph per Touch wird der aktuelle Wert des Bargraphs mit der Nummer nr gesendet."Bargraphwert Senden" muss aktiviert sein (siehe Befehl #AQ). Nach dem Befehl #BS ' wird der aktuelle Wert des Bargraphs mit der Nummer nr gesendet.
ESC	N	1	Code	Nach dem Auswählen eines Menüeintrags per Touch wird der aktuelle Menüeintrag Code gesendet. Es wird nur gesendet, wenn kein Menü-Makro mit der Nr. Code definiert ist. Nach dem Befehl #NS ' wird der aktuell ausgewählte Menüeintrag gesendet. Code=0 : kein Menüeintrag ist ausgewählt..
ESC	T	0		Falls das automatische Öffnen eines Touchmenüs deaktiviert ist (siehe Befehl #NT), so wird diese Anforderung an den Hostrechner gesendet. Dieser kann dann das Touchmenü mit dem Befehl 'ESC N T 2' öffnen.
ESC	P	1	Wert	Nach der Änderung des Eingangs-Port wird der neue 8-Bit Wert gesendet. Der Port-Scan muss aktiviert sein. (siehe Befehl #YA)Es wird nur gesendet wenn kein Port-Makro mit der Nr. Wert definiert ist!
ESC	H	3	Type, x, y	Bei einem freien Touchbereich-Ereignis wird folgendes gesendet: Type =0: Loslassen; =1: Berühren; =2: Drag innerhalb des freien Touchbereiches an den Koordinaten x, y
ESC	X	2	Code, Wert	Nach dem Befehl #AX wird der aktuelle Zustand des Touch-Schalters mit dem Return- Code gesendet (Wert =0 or 1) .
ESC	G	2	nr, Code	Nach dem Befehl #AG ' wird der Code des aktiven Touch-Schalters von der Radiogroup nr gesendet.
ESC	Y	2	nr, Wert	Nach dem Befehl #YR ' wird der angeforderte Eingangs-Port gesendet.: nr=0 : Wert ist ein Binärwert aller Eingänge. nr=1..8 : Wert ist 0 oder 1 je nach Zustands des Eingangs nr
ESC	D	3	nr, Lo-Wert, High-Wert	Nach dem Befehl #VD ' wird der aktuelle Analogwert vom Analogeingang nr=1...2 gesendet. (Value = 0..3100 mV)
ESC	V	Num	Version string ...	Nach dem Befehl #SV ' wird die Version der PLUG-Firmware als Zeichenkette gesendet z.B. "EA PLUGL128-6 V1.0 T+"
ESC	J	Num	Projectname string ...	Nach dem Befehl #SJ ' wird der Makro-Projektname als Zeichenkette gesendet. z.B. "init / delivery state"
ESC	I	21	X-dots, Y-dots, Version, Touchinfo, CRC-ROM, CRC-ROMsoll DF in KB, CRC-DF, CRC-DFsoll, DFlen	Nach dem Befehl #SI ' werden interne Informationen vom PLUG (16-Bit integer Werte LO-HI Byte) gesendet Version : LO-Byte =Versionsnummer Software Touchinfo : LO-Byte = '- +' X-Richtung erkannt; HI-Byte = '- +' Y-Richtung erkannt DFlen : Anzahl benutzter Bytes im FLASH (3 Bytes: LO-, MID- HI-Byte)

ESC	W	Num	Kanal, Formatierte Zeichenkette	Nach dem Befehl #VS wird eine formatierte Zeichenkette des analogen Kanals (ADC: 1..2/ DAC) übertragen
-----	---	-----	---------------------------------------	---








Antworten ohne Längenangabe

ESC	U	L	x, y, Bilddaten...	Nach dem Befehl #UH wird ein Hardcopy gesendet. x, y = Startkoordinaten des Hardcopy (Linke obere Ecke) BLH-Bilddaten: 2 Byte: Breite, Höhe (in Pixel)+ Anzahl Bytes Bilddaten = (((Breite+7)/8)*Höhe
-----	---	---	--------------------	---

BEFEHLSBEISPIELE

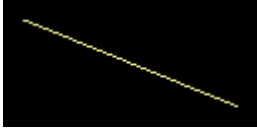
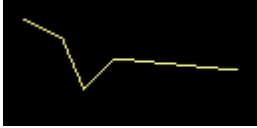

