

Dokumentation IF555-3

Dok-Rev. 3.0 vom 21.09.2010

Hardware-Rev. 2.2 vom 04.06.2010

Inhaltsverzeichnis

1	Allgemeine Hinweise.....	5
1.1	Handhabung	5
1.2	Installation	5
1.3	Erklärung	5
1.4	Reparaturen	5
2	Technische Daten.....	6
2.1	Umgebungsbedingungen	6
2.2	Mechanische Abmessungen	6
2.3	Technische Daten	6
3	Inbetriebnahme.....	7
3.1	Einbau	7
3.2	Spannungsversorgung	7
3.3	Steckverbinder	7
3.4	Beschreibung der Taster	8
3.4.1	TA1	8
3.4.2	TA2	8
3.5	Lage der Jumper und Anschlüsse	8
3.6	Beschreibung der Jumper	9
3.6.1	ST3: Programmierspannung für das interne FLASH	9
3.6.2	ST5: Versorgungsspannung des BDM-Steckers	9
3.6.3	ST12 / ST13: RS485 Receiver enable	9
3.6.4	ST14 / ST17: Abschlußwiderstand CAN-Bus	9
3.6.5	ST28: Display invertiert	10
3.6.6	ST29: Display Fontgröße	10
3.6.7	ST39 / ST40: Polarität des analogen Ausgangs Nr. 8	10
3.6.8	ST51-ST54 / ST56-ST59: Digitale Eingänge 13-16	10
3.6.9	ST60-ST63 / ST65-ST68: Digitale Eingänge 29-32	10
3.6.10	ST41-ST48: Analoge Ausgänge Strom/Spannung umschalten	11
3.6.11	ST30: Watchdog Timeout	11
3.6.12	ST31: Watchdog automatisch starten	11
3.7	Jumper des phyCORE-MPC555 Moduls	11
4	Hardwarebeschreibung.....	13
4.1	Verteilung der 24 Volt	13
4.2	Ethernetschnittstelle	13
4.3	Serielle Schnittstellen	13

4.3.1 Schnittstelle A1	13
4.3.1.1 Belegung A1	13
4.3.2 Schnittstelle A2	14
4.3.2.1 Belegung A2	14
4.3.3 Schnittstelle A3	14
4.3.3.1 Belegung A3	14
4.3.4 Schnittstelle A4	14
4.3.4.1 Belegung A4	15
4.4 CAN-Bus	15
4.4.1 Belegung CAN-Bus	15
4.5 Tastatur	15
4.5.1 Belegung Matrixtastatur-Anschluß	15
4.6 LCD	16
4.6.1 Belegung LCD-Anschluß	16
4.7 CompactFlash Memory Card	16
4.8 Digitale Ausgänge	16
4.9 Digitale Eingänge	17
4.10 Analoge Ausgänge	18
4.11 Analoge Eingänge	19
4.12 Leuchtdioden	19
4.13 Batterie/Goldcap	20
4.14 Externer Watchdog	20
5 Programmierung	21
5.1 Adreßbelegung	21
5.2 Interruptquellen	21
5.2.1 Events	22
5.3 Watchdog	22
5.4 Digitale Ausgänge	22
5.4.1 Abbildung QSPI-Transmitregister <-> Ausgänge	23
5.5 Digitale Eingänge	23
5.6 Analoge Ausgänge	23
5.7 Analoge Eingänge	24
5.8 Compact Flash Card	24
5.8.1 Formatieren der CF-Card	24
5.9 Serielle Schnittstellen	25
5.10 Port-Pins der Tastatur	25
6 RTOS-UH im internen FLASH ablegen	26
7 Terminalemulation	27
7.1 Allgemeines	27

7.2 Befehlsvorrat Terminalemulation	27
7.3 Umsetzung der Eingabezeichen	29
7.4 Grafikfunktionen	30
7.4.1 Funktionsumfang und Implementierungsabhängigkeiten	31
7.4.2 Allgemeines	32
7.4.3 Funktionsreferenz	33
7.4.4 Zeichensatz	38

Revisionsliste:

Rev.	Datum	Na.	Änderung
1.0	21.09.2001	Ko	Erstellung
1.1	02.11.2001	Ko	Beschreibung erweitert
1.2	15.01.2002	Ko	Dig. Ausgänge vervollständigt
1.3	31.01.2002	Ko	Jumperung des phyCORE MPC555 ergänzt
1.4	14.02.2002	Ko	Kapitel zum Brennen von RTOS-UH ergänzt
1.5	12.03.2002	Ko	Änderungen für HW Rev. 1.1 (TPU0-7 A<->B getauscht...)
1.6	23.04.2002	Ko	Terminalemulation: Unterschied Text<->Grafik Mode
1.7	10.01.2003	Ko	Events ergänzt
1.8	18.03.2005	Ko	Änderungen Hardware 1.3 ergänzt
1.9	07.05.2005	Ko	div. Ergänzungen
2.0	02.05.2006	Ko	Analoge Eingänge Abbildung auf MPC555 ergänzt
2.1	30.01.2007	Ko	Watchdog Jumper berichtigt
2.2	12.05.2009	Ko	ST54 ergänzt
2.3	27.05.2009	Ko	BOX_READ/WRITE korrigiert, CF-Card formatieren (S. 24)
2.4	11.06.2009	Ha	Beschreibungen Watchdog u. Batteriepufferung ergänzt
3.0	20.09.2010	Ko	Umstellung auf IF555-3

1 Allgemeine Hinweise

1.1 Handhabung

1. Lesen Sie bitte zuerst sorgfältig diese Dokumentation bevor Sie die Hardware auspacken und einschalten. Sie sparen Zeit und vermeiden Probleme.
2. Beachten Sie bitte die Vorsichtsmaßnahmen bei der Handhabung elektrostatisch gefährdeter Hardware.
3. Wenn die Hardware Batterien enthält, legen Sie sie nicht auf elektrisch leitfähige Unterlagen. Die Batterie könnte kurzgeschlossen werden und Schäden verursachen.
4. Achten Sie bitte darauf, daß der spezifizierete Temperaturbereich nicht verlassen wird.

1.2 Installation

1. Überprüfen Sie, ob alle Jumper entsprechend Ihrer Anwendung gesetzt sind.
2. Schalten Sie die Spannungsversorgung der externen Anschlüsse ab, bevor Sie eine Verbindung herstellen.
3. Wenn Sie sicher sind, daß alle Verbindungen korrekt installiert sind, schalten Sie die Spannungsversorgung ein.

1.3 Erklärung

Wir behalten uns das Recht vor, Änderungen, die einer Verbesserung der Schaltung oder des Produktes dienen, ohne besondere Hinweise vorzunehmen. Trotz sorgfältiger Kontrolle kann für die Richtigkeit der hier gegebenen Daten, Schaltpläne, Programme und Beschreibungen keine Haftung übernommen werden. Die Eignung des Produktes für einen bestimmten Einsatzzweck wird nicht zugesichert.

1.4 Reparaturen

Sollte das Produkt defekt sein, so senden Sie es bitte frei in geeigneter Verpackung mit folgender Beschreibung an uns zurück:

- Fehlerbeschreibung
- Trat der Fehler nur unter bestimmten Bedingungen auf?
- Was war angeschlossen?
- Wie sahen die angeschlossenen Signale aus?
- Garantiereparatur oder nicht?

2 Technische Daten

2.1 Umgebungsbedingungen

Umgebungstemperatur (Betrieb)	0-50° C
Umgebungstemperatur (Lagerung)	-20-85° C
rel. Luftfeuchte	max. 95%, nicht kondensierend
Höhe	-300m bis +3000m

2.2 Mechanische Abmessungen

Kartengröße	257 x 222 mm
Anschlüsse	Pfostensteckverbinder

2.3 Technische Daten

Versorgungsspannung:	18-30 Volt DC, max. 2A
Prozessor	phyCORE-MPC555
ext. Speicher	Compact-Flash Card Adapter
Batteriepufferung:	für Echtzeituhr, über Lithium-Batterie oder Gold-Cap-Kondensator
Analogeingänge	32 differenzielle, unipolar, Eingangsempfindlichkeit 0..5V oder anpassbar, Auflösung 10 Bit
Analogausgänge	8 Stück, 0..20mA oder 0-10 Volt über PWM
Digitaleingänge:	32 Stück, 24 Volt über Optokoppler, z.t. als Zähler nutzbar (max. 80 KHz)
Digitalausgänge:	32 Stück 24 Volt, max. 0,4 A
Matrix-Tastatur:	bis 7x8 Tasten, frei konfigurierbar
LCD-Display:	Grafik-Module mit dem Controller T6963
Ser. Schnittstellen	2x 5-Draht RS232, 2x RS422/485
CAN	1x CAN galvanisch getrennt, 1x CAN nicht getr.
Ethernet	10/100BaseT

3 Inbetriebnahme

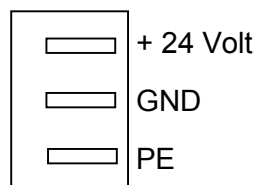
3.1 Einbau

Das MPC555-Trägerboard ist zum Einbau in Schaltschränke oder ähnliche EMV-dichte Gehäuse bestimmt. Die Verkabelung ist EMV-gerecht von entsprechend ausgebildetem Fachpersonal durchzuführen.