DISPLAYFUNKTIONEN

Bildschirmschoner

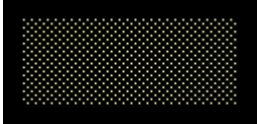


<p>#DZ2,1, #DX0,5,</p>	
<p>#DZ2,1, #DX1,10, EA PICTURE: nr,<file> Das Bild wurde mit dem Kit Editor im internen Speicher des Display abgespeichert (Bildnr.: 1)</p>	
<p>#DZ1,1, #DW10,50,5,50,</p>	
	
	
<p>#DZ3,1, #DV10,5,</p>	
	

GERADEN- UND BEREICHS-FUNKTIONEN

Gerade/ Rechteck zeichnen

#GD10,10,117,53	
#GD10,10,30,20, #GW40,45, #GW55,30, #GW117,35,	
#GR10,10,117,53	

Bereich/ Box zeichnen

#RM10,10,117,53,2	
#RO10,10,117,53,2	
#RR10,10,117,53,15	

TEXTFUNKTIONEN

Einstellungen

#ZF6,	Font Nummer 6 (Swiss 30 Bold proportional)
-------	--

Zeichenkette platzieren



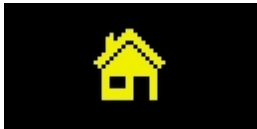
#ZL10,20,"Left"	Left
#ZC63,20,"Center"	Center
#ZR117,20,"Right"	Right

BILDFUNKTIONEN

Einstellungen

#UZ2,2	Zoomfaktor: x=2; y=2
--------	----------------------

Bild laden

<p>#UI17,0,1</p>  <p>PICTURE: nr,<file></p> <p>Das Bild wurde mit dem Kit Editor im internen Speicher des Display abgespeichert (Bildnr.: 1)</p>	
<p>#UL39,7,BLH data</p>	

BLH image data:

00	01	02	03	04	05	06	07	08	09	0A	0B	0C	0D	0E	0F
1B	55	4C	27	07	18	18	00	00	00	00	00	00	00	00	00
00	18	00	01	3C	00	03	7E	00	03	E7	00	03	DB	80	03
BD	C0	07	7E	E0	0E	FF	70	1D	FF	B8	0B	FF	D0	07	FF
E0	07	FF	E0	06	18	60	06	18	60	06	18	60	07	F8	60
07	F8	60	07	F8	60	00	00	00	00	00	00	00	00	00	00

Bildformat (BLH)

Mit dem BitmapEdit (Teil der EA LCD-Tools) können Bilder bearbeitet und in das BLH-Format umgewandelt werden.

Struktur einer Bilddatei im BLH-Format:

Beschreibung	Anzahl an Bytes
Bildbreite	1
Bildhöhe	1
Bilddaten	<p>((Breite+7/8) · Höhe Das Bild wird von oben nach unten gespeichert. Ein Byte steht für 8 horizontale Pixel auf dem Bildschirm. (MSB links; LSB rechts; 0=schwarz; 1=gelb)</p>

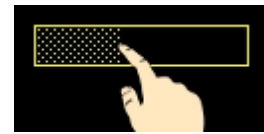
BARGRAPH/SCHIEBEREGLER

#BR1,10,10,117,30,0,100,1,2,

#BA1,60,

#AB1,

Antwort des Displays (Sendepuffer):
<ESC>B

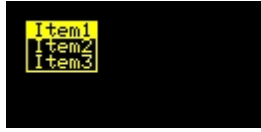
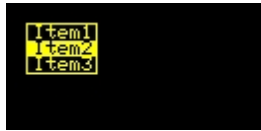


MENÜBOX/ TOUCHMENÜ



Einstellungen

#NF2, (für Menübox)	Font Nummer 2 (6x8 monospaced)
#AF2, (für Touchmenü)	Font Nummer 2 (6x8 monospaced)

Menübox (Steuerung mit Tasten nicht per Touch)

#ND10,10,1,"Item1 Item2 Item3"	
#NN	

Touchmenü



#AM10,10,50,25,0,0,1,"UCMenu Item1 Item2 Item3"	
Antwort des Displays (Sendepuffer): <ESC>N	