3.2 Spannungsversorgung

Die Spannungsversorgung ist für 24 Volt Gleichspannung ausgelegt. Spannungen von 18-30 Volt sind zulässig. Die Spannungsversorgung erfolgt über den 3 poligen Steckverbinder ST6 mit folgender Belegung:

Aufsicht von oben:



Die Baugruppe ist gegen Verpolung der Speisespannung geschützt. Als Steckverbinder wird der Typ MSTB 2,5/3-ST-5,08 der Firma Phoenix Contact eingesetzt.

3.3 Steckverbinder

Die Anschlüsse für die I/O's werden über 2,5 mm Pfostensteckverbinder geführt:

Anschluß	Verbinder
digitale Eingänge	2x 34 polig (je 16 Eingänge)
digitale Ausgänge	4x 10 polig (je 8 Ausgänge)
analoge Eingänge	2x 34 polig (je 16 Eingänge)
analoge Ausgänge	16 polig (8 Ausgänge)
serielle Schnittstellen	4x 10 polig
CAN-Schnittstellen	2x 10 polig
Display	20 polig
Tastatur	16 polig
Ethernet	RJ45 Buchse
Compact Flash Card	50 polig Typ I oder Typ II

3.4 Beschreibung der Taster

Die auf dem Board vorhandenen Taster haben folgende Bedeutung:

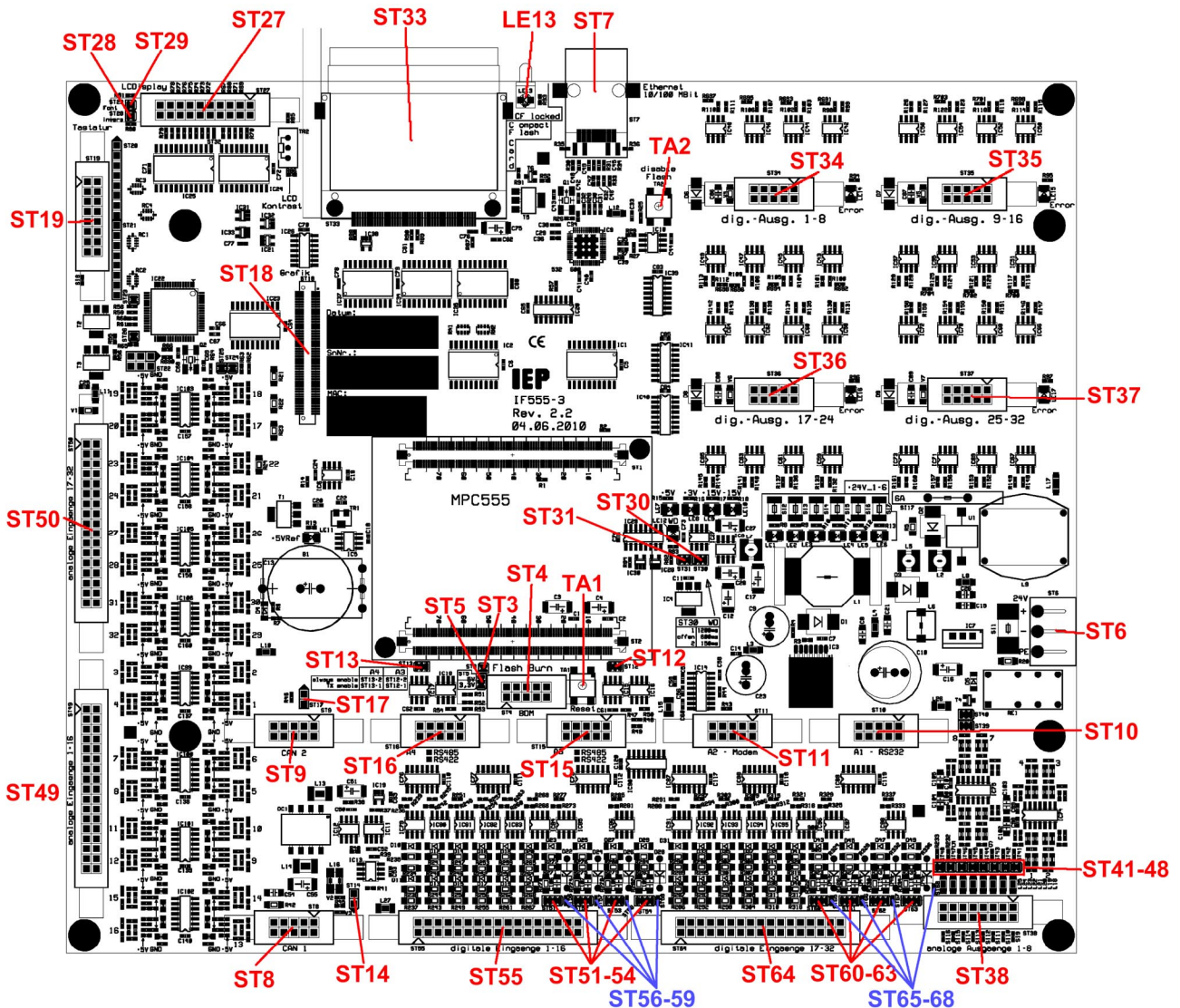
3.4.1 TA1

TA1 löst bei Betätigung einen Reset aus.

3.4.2 TA2

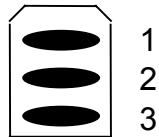
Mit TA2 besteht die Möglichkeit RTOS-UH zu starten, ohne das externe Flash zu scannen. Dazu muß TA2 gleichzeitig mit TA1 (Reset) gedrückt werden. TA1 muß vor TA2 losgelassen werden.

3.5 Lage der Jumper und Anschlüsse



3.6 Beschreibung der Jumper

Die Löt-Jumper werden folgendermaßen gezählt:



3.6.1 ST3: Programmierspannung für das interne FLASH

Mit ST3 wird die Programmierspannung an den MPC555 gelegt. Ohne diese Spannung kann das interne FLASH weder gelöscht noch programmiert werden

Programmierung	Jumper
möglich	1-2
nicht möglich	offen

3.6.2 ST5: Versorgungsspannung des BDM-Steckers

ST5 legt die Versorgungsspannung des BDM-Steckers fest.

Spannung	Jumper
3,3 V	1-2
5 V	2-3

3.6.3 ST12 / ST13: RS485 Receiver enable

ST12 bestimmt das Verhalten des Receivers der Schnittstelle A3 (ST15), ST13 das der Schnittstelle A4 (ST16), wenn die Schnittstelle für RS485-Mode bestückt ist.

Der Receiver kann auf immer „enabled“ gejumpert werden, dann werden auch die vom eigenen Transmitter gesendeten Zeichen empfangen. In der anderen Jumperstellung wird nur empfangen, wenn der eigene Transmitter nicht sendet.

Receiver	Jumper
nicht beim Senden enabled	1-2
immer enabled	2-3

3.6.4 ST14 / ST17: Abschlußwiderstand CAN-Bus

Mit ST14 wird der Abschlußwiderstand des CAN-Bus Kanals 1, mit ST17 von Kanal 2 ein oder ausgeschaltet.

Abschlußwiderstand	Jumper
abgeschaltet	offen
eingeschaltet	1-2

3.6.5 ST28: Display invertiert

Wird ST28 geschlossen, invertiert das angeschlossenen Display seine Anzeige.


Display	Jumper
normal	Offen
invertiert	1-2

3.6.6 ST29: Display Fontgröße

ST29 bestimmt den Font des Displays.

Display Font	Jumper
6x8 Pixel	Offen
8x8 Pixel	1-2

3.6.7 ST39 / ST40: Polarität des analogen Ausgangs Nr. 8


ST39 / ST40 legen die Ausgangspolarität des analogen Ausgangs Nr. 8 fest. Um Kurzschlüsse und Beschädigungen zu vermeiden, dürfen nur die Kombinationen aus der folgenden Tabelle gejumpert werden: 

Polarität	ST39	ST40
positiv	1-2	2-3
negativ	2-3	1-2

3.6.8 ST51-ST54 / ST56-ST59: Digitale Eingänge 13-16

Es wird die Eingangsbeschaltung der digitalen Eingänge DI13-16 konfiguriert. ST51/ST56 konfigurieren DI13, ST52/ST57 konfiguriert D14 usw.


Eingang	ST51-54	ST56-59
massebezogen	1-2 und 3-4	1-2
potentialfrei	2-3	Offen

Bitte beachten Sie, dass sich die Polarität des Eingangs beim potentialfreien Betrieb umkehrt! 

3.6.9 ST60-ST63 / ST65-ST68: Digitale Eingänge 29-32

Es wird die Eingangsbeschaltung der digitalen Eingänge DI29-32 konfiguriert. ST60/ST65 konfigurieren DI29, ST61/ST66 konfiguriert D30 usw.

Eingang	ST60-63	ST65-68
massebezogen	1-2 und 3-4	1-2
potentialfrei	2-3	Offen

Bitte beachten Sie, dass sich die Polarität des Eingangs beim potentialfreien Betrieb umkehrt! 

3.6.10 ST41-ST48: Analoge Ausgänge Strom/Spannung umschalten

Die analogen Ausgänge können 0-20mA liefern. Wird der entsprechende Jumper geschlossen, so wird eine Bürde von 510 Ω auf den Ausgang geschaltet, so dass 0-10V ausgegeben werden.

Ausgang	Jumper
0-20 mA	Offen
0-10 V	1-2

Die Zuordnung der Jumper zu den Ausgängen erfolgt mit aufsteigenden Nummern, d.h. ST41 ist der Jumper für Kanal 1, ST42 für Kanal 2 usw.

3.6.11 ST30: Watchdog Timeout

Die Zeit, nach der der Watchdog anspricht kann eingestellt werden


Zeit			Jumper
min.	typ.	max.	
62,5 ms	150 ms	250 ms	2-3
250 ms	600 ms	1000 ms	offen
500 ms	1200 ms	2000 ms	1-2

3.6.12 ST31: Watchdog automatisch starten

Funktion	Jumper
Watchdog startet automatisch beim Einschalten	1-2
Watchdog startet mit dem ersten Zugriff auf den Watchdog	2-3

3.7 Jumper des phyCORE-MPC555 Moduls

Die folgenden Jumper müssen bzw. sollten auf dem MPC555-Modul geändert werden:

- J2 **muß** von Pos. 1-2 auf 2-3 geändert werden (Boot aus int. FLASH) 
- Achtung: In der Phyttec-Dokumentation ist J5 verzeichnet, erst im Anhang wird auf die Vertauschung von J2 und J5 hingewiesen!
- J6 kann von Pos. 2-3 auf 1-2 geändert werden, um die Batterie zu entlasten (Batteriepufferung des int. RAM, cf. cap. 4.13)

	Versorgung	Stromverbrauch (aus Batterie)
RAM mit Batteriepufferung	On	5 μ A
	Off	10 μ A
RAM ohne Batteriepufferung (nur Echtzeituhr versorgt)	On	0,27 μ A
	Off	0,63 μ A

- J7 sollte geöffnet werden (Referenzspannung von extern)

-
- J11 muß geöffnet werden, wenn der CAN 1 Kanal genutzt werden soll

Weitere Informationen liefert das phyCORE-MPC555 Hardware-Manual. Wir empfehlen die Arbeiten nur von Fachpersonal mit entsprechendem Werkzeug durchführen zu lassen, die Jumper sind als 0402-Widerstände ausgeführt!



4 Hardwarebeschreibung

Das phy555-Trägerboard ist für die Aufnahme des phyCORE-MPC555 Moduls der Firma Phytec ausgelegt. Es stellt die Anpassung für die I/O-Signale des MCP555 zur Verfügung. Die Beschreibung finden Sie in den folgenden Kapiteln.

4.1 Verteilung der 24 Volt

Die eingespeisten 24 Volt werden auf verschiedene Stecker verteilt. Zum Teil sind eigene Sicherungen auf der Platine vorgesehen. Die 24 Volt werden vor dem EMV-Filter abgegriffen. Um die Spannungen auf einen Blick überwachen zu können, sind LED's vorhanden. Die folgende Tabelle gibt einen Überblick:

Signal	Sicherung	Stecker	Anschluß	LED
+24V_1	0,2 A selbstrückstellend	ST55	DI1 - DI16	LE1
+24V_2	0,2 A selbstrückstellend	ST64	DI17- DI32	LE2
+24V_3	0,2 A selbstrückstellend	ST34/35	D01 - D016	LE3
+24V_4	0,2 A selbstrückstellend	ST36/37	D017- D032	LE4
+24V_5	0,2 A selbstrückstellend	ST49	AI1 - AI16	LE5
+24V_6	0,2 A selbstrückstellend	ST50	AI17- AI32	LE6
+24V	keine	ST18	Tastatur	-

4.2 Ethernetschnittstelle

Das Board stellt eine 10/100BaseT-Schnittstelle (ST7) zur Verfügung. Über die beiden Leuchtdioden im Stecker kann der Zustand der Ethernet-Schnittstelle beobachtet werden. Die gelbe LED leuchtet bei einer 100 MBit Verbindung. Die grüne LED leuchtet bei einer bestehenden Verbindung, sie blinkt, wenn Daten übertragen werden.

4.3 Serielle Schnittstellen

Es stehen bis zu 4 serielle Schnittstellen zur Verfügung, die unterschiedlich bestückt sein können.

4.3.1 Schnittstelle A1

Die Schnittstelle A1 ist die Bedienschnittstelle des RTOS-UH. Über diese Schnittstelle werden z.B. die Einschaltmeldung und ggf. Fehlermeldungen ausgegeben. Diese Schnittstelle ist als 5-Draht RS232-Schnittstelle realisiert. Die Startbaudrate beträgt 57600 Baud.

4.3.1.1 Belegung A1

Die Schnittstelle A1 ist eine 5-Draht Schnittstelle (ST10). Die Belegung wurde so gewählt, daß ein angepreßter 9 poliger SubD-Stecker eine PC-kompatible Belegung hat:

ST10	PIN	PIN	ST10
	1	2	
RxD	3	4	RTS
TxD	5	6	CTS
	7	8	
GND	9	10	

4.3.2 Schnittstelle A2

Die Schnittstelle A2 ist als Modemschnittstelle ausgelegt. Sie ist als 5-Draht RS232-Schnittstelle realisiert. Die Startbaudrate beträgt 57600 Baud.

4.3.2.1 Belegung A2

Die Schnittstelle A2 ist eine 5-Draht Schnittstelle (ST11). Die Belegung wurde so gewählt, daß ein angepreßter 9 poliger SubD-Stecker eine PC-kompatible Belegung hat:

ST11	PIN	PIN	ST11
	1	2	
RxD	3	4	RTS
TxD	5	6	CTS
	7	8	
GND	9	10	

4.3.3 Schnittstelle A3

Die Schnittstelle A3 kann als RS422- (IC16) oder RS485-Schnittstelle (IC15) bestückt sein. Dementsprechend können die Abschlußwiderstände R47-R50 bestückt werden.

4.3.3.1 Belegung A3

Ist die Schnittstelle als RS485 bestückt, sind die Signale Rx am Anschluß ST15 nicht belegt.

ST15	PIN	PIN	ST15
	1	2	
	3	4	Rx -
Tx +	5	6	Tx -
Rx +	7	8	
GND	9	10	

4.3.4 Schnittstelle A4

Die Schnittstelle A4 kann als RS422- (IC18) oder RS485-Schnittstelle (IC17) bestückt sein. Dementsprechend können die Abschlußwiderstände R51-R54 bestückt werden.


4.3.4.1 Belegung A4

Ist die Schnittstelle als RS485 bestückt, sind die Signale Rx am Anschluß ST16 nicht vorhanden.

ST16	PIN	PIN	ST16
	1	2	
	3	4	Rx -
Tx +	5	6	Tx -
Rx +	7	8	
GND	9	10	

4.4 CAN-Bus

Es stehen die 2 CAN-Busschnittstellen des MPC555 zur Verfügung. Die Schnittstelle CAN 1 (ST8) ist galvanisch vom Rest des Boards getrennt. Mit ST14 kann der Abschlußwiderstand eingeschaltet werden.

Die Schnittstelle CAN 2 (ST9) ist direkt mit dem CAN-Tranceiver auf dem Prozessor-Board verbunden. Überspannungen können zur **Zerstörung** des Prozessors-Boards führen! Mit ST17 kann der Abschlußwiderstand eingeschaltet werden. 

4.4.1 Belegung CAN-Bus

Beide Schnittstellen sind identisch belegt. Wird eine 9 polige SubD-Buchse angepreßt, so ergibt sich eine CANopen konforme Belegung:

ST8/ST9	PIN	PIN	ST8/ST9
	1	2	GND
CAN Low	3	4	CAN High
GND	5	6	
	7	8	
PE	9	10	

4.5 Tastatur

Es werden Matrix-Tastaturen mit bis zu 7x8 Tasten unterstützt.