TOUCH

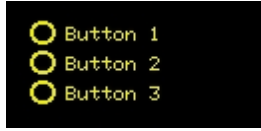
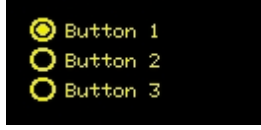

Einstellungen

#AE9,	Rahmentyp: 9
#AE2,	Font Nummer: 2 (6x8 monospaced)

Taster

<p>#AT39,14,89,39,1,0,"Key"</p> <p>Antwort des Displays (Sendepuffer): <ESC>A</p>	
	

Schalter (Radiogroup)

<p>#AR1, #AJ10,10,1,1,0,"RButton 1" #AJ10,25,1,2,0,"RButton 2" #AJ10,40,1,3,0,"RButton 3" #AR0, <input type="radio"/> <input checked="" type="radio"/> PICTURE: nr,<file1>,<file2></p> <p>Die beiden Bilder wurden mit dem Kit Editor im internen Speicher des Display abgespeichert (Bildnr.: 1)</p> <p>#AP1,1,</p> <p>Antwort des Displays (Sendepuffer): <ESC>A</p>	
	
	

Touch/Zeichenbereich

#AD0,0,127,63,2,	
------------------	---

KITEDITOR

Der EA KitEditor vereint 3 Funktionen:

- ... den Editor ermöglicht eine einfache Definition der Makros, Bilder und Schriften wie ein Standard-Texteditor.
- ... den Compiler, der den Text in den Upload-Code übersetzt und Syntaxfehler anzeigt.
- ... den Sender, der die richtige Verbindung zum Modul sucht und die Daten in die EA PLUG-Serie hochlädt.

Der KitEditor ist Teil der EA LCD-Tools. Dazu gehören auch die notwendigen Compiler und andere Tools wie der BitmapEdit oder das LCDterminal. Sie können die LCD-Tools von unserer Website herunterladen.

ELEKTRISCHE SPEZIFIKATION ALLGEMEIN ($T_A=20^{\circ}\text{C}$; $V_{in}=3,3\text{V}$)

Wert	Zustand	min.	typ.	max.	Einheit
Betriebstemperatur	ohne Touchpanel	-40	-	80	$^{\circ}\text{C}$
Betriebstemperatur	mit Touchpanel	-20	-	70	$^{\circ}\text{C}$
Lagertemperatur		-30	-	80	$^{\circ}\text{C}$
Lagerungsfeuchtigkeit	<40 $^{\circ}\text{C}$	-	-	80	% RH
Betriebsspannung V_{in}		3,1	3,3	5,25	V
Ausgangsspannung V_{out}		-	-	3,1	V
		-	-	100	mA
Unterer Spannungspegel Einänge		-0,3	-	0,93	V
Oberer Spannungspegel Eingänge (außer AIN1 / AIN2)		2,32	-	3,4	V
Oberer Spannungspegel AIN1 / AIN2		2,03	-	3,4	V
Unterer Spannungspegel Ausgänge		-	-	0,6	V
Oberer Spannungspegel Ausgänge		2,4	-	-	V
Ausgänge (I/O1...I/O4) (Open Collector)	low level	-	-	360	mA
	high level ¹⁾	-	100	-	k Ω
Ausgangsstrom (I/O5...I/O8) (Push/ Pull, CMOS)		-	-	5	mA
I/O Pull-up Widerstand	vorhanden	10	20	100	k Ω
I ² C-bus Pull-up	vorhanden	-	4,7	-	k Ω