4.5.1 Belegung Matrixtastatur-Anschluß

Auf diesem Anschluß (ST19) sind neben der Tastatur noch die Ansteuerung für 2 LED's sowie LCD-Hintergrundbeleuchtung und einen „Beep“ untergebracht:

ST19	PIN	PIN	ST19
Tastatur Zeile 1	1	2	Tastatur Zeile 2
Tastatur Zeile 3	3	4	Tastatur Spalte 1
Tastatur Spalte 2	5	6	Tastatur Spalte 3
LED rot	7	8	LED grün
+5 Volt	9	10	GND
GND	11	12	GND
+24 Volt	13	14	GND
Beep	15	16	LCD-Beleuchtung

Die +24 Volt kommen direkt von der 24 Volt Versorgung, d.h. es ist keine EMV-Schutzbeschaltung oder Sicherung vorhanden.



4.6 LCD

Der LCD-Anschluß ist für die Ansteuerung eines LCD's mit dem Toshiba 6963 ausgelegt, z.B. für ein TLX-1741-C3M.

4.6.1 Belegung LCD-Anschluß

ST27	PIN	PIN	ST27
	1	2	GND1
+5 Volt	3	4	VEE
WR	5	6	RD
CS	7	8	A0
	9	10	/Reset
D0	11	12	D1
D2	13	14	D3
D4	15	16	D5
D6	17	18	D7
Font	19	20	Inv

4.7 CompactFlash Memory Card

ST33 ist der Anschluß für CompactFlash Memory Cards vom Typ I oder Typ II. Die Belegung entspricht der Norm und wird hier daher nicht wiedergegeben. Sie ist z.B. bei <http://www.sandisk.com> zu finden. Die CF-Card kann im PC formatiert werden, es darf aber kein FAT32-Dateisystem verwendet werden. Unter dem Device /H0 ist die CF-Card im RTOS-UH zu finden. Die Leuchtdiode LE13 zeigt Zugriffe auf die CF-Card an, wird die Karte entfernt, während die LED leuchtet, kann dies zu Datenverlusten führen!



4.8 Digitale Ausgänge

Es stehen 32 digitale Ausgänge in 4 Gruppen á 8 Ausgängen zur Verfügung. Die High-Side-Schalter stellen 24 Volt / 400 mA zur Verfügung. Alle 32 Ausgänge können max. 6 A

Summenstrom liefern, die selbstrückstellende Sicherung SI17 übernimmt die Absicherung. Die Ausgänge sind zum Schalten induktiver Lasten geeignet und dauerkurzschlußfest. Die Treiberbausteine sind gegen Übertemperatur gesichert. Im Fehlerfall wird der betroffene Ausgang zurück gesetzt, d.h. der Ausgang ist gesperrt. Dieser Zustand wird über eine rote Error-LED je Gruppe angezeigt. Der Fehlerzustand wird bei Beendigung des Fehlers automatisch zurück gesetzt. Es werden weitere Fehlerzustände erkannt, z.B. keine Last angeschlossen usw. Nähere Informationen entnehmen Sie bitte dem Datenblatt des BSP742.

Die 4 Stecker sind gleich belegt:

ST34	ST35	ST36	ST37	PIN	PIN	ST37	ST36	ST35	ST34
DO1	DO9	DO17	DO25	1	2	DO26	DO18	DO10	DO2
DO3	DO11	DO19	DO27	3	4	DO28	DO20	DO12	DO4
DO5	DO13	DO21	DO29	5	6	DO30	DO22	DO14	DO6
DO7	DO15	DO23	DO31	7	8	DO32	DO24	DO16	DO8
+24V_3	+24V_3	+24V_4	+24V_4	9	10	GND	GND	GND	GND

Die Ausgänge D01-D014 sind an die A_TPU angeschlossen und können Sonderfunktionen ausführen.

4.9 Digitale Eingänge

Die digitalen Eingänge DI1-DI12 und DI17-DI28 sind einseitig auf Masse bezogen. Die Eingänge DI13-DI16 und DI29-DI32 können sowohl einseitig auf Masse bezogen werden, als auch als potentialgetrennte Eingänge genutzt werden. Beide Steckverbinder sind identisch belegt:

ST55	ST64	PIN	PIN	ST64	ST55
DI1	DI17	1	2	GND	GND
DI2	DI18	3	4	GND	GND
DI3	DI19	5	6	GND	GND
DI4	DI20	7	8	GND	GND
DI5	DI21	9	10	GND	GND
DI6	DI22	11	12	GND	GND
DI7	DI23	13	14	GND	GND
DI8	DI24	15	16	GND	GND
DI9	DI25	17	18	GND	GND
DI10	DI26	19	20	GND	GND
DI11	DI27	21	22	GND	GND
DI12	DI28	23	24	GND	GND
+DI13	+DI29	25	26	-DI29	-DI13
+DI14	+DI30	27	28	-DI30	-DI14
+DI15	+DI31	29	30	-DI31	-DI15
+DI16	+DI32	31	32	-DI32	-DI16
+24V_1	+24V_2	33	34	+24V_2	+24V_1

Werden die Eingänge DI13-DI16 und DI29-DI32 potentialfrei genutzt, kehrt sich die Polarität der Eingänge um!



4.10 Analoge Ausgänge

Die analogen Ausgänge sind für 0..20mA ausgelegt. Ausgang Nr. 8 kann auch negative Spannungen ausgeben. Je nach Bestückung sind auch andere Bereiche möglich. Der Rückkanal der analogen Ausgänge ist jeweils über eine selbststrückstellende 20 mA Sicherung geführt, die bei zu großen Unterschieden im Ground-Potential auslöst. Spannungsausgänge von 0-10 Volt stehen durch Schließen einer Bürde von 536 Ω zur Verfügung.



ST38	PIN	PIN	ST38	Sicherung
AO1	1	2	GND	SI9
AO2	3	4	GND	SI10
AO3	5	6	GND	SI11
AO4	7	8	GND	SI12
AO5	9	10	GND	SI13
AO6	11	12	GND	SI14
AO7	13	14	GND	SI15
AO8	15	16	GND	SI16

4.11 Analoge Eingänge

Die analogen Eingänge sind als differentielle Eingänge ausgeführt. Der Eingangsspannungsbereich kann über eine unterschiedliche Bestückung in weiten Grenzen variiert werden.

ST50	ST49	PIN	PIN	ST49	ST50
AI17+	AI1+	1	2	AI1-	AI17-
AI18+	AI2+	3	4	AI2-	AI18-
AI19+	AI3+	5	6	AI3-	AI19-
AI20+	AI4+	7	8	AI4-	AI20-
AI21+	AI5+	9	10	AI5-	AI21-
AI22+	AI6+	11	12	AI6-	AI22-
AI23+	AI7+	13	14	AI7-	AI23-
AI24+	AI8+	15	16	AI8-	AI24-
AI25+	AI9+	17	18	AI9-	AI25-
AI26+	AI10+	19	20	AI10-	AI26-
AI27+	AI11+	21	22	AI11-	AI27-
AI28+	AI12+	23	24	AI12-	AI28-
AI29+	AI13+	25	26	AI13-	AI29-
AI30+	AI14+	27	28	AI14-	AI30-
AI31+	AI15+	29	30	AI15-	AI31-
AI32+	AI16+	31	32	AI16-	AI32-
+24V_6	+24V_5	33	34	+24V_5	+24V_6

4.12 Leuchtdioden

Zur Kontrolle der unterschiedlichen Spannungen sind auf der Platine 11 grüne Leuchtdioden vorhanden:

Spannung	LED
+24V_1	LE1
+24V_2	LE2
+24V_3	LE3
+24V_4	LE4
+24V_5	LE5
+24V_6	LE6
+5 Volt	LE7
+3,3 Volt	LE8
+15 Volt	LE9
-15 Volt	LE10
+5 V Referenz	LE11

4.13 Batterie/Goldcap

Das Gerät ist ggf. mit einem gepufferten RAM ausgestattet. Die Pufferung erfolgt entweder über eine 3,0 Volt Lithiumbatterie oder einen 1 F Goldcap. Die Batterie wird erst bei der Lieferung eingesetzt und hat dann eine Mindestlebensdauer von 5 Jahren, unabhängig von der Einschaltdauer des Gerätes. Die Pufferdauer mit dem Goldcap hängt vom Ladezustand ab, bei geladenem Goldcap ist eine Pufferdauer von min. 14 Tagen gegeben.

4.14 Externer Watchdog

Es steht zusätzlich zum internen Watchdog des MPC555 noch ein externer Watchdog zur Verfügung. Dieser Watchdog wird erst mit der ersten Ansprache aktiviert und muß dann regelmäßig getriggert werden (s. cap. 5.3). Spricht der Watchdog an, so unterbricht ein Relais die 24 Volt Versorgungsspannung des Boards.

5 Programmierung

Die Programmierung der auf dem phyCORE-MPC555 befindlichen Bausteine wird vom Betriebssystem durchgeführt. Die Programmierung aller externen Komponenten wird im Folgenden beschrieben. Absolut notwendig zum Verständnis ist die detaillierte Kenntnis des MPC555 User's Manual.

Es ist unter <http://e-www.motorola.com/brdata/PDFDB/docs/MPC555UM.pdf> erhältlich.

Für die Programmierung der TPU sind die 20 Applikation-Notes hilfreich:

<http://e-www.motorola.com/brdata/PDFDB/docs/TPUPN01.pdf> bis

<http://e-www.motorola.com/brdata/PDFDB/docs/TPUPN20.pdf>.

5.1 Adreßbelegung

Chip-Select	Anschluß	Größe	Adresse	Programmierung
CS0	ext. FLASH	bis 4 MB	\$00C00000-\$00CFFFFFFF	0 WS
CS1	Burst-RAM	bis 4 MB	\$00000000-\$00xFFFFFFF	0 WS
CS2	Ethernet	16 MB	\$82000000-\$82FFFFFFF	7 WS
CS2	Grafik (ST18)	16 MB	\$83000000-\$83FFFFFFF	7 WS
CS3	CompactFlash	4 KB	\$80810000-\$80810FFF	4 WS
CS3	LC-Display	4 KB	\$80811000-\$80811FFF	4 WS
CS3	Tastatur	4 KB	\$80812000-\$80812FFF	4 WS
CS3	Watchdog	4 KB	\$80813000-\$80813FFF	4 WS
Interner MPC555 Bereich		4 MB	\$01C00000-\$01FFFFFFF	

Je nach Bestückung des phyCORE-MPC555 Moduls erstreckt sich der CS1-Bereich bis:

RAM-Größe	CS1-Endadresse
1 MB	\$000FFFFFFF
2 MB	\$001FFFFFFF
4 MB	\$003FFFFFFF
8 MB	\$007FFFFFFF

5.2 Interruptquellen

Quelle	Funktion	Level
IRQ1	Ethernet	1
TPU_B	TPU_B	2
IRQ3	Tastatur	3
QSMCM	Serielle	4
TOUCAN	CAN-Bus	5
TPU_A	TPU_A	6

Die restlichen IRQ-Pin's werden als digitale Eingänge genutzt.

5.2.1 Events

Folgende Events sind belegt:

Event	Funktion	Event	Funktion
EV 00000001	TPU A Kanal 0	EV 00010000	TPU B Kanal 0
EV 00000002	TPU A Kanal 1	EV 00020000	TPU B Kanal 1
EV 00000004	TPU A Kanal 2	EV 00040000	TPU B Kanal 2
EV 00000008	TPU A Kanal 3	EV 00080000	TPU B Kanal 3
EV 00000010	TPU A Kanal 4	EV 00100000	TPU B Kanal 4
EV 00000020	TPU A Kanal 5	EV 00200000	TPU B Kanal 5
EV 00000040	TPU A Kanal 6	EV 00400000	TPU B Kanal 6
EV 00000080	TPU A Kanal 7	EV 00800000	TPU B Kanal 7
EV 00000100	TPU A Kanal 8	EV 01000000	TPU B Kanal 8
EV 00000200	TPU A Kanal 9	EV 02000000	TPU B Kanal 9
EV 00000400	TPU A Kanal 10	EV 04000000	frei
EV 00000800	TPU A Kanal 11	EV 08000000	frei
EV 00001000	TPU A Kanal 12	EV 10000000	frei
EV 00002000	TPU A Kanal 13	EV 20000000	frei
EV 00004000	TPU A Kanal 14	EV 40000000	A/D-Wandler B
EV 00008000	TPU A Kanal 15	EV 80000000	A/D-Wandler A

Damit der entsprechende Event ausgelöst wird, muß der Interrupt physikalisch freigegeben werden.

5.3 Watchdog

Der interne Watchdog des MPC555 wird vom Betriebssystem initialisiert und von der Task Watch auf sehr hoher Priorität (-9) im 1s-Zyklus getriggert. Unterbleibt die Triggerung des Watchdogs für mehr als 2 s, wird ein Reset des IF555 ausgelöst.

Die Task Watch kann problemlos terminiert und mit einer anderen Wunschkategorie wieder aktiviert werden. Auch eine Neutriggierung des Watchdogs direkt über eine Anwendung ist möglich.

Der externe Watchdog des IF555 muß anwendungsspezifisch konfiguriert (s. Jumper, cap. 3.6.11) und von der Anwendung regelmäßig getriggert werden. Zum Starten und Triggern des Watchdogs ist ein beliebiger Zugriff in den Adressbereich des Watchdogs (s. Adressbelegung, CS2, \$80812000-\$80812FFF) erforderlich.

5.4 Digitale Ausgänge

Die digitalen Ausgänge D01 bis D014 sind an die Anschlüsse A_TPU0 bis A_TPU13 angeschlossen. Die Ausgänge D015 bis D032 sind über Schieberegister an die SPI-Schnittstelle angeschlossen. Chip-Select ist der PCS0.

Die Fehlermeldungen der 4 Ausgangsgruppen sind an folgende Ports angeschlossen:

Port	Ausgangsgruppe
SGPIOC6	D01 -D08
IRQ2	D09 -D016
IRQ4	D017-D024
MDA3	D025-D032

Ein Low-Pegel zeigt einen Fehler an.

5.4.1 Abbildung QSPI-Transmitregister <-> Ausgänge

Es werden die ersten 3 Worte im QSPI-Transmitregister belegt. Die folgende Übersicht beschreibt die Abbildung des Transmitregisters auf die Ausgänge:

0x1F05180 **TR0** 15 ... 8 7 6 5 4 3 2 1 0

ST26	-	-	-	8	7	-	-	-	-	-	-
-------------	---	---	---	---	---	---	---	---	---	---	---

0x1F05182 **TR1** 15 ... 8 7 6 5 4 3 2 1 0

ST25	-	-	-	8	7	6	5	4	3	2	1
-------------	---	---	---	---	---	---	---	---	---	---	---

0x1F05184 **TR2** 15 ... 8 7 6 5 4 3 2 1 0

ST24	-	-	-	8	7	6	5	4	3	2	1
-------------	---	---	---	---	---	---	---	---	---	---	---

5.5 Digitale Eingänge

Die digitalen Eingänge sind folgendermassen angeschlossen:

Eingang	Port
DI1 -DI8	MPIO0 -MPIO7
DI9 -DI16	B_TPU0-B_TPU7
DI17-DI24	MPIO9 -MPIO15
DI25-DI30	QSPI
DI31-DI32	A_TPU14-A_TPU15

Fließt Strom durch den Eingang (Schalter geschlossen), liegt am Eingang High-Pegel.

Die Eingänge DI25-DI30 bilden sich folgendermaßen ab:

0x1F05140 **RR0** 15 ... 8 7 6 5 4 3 2 1 0

ST44	-	-	-	-	-	6	5	4	3	2	1
-------------	---	---	---	---	---	---	---	---	---	---	---

5.6 Analoge Ausgänge

Die analogen Ausgänge sind an die Anschlüsse MPWM0 bis MPWM7 angeschlossen.

5.7 Analoge Eingänge

Die analogen Eingänge AI1 bis AI16 sind an die Eingänge A_AD0 bis A_AD15, AI17 bis AI32 an B_AD0 bis B_AD15 angeschlossen. Die Zuordnung zu den MPC555 Kanälen ist folgende:

Eingang	MPC555
AI1	A_AN0
AI2	A_AN1
...	...
AI5	A_AN48
...	...
AI16	A_AN59
AI17	B_AN0
AI18	B_AN1
...	...
AI21	B_AN48
...	...
AI32	B_AN59

5.8 Compact Flash Card

Die CF-Card belegt 4 Ports:

Signal	Port	Richtung	Funktion
CD1/CD2	MDA2	in	Low = Karte steckt
RDY	MDA1	in	High = Karte ist bereit
Reset	MDA0	out	Low = Versorgung liegt an High = spannungslos

5.8.1 Formatieren der CF-Card

Die CF-Card lässt sich mit folgendem Kommando formatieren:

```
FORM D /H0/X9H8xxxx
```

Der Wert von *xxxx* hängt von der Größe der CF-Card ab. Soll die Karte mit der maximal möglichen Größe von 1GB formatiert werden, so ist ein Wert von 4094 einzusetzen. Kleine Größen errechnen sich entsprechend.

Bitte beachten Sie, dass das Formatieren bis zu 20 Minuten dauern kann und die Schnittstelle solange blockiert ist.