1) Lötbrücken (SB1...SB4) schließen um Pull-up zu aktivieren

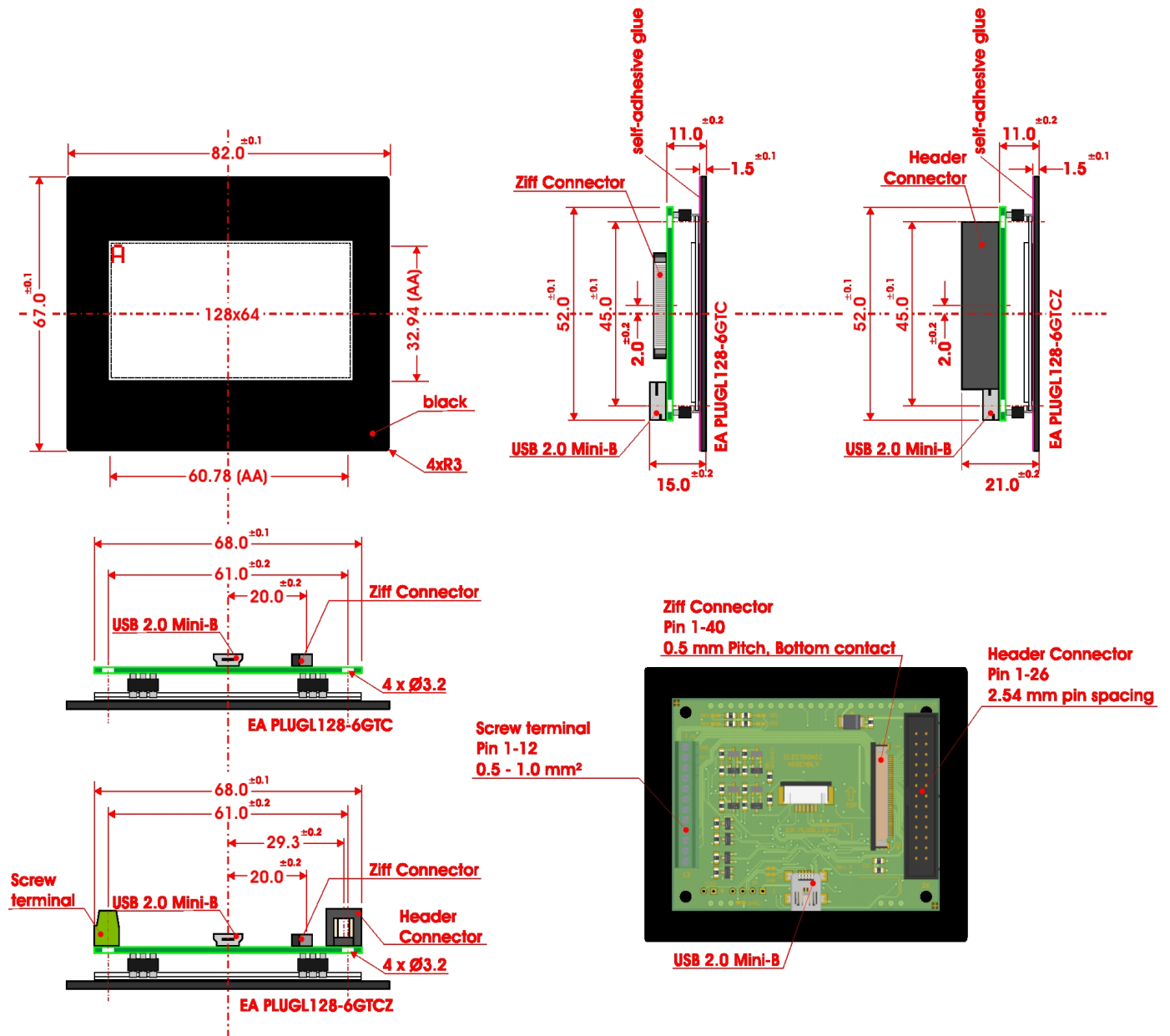
ELEKTRISCHE SPEZIFIKATION PLUGL128-6

Wert	Zustand	min.	typ.	max.	Einheit
Versorgungsstrom @3,3V	alle Pixel ein	-	176	-	mA
	alle Pixel aus	-	20	-	mA
	Display aus	-	15	-	mA
Versorgungsstrom @5V	alle Pixel ein	-	114	-	mA
	alle Pixel aus	-	19	-	mA
	Display aus	-	15	-	mA

ELEKTRISCHE SPEZIFIKATION PLUGS102-6

Wert	Zustand	min.	typ.	max.	Einheit
Versorgungsstrom @3,3V	alle Pixel ein	-	137	-	mA
	alle Pixel aus	-	25	-	mA
	Display aus	-	15	-	mA
Versorgungsstrom @5V	alle Pixel ein	-	97	-	mA
	alle Pixel aus	-	23	-	mA
	Display aus	-	15	-	mA

MAßZEICHNUNG EA PLUGL128-6



MAßZEICHNUNG EA PLUGS102-6