5.9 Serielle Schnittstellen

Für die seriellen Schnittstellen finden folgende Port-Anschlüsse Verwendung:

Signal	Port
TxD_1	TXD1
RxD_1	RXD1
RTS_1	MDA6
CTS_1	B_TPU10
TxD_2	TXD2
RxD_2	RXD2
RTS_2	MDA7
CTS_2	B_TPU11
TxD_3	B_TPU12
RxD_3	B_TPU13
RTS_3	MDA8
TxD_4	B_TPU14
RxD_4	B_TPU15
RTS_4	MDA9

5.10 Port-Pins der Tastatur

Neben den Anschlüssen für die Tastatur, die direkt auf den Tastatur-Controller gehen, stehen noch weitere Port-Pins auf dem Tastatur-Anschluß zur Verfügung:

Signal	Port
LED rot	MDA5
LED grün	MDA4
Beep	B_TPU9
LCD Light	QGPI03

Die beiden LED's sind TTL-Ausgänge, die nach Low schalten. Max. dürfen 20 mA geschaltet werden, die maximal zulässige Spannung beträgt 5V.

Die anderen beiden Ausgänge sind über FET's geführt, die nach Masse schalten. Der maximale On-Widerstand beträgt 0,5Ω. Der Strom sollte 0,5 A nicht übersteigen. Die Sperrspannung beträgt 50 Volt.

6 RTOS-UH im internen FLASH ablegen

Um das Betriebssystem zum ersten Mal im internen FLASH des MPC555 abzulegen ist ein Background-Debugger nötig. Auf dem EVA-Board der Firma Phytec ist eine zum Macraigor Debugger kompatible Schnittstelle (Wiggler) integriert.

Das Board ist entsprechend der Phytec Beschreibung zu jumpern, über den Parallel-Port des PC's sollte dann Kontakt zum MPC555 aufgenommen werden können. Um das RTOS-UH auf das Board zu laden, muß zuerst das Macro `iep555.mac` gestartet werden. Damit werden z.B. die ChipSelects für das RAM initialisiert. Dann kann der SRecord des Betriebssystems geladen werden. Hierbei ist zu beachten, dass der S0-Record entfernt wurde. Mit dem Macro `start55.mac` kann RTOS-UH gestartet werden. Über die erste serielle Schnittstelle muß die Überschriftmeldung ausgegeben werden. Mit dem folgenden Sript wird das laufende Betriebssystem ins interne FLASH des MPC555 gebrannt:

```
WAIT ; SM-L 80828 0 0 1FFFFFF0
WAIT ; FCLEAR FFFC
WAIT ; FDUMP 80000:8000 1C00000
WAIT ; FDUMP 88000:8000 1C08000
WAIT ; FDUMP 90000:8000 1C10000
WAIT ; FDUMP 98000:8000 1C18000
WAIT ; FDUMP A0000:8000 1C20000
WAIT ; FDUMP A8000:8000 1C28000
WAIT ; FDUMP B0000:8000 1C30000
WAIT ; FDUMP B8000:8000 1C38000
WAIT ; FDUMP C0000:8000 1C40000
WAIT ; FDUMP C8000:8000 1C48000
WAIT ; FDUMP D0000:8000 1C50000
WAIT ; FDUMP D8000:8000 1C58000
WAIT ; FDUMP E0000:8000 1C60000
WAIT ; FDUMP E8000:8000 1C68000
WAIT ; SM-L 80000 0 0 0
WAIT ; FDUMP 80000:4 1C00000 S
```

Ist schon ein RTOS-UH auf dem MPC555 am laufen, kann der neue RTOS-UH S-Record auf die Adresse 80000 geladen und dann mit `GO 80818` angesprungen werden. Danach kann das System mit den obigen Befehlen upgedated werden.

Bitte beachten Sie: Um das interne Flash löschen und brennen zu können, muß die Lötbrücke ST3 geschlossen sein.



7 Terminalemulation

7.1 Allgemeines

Die Terminalemulation ermöglicht den Anschluß von getrennten Tastaturen und Displays an einem RTOS-UH-Rechner und verhält sich wie eine übliche serielle Schnittstelle.

Unter den mnemotechnischen Bezeichnungen /AT, /BT und /CT ist die Terminalemulation in der A-, B- oder C-Betriebsart unter der LDN 4 mit den Drives 0, 2 und 6 erreichbar. Unter der Bezeichnung /DT wird der Ausgabekanal im Vollduplex-Betrieb mit der LDN 14 angesprochen.

Zusätzlich zu den üblichen Betriebsarten kann die Terminalemulation bei LDN 4 noch unter den Drives 64, 66 und 70 angesprochen werden. Die Terminalemulation liefert in diesem Fall bei Eingaben die Tastatur-Scancodes ohne Übersetzung in ASCII-Zeichen. Auch hier ist ein Betrieb entsprechend der A-, B- oder C-Betriebsart möglich, jedoch findet grundsätzlich kein Echo statt. Besonders ist zu beachten:

- Bei Betriebsartenumschaltung Scan-Code/ASCII kann der Empfangspuffer noch Zeichen in der jeweils anderen Codierung enthalten. Zur Sicherheit sollte der Empfangspuffer daher durch ein erstes Lesen in der A-Betriebsart gelöscht werden.
- Bei Scan-Code-Betrieb müssen die Ausgaben ebenfalls über die Drives 64, 66 oder 70 erfolgen, da ansonsten wieder eine Rückschaltung in den ASCII-Betrieb erfolgt.

Der Tastaturreiber bietet keine Auto-Repeat-Funktion. Statt dessen sind Treiberversionen verfügbar, die bei Loslassen einer gedrückten Taste den ASCII-Wert \$FF liefern. Hierdurch kann die Dauer einer Tastenbetätigung erfasst und ggf. ein Auto-Repeat emuliert werden.

7.2 Befehlsvorrat Terminalemulation

Die Terminalemulation verhält sich weitgehend Televideo-kompatibel. Sie versteht die folgende Befehlssequenzen (Erläuterung der Angaben *Ps*, *Pc*, *Pn*, *r* und *c* siehe unten). Es ist nicht gewährleistet, daß alle Funktionen auf allen Terminals zur Verfügung stehen! So können auf einen schwarz/weiß-Display natürlich keine Farben eingestellt werden, oder auf einen Textdisplay wird es schwierig, Grafik auszugeben.

Beachten Sie bitte, dass die folgenden Sequenzen nur nutzbar sind, wenn der Grafik-Mode nicht benutzt wird!



Cursor style (0..4)		
	ESC . <i>Ps</i>	
	0	Cursor off
	1	Blinking block
	2	Steady block
	3	Blinking underline
	4	Steady underline

Define Visual attributes																				
	ESC	G	P	0	1	2	3	4	5	6	7	8	9	:	;	<	=	>	?	
												X	X	X	X	X	X	X	X	
			Underline									X	X	X	X	X	X	X	X	
			Reverse					X	X	X	X					X	X	X	X	
			Blinking			X	X			X	X			X	X				X	X
			Invisible		X		X		X		X		X		X		X		X	
Operating Modes																				
)	Enable invers																
			(Disable invers																
			V	Autoscroll off																
			W	Autoscroll on																
			\$	Grafics mode on																
			%	Grafics mode off																
			U	Monitor mode on																
			X	Monitor mode off																
			[=7h	Auto Wrap on																
			[=7l	Auto Wrap off																
			! c	c >= ,A': Schreibfarbe, c<'A': Hintergrundfarbe setzen																
			[Pc ; Pc r	Set scroll region rows (Pc: start, end)																
			[Pc ; Pc s	Set scroll region columns (Pc: start, end)																
Cursor Movement																				
			J	Line feed																
			K	Up																
			V	Down																
			L	Right																
			M	Carriage Return																
			H	Left																
			= r c	Set Cursor row column																
			[Pn A	Cursor up Pn rows																
			[Pn B	Cursor down Pn rows																
			[Pn C	Cursor right Pn columns																
			[Pn D	Cursor left Pn columns																
			[c ; r H	Set Cursor position column, row																
Tabulator Control																				
			1	Set tab stop at actual column																
			2	Clear tab stop at actual column																
			3	Clear all tab stops																
			i	Move Cursor to next tab stop																
			l	Move Cursor back one tab stop																
Insertion																				
			Q	Insert space at cursor																
			E	Insert line of spaces																
Miscellaneous																				
			~	Reset Terminal																
Deletion																				
			W	Delete char at cursor																
			R	Delete current line to spaces																
			T	Delete to end of line																

	t	
	Y	Delete to end of screen
	y	
	*	Delete Screen
	,	
	;	
	+	
	:	
CTRL Z		

Die Befehlssequenz zur Änderung der Schreibfarbe (ESC ! c) ist nicht zu gängigen Terminal emulationen kompatibel. Neben der einfachen Änderung der Schreibfarbe kann durch Änderung des Grundwertes der Farbangabe folgendes Verhalten erzeugt werden:

Grundwert	Verhalten
32 (Leerzeichen)	Änderung der Schreibfarbe
64 (A)	Änderung der Hintergrundfarbe

Die Angaben für mit *Ps*, *Pc*, *Pn*, *r* oder *c* gekennzeichnete Zahlenwerte erfolgen durch Werte, die sich ASCII-codiert aus 32 + Zahlenwert errechnen.

7.3 Umsetzung der Eingabezeichen

Bei der Matrixtastatur sind unterschiedliche Belegungen möglich. So können z.B. sämtliche Tasten als frei programmierbare Funktionstasten zur Verfügung gestellt werden. Bei der PC-Tastatur sind die normalen Funktionstasten belegt, die Tasten von 13 bis 24 werden durch gleichzeitiges Drücken von *Shift* und *Funktionstaste* erreicht. Standardmäßig liefern die Funktionstasten *S0H zeichen CR* mit folgendem *zeichen*

Funktionstaste	Zeichen	Shift+F-Taste	Zeichen
1	@	13	'
2	A	14	a
3	B	15	b
4	C	16	c
5	D	17	d
6	E	18	e
7	F	19	f
8	G	20	g
9	H	21	h
10	I	22	i
11	J	23	j
12	K	24	k

Bei Matrixtastaturen hängt die Zuordnung der einzelnen Tasten zu den Tastencodes von dem Anschluß der Tastatur ab und muß für jeden Anwendungsfall ermittelt werden.

Die Programmierung der Funktionstasten erfolgt mit der Befehlssequenz

ESC | *p11 text* CTRL-Y

und den Parametern

p1 Kennzeichnung der zu programmierenden Taste

text gewünschte Tastenbelegung

Funktionstaste	Kennung	Shift+F-Taste	Kennung
1	1	13	<
2	2	14	=
3	3	15	>
4	4	16	?
5	5	17	@
6	6	18	A
7	7	19	B
8	8	20	C
9	9	21	D
10	:	22	E
11	;	23	F
12	G	24	L

7.4 Grafikfunktionen

Bei grafikfähigen Displays stellt die Terminalemulation Funktionen zur Nutzung der grafischen Fähigkeiten des Displays zur Verfügung. Der Leistungsumfang richtet sich nach der Leistungsfähigkeit des jeweiligen Grafik-Displays bzw. des verwendeten Controllers.

Die Grafikfunktionen sind aus Geschwindigkeitsgründen nicht auf Multi-Tasking ausgelegt. Grundsätzlich sollte ein Grafiksystem nur von einer Task angesprochen werden. Besonders der kombinierte Betrieb von Terminalemulation und Grafikfunktionen erfordert besondere Aufmerksamkeit.

Bitte beachten Sie, dass bei der Nutzung des Grafik-Mode bei bestimmten Displays der Attributspeicher nicht mehr zur Verfügung steht. Dies hat z.B. zur Folge, dass die Ausgabe von invertiertem Text nicht mehr möglich ist.



Neben den Grafik-Primitiven SETPIX, GETPIX und LINE (s. RTOS-UH-Handbuch Kapitel 5.7.3) gibt es folgende Prozeduren:

7.4.1 Funktionsumfang und Implementierungsabhängigkeiten

Prozedur	IP-SVGA	PVGA	VHI	NBS300	MPC555
BACK PEN	Ja	Ja	—	—	—
BIN TEXT	Ja	Ja	Ja	Ja	Ja
BOX	Ja	Ja	Ja	Ja	Ja
BOX FILLED	Ja	Ja	Ja	Ja	Ja
BOX MOVE	Ja	Ja	Ja	Ja	Ja
BOX READ	Ja	Ja	—	—	—
BOX WRITE	Ja	Ja	—	—	—
CIRCLE	Ja	Ja	Ja	Ja	Ja
CLEAR	Ja	Ja	Ja	Ja	Ja
COLOR CLEAR	Ja	Ja	—	—	—
DISPLAY MODE	Ja	Ja	Ja	Ja	Ja
DISPLAY STATE	Ja	Ja	—	—	—
DOT LINE	Ja	Ja	Ja	Ja	Ja
DOT LINE RANGE	Ja	Ja	Ja	Ja	Ja
FILLED HISTOGRAM	Ja	Ja	Ja	Ja	Ja
GET PEN	Ja	Ja	—	—	—
GET BACK PEN	Ja	Ja	—	—	—
GET TEXT FONT	Ja	Ja	—	—	—
GET TEXT WIDTH	Ja	Ja	Ja	Ja	Ja
GRAPH CURSOR	Ja	Ja	—	—	—
GRAPH CURSOR SELECT	Ja	Ja	—	—	—
HIDE CURSOR	Ja	Ja	—	—	—
HISTOGRAM	Ja	Ja	Ja	Ja	Ja
PEN	Ja	Ja	—	—	—
PLANE	—	Ja	—	—	—
PLINIT	Ja	Ja	Ja	Ja	Ja
POLYGON	Ja	Ja	Ja	Ja	Ja
SELECT FONT	Ja	Ja	—	—	—
SELECT TEXT FONT	Ja	Ja	—	—	—
SET TEXT WIDTH	Ja	Ja	Ja	Ja	Ja
SET VGA SCREEN	Ja	Ja	—	—	—
TEXT	Ja	Ja	Ja	Ja	Ja
VGA GET PLANE	Ja	Ja	—	—	—
VGA LOCK TEXT	Ja	Ja	—	—	—
VGA SET PLANE	Ja	Ja	—	—	—
VGA UNLOCK TEXT	Ja	Ja	—	—	—
WIDTH	Ja	Ja	Ja	Ja	Ja

7.4.2 Allgemeines

Zum Verständnis der Funktionsbeschreibung sind Kenntnisse über die folgenden Begriffe erforderlich:

➤ Farbe

Farben werden in RGB-Notation angegeben. Die Angabe kennzeichnet die Intensität, in der der jeweilige Farbanteil dargestellt wird.

Der bei den verschiedenen Grafikfunktionen erforderliche `col`-Parameter gibt jedoch üblicherweise nicht direkt die Zeichenfarbe als RGB-Wert an. Vielmehr enthält diese Größe folgende Daten:

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Zeichenmode				Farbpalette											

Farbpalette

gibt die Nummer einer Farbpalette an, die den tatsächlichen RGB-Farbwert enthält. Die Anzahl der verfügbaren Paletten und damit der größtmögliche Farbpaletten-Wert hängen vom verwendeten Grafiksysteem ab.

Zeichenmode

Bei unterschiedlichen Grafiksysteemen stehen unterschiedliche Zeichenmodi zur Verfügung. Folgende Zeichenmodi sind verfügbar:

Kennung	Mode	Verfahren
0	absolut	Es wird mit der angegebenen Farbe gezeichnet
2	Invers	Es wird mit invertiertem Farbwert gezeichnet.
4	AND	Die Zeichenfarbe ergibt sich aus der bitweisen UND-Verknüpfung der angegebenen Farbe und der aktuellen Farbe des darzustellenden Bildpunkts.
8	OR	Die Zeichenfarbe ergibt sich aus der bitweisen ODER-Verknüpfung der angegebenen Farbe und der aktuellen Farbe des darzustellenden Bildpunkts.
12	XOR	Die Zeichenfarbe ergibt sich aus der bitweisen EXKLUSIV-ODER-Verknüpfung der angegebenen Farbe und der aktuellen Farbe des darzustellenden Bildpunkts.

Die Verknüpfung unterschiedlicher Zeichenmodi ist möglich.

➤ Pixel

Kennzeichnet einen Bildpunkt. Je nach verwendetem Grafiksysteem kann ein Bildpunkt unterschiedliche Farben annehmen.

➤ Pen

Ein Pen ist durch eine Farbe gekennzeichnet.

Es gibt einen Schreib- und einen Hintergrund-Pen. Pixel, Linien u.ä. werden nur unter Verwendung des Schreib-Pens gezeichnet.

Vorgegebene Konstrukte wie Fonts o.ä. füllen den Zeichenhintergrund mit dem Hintergrund-Pen und stellen die Zeichen mit dem Schreib-Pen dar.

➤ Font

Ein Font (Zeichensatz) ist als Bitmap abgelegt. Alle Zeichen eines Fonts haben gleiche Größe. Die Größe eines Fonts beinhaltet Ober- und Unterlängen.

Die Bitmap enthält Informationen darüber, welche Pixel der von einem Zeichen eingenommenen Fläche mit dem Schreib-Pen und welche Pixel mit dem Hintergrund-Pen darzustellen sind. Ein Font selber enthält keine Farbinformationen.

➤ Palette oder Plane

Die im Bildschirmspeicher abgelegten Pixel-Farbinformationen werden über Paletten in tatsächliche Farbtintensitätswerte übersetzt.

Die Pixel-Farbinformation kann in diesem Fall mit geringem Speicherbedarf abgelegt werden (z.B. 4 Bit). Diese Information wird als Index beim Zugriff auf die Palette verwendet. Die Palette liefert dann die tatsächliche Farbinformation als RGB-Wert mit deutlich höherer Farbaufösung (z.B. 3x 8 Bit).

➤ Screen

Ein Screen ist die Menge aller für einen sichtbaren Bildschirm im Bildspeicher abgelegten Daten. Bei Grafiksystemen, die entweder mehrere unterschiedliche Bildschirme unterstützen oder im Bildspeicher gleichzeitig Daten für mehrere Bildschirme ablegen können, ist die Wahl des Screens, auf dem die folgenden Zeichenoperationen erfolgen sollen, möglich.

7.4.3 Funktionsreferenz

PEARL-Spezifikation	Funktion
BACK_PEN: PROC(col FIXED(15)) GLOBAL;	Setzt die angegebene Farbe col als Hintergrundfarbe. Bei folgenden Operationen, die eine Hintergrundfarbe darstellen müssen, aber keine eigene Hintergrundfarbe spezifizieren (TEXT und BIN_TEXT), wird col als Hintergrundfarbe verwendet.
BIN_TEXT: PROC((x, y) FIXED(15), col FIXED(15), zeichen_adr FIXED(31)) GLOBAL;	Überträgt ein Zeichen aus dem Zeichengenerator-ROM in der Farbe col auf den Bildschirm an die Position (x, y). Die Adresse im ROM wird mit zeichen_adr angegeben. Sie hängt von der Größe des auszugebenden Zeichens ab: $zeichen_adr = \frac{xhöhe * ybreite + 7}{8}$
BOX: PROC((x, y) FIXED(15), (Xend, Yend) FIXED(15), col FIXED(15)) GLOBAL;	Zeichnet einen rechteckigen Rahmen in der Farbe col mit den diagonalen Punkten (x, y) (linke obere Ecke) und (Xend, Yend) (rechte, untere Ecke).
BOX_FILLED: PROC(Zeichnet ein in der Farbe col gefülltes Rechteck mit

<pre>(x, y) FIXED(15), (Xend, Yend) FIXED(15), col FIXED(15)) GLOBAL;</pre>	<p>den diagonalen Punkten (x, y) und (Xend, Yend)</p>																		
<pre>BOX_MOVE: PROC((Xstart, Ystart) FIXED(15), (Breite, Hoehe) FIXED(15), (Xziel, Yziel) FIXED(15), mode FIXED(15)) GLOBAL;</pre>	<p>Kopiert einen rechteckigen Bildausschnitt von Breite und Höhe beginnend bei dem Startpunkt (Xstart, Ystart) auf einen Bildbereich gleicher Größe mit dem Startpunkt (Xziel, Yziel). Der Zielbereich kann außerhalb des sichtbaren Bildschirmbereichs liegen, d.h. Ywidth wird überschritten. mode legt die Zeichenart fest, im unteren Nibble sind folgende Werte zulässig:</p> <table border="1" data-bbox="718 761 957 963"> <thead> <tr> <th>Dez.</th> <th>hex</th> <th>Art</th> </tr> </thead> <tbody> <tr> <td>12</td> <td>\$0C</td> <td>absolut</td> </tr> <tr> <td>3</td> <td>\$03</td> <td>not</td> </tr> <tr> <td>8</td> <td>\$08</td> <td>and</td> </tr> <tr> <td>14</td> <td>\$0E</td> <td>or</td> </tr> <tr> <td>6</td> <td>\$06</td> <td>exor</td> </tr> </tbody> </table> <p>Das nächste Nibble legt die Schreibfarbe fest, d.h. für ein Pixel werden die 4 Bit aus dem Video-RAM entsprechend dem unteren Nibble gelesen, mit der Schreibfarbe „verundet“ und wieder im Video-RAM abgelegt. Wird der Parameter mode auf absolut (=12) gesetzt, so wird der original Bildausschnitt an der Zielstelle abgelegt.</p>	Dez.	hex	Art	12	\$0C	absolut	3	\$03	not	8	\$08	and	14	\$0E	or	6	\$06	exor
Dez.	hex	Art																	
12	\$0C	absolut																	
3	\$03	not																	
8	\$08	and																	
14	\$0E	or																	
6	\$06	exor																	
<pre>BOX_READ: PROC((Xstart, Ystart) FIXED(15), (Breite, Hoehe) FIXED(15), feld STRUCT[REF CHAR(1)]) GLOBAL;</pre>	<p>Kopiert einen rechteckigen Bildausschnitt mit Breite und Hoehe beginnend bei dem Startpunkt (Xstart, Ystart) in den Speicherbereich, der mit feld angegeben wird. Dabei findet keine Überprüfung der Größe des Speicherbereiches statt. Ist das angegebene feld zu klein, ist der Absturz des Rechners so gut wie sicher!</p>																		
<pre>BOX_WRITE: PROC((Xstart, Ystart) FIXED(15), (Breite, Hoehe) FIXED(15), mode FIXED(15), feld STRUCT[REF CHAR(1)]) GLOBAL;</pre>	<p>Kopiert den Speicherbereich, der mit feld angegeben wird, auf einen rechteckigen Bildausschnitt von Breite und Höhe, beginnend bei dem Startpunkt (Xstart, Ystart). Dabei findet keine Überprüfung der Größe des Speicherbereiches statt. Ist das angegebene feld zu klein, wird der folgende Speicher auf den Bildschirm geschrieben. Der Parameter mode ist bei der Prozedur BOX_MOVE beschrieben.</p>																		
<pre>CIRCLE: PROC((Xmitte, Ymitte) FIXED(15), Radius FIXED(15), col FIXED(15)) GLOBAL;</pre>	<p>Zeichnet geschlossene Kreislinien auf den Bildschirm.</p>																		
<pre>CLEAR: PROC GLOBAL;</pre>	<p>Löscht den sichtbaren Bildschirmbereich mit der Farbe 0.</p>																		

COLOR_CLEAR: PROC(col FIXED(15)) GLOBAL;	Löscht den sichtbaren Bildschirmbereich mit der angegebenen Farbe col.
DISPLAY_MODE: PROC(mode FIXED(31)) GLOBAL;	Ermöglicht bei einigen Systemen die Umschaltung der Betriebsart des Grafiksystems (Text=0/Grafik=1).
DISPLAY_STATE: PROC(State FIXED(31)) GLOBAL;	Schaltet Display ein (State = 1) oder aus (State = 0)
DOT_LINE: PROC((Xstart, Ystart) FIXED(15), (XEnd, YEnd) FIXED(15), col FIXED(15), mask FIXED(15)) GLOBAL;	Zeichnet ein (strichpunktierte) Linie vom Punkt (Xstart, Ystart) zum Punkt (XEnd, YEnd) in der Farbe col. Mit mask wird das Linienmuster vorgegeben: -1: durchgezogene Linie TOFIXED 'AAAA'B4: einfach punktiert
DOT_LINE_RANGE: PROC((Xstart, Ystart) FIXED(15), (XEnd, YEnd) FIXED(15), col FIXED(15), mask FIXED(15), (YU, YO) FIXED(15), col2 FIXED(15),) GLOBAL;	Zeichnet ein (strichpunktierte) Linie vom Punkt (Xstart, Ystart) zum Punkt (XEnd, YEnd) in der Farbe col. Mit mask wird das Linienmuster vorgegeben: -1: durchgezogene Linie TOFIXED 'AAAA'B4: einfach punktiert Die Y-Koordinaten YU (unten) und YO (oben) sind Umschaltsschwellen für die Zeichenfarbe. Liegt ein zu zeichnender Bildpunkt unterhalb YU oder oberhalb YO, so wird er in der Farbe col2 anstatt in col dargestellt.
FILLED_HISTOGRAMM: PROC((XS, YS) FIXED(15) IDENT, count FIXED(15), col FIXED(15), b_col FIXED(15), width FIXED(15), ySave FIXED(15) IDENT) GLOBAL;	Zeichnet ein Histogramm aus count nicht gefüllten Balken der Breite width in der Farbe col. Die einzelnen Histogrammbalken werden durch ihre linke, obere Ecke (XS, YS) und die Breite width bestimmt. XS und YS müssen in getrennten FIXED(15)-Feldern gegeben werden. Die Y-Grundlinie des Histogramms wird durch die Initialwerte im Feld ySave gegeben. Zur Erhöhung der Zeichengeschwindigkeit bei wiederholter Darstellung eines Histogramms wird nur die Änderung der Balkenhöhe gegenüber der im Feld ySave gespeicherten Höhe neu gezeichnet.
GET_PEN: PROC RETURNS(FIXED(15)) GLOBAL;	Gibt die Farbe des aktuell gewählten Schreib-Pens zurück.
GET_BACK_PEN: PROC RETURNS(FIXED(15)) GLOBAL;	Gibt die Farbe des aktuell gewählten Hintergrund-Pens zurück.
GET_TEXT_FONT: PROC RETURNS(FIXED(15)) GLOBAL;	Gibt die Nummer des aktuellen Textfonts zurück (von SELECT_FONT oder SELECT_TEXT_FONT gesetzt). Anzahl und Art der verfügbaren Fonts sind abhängig vom eingesetzten Grafiksystem.
GET_TEXT_WIDTH: PROC(Xhoehe FIXED(15) IDENT, ybreite FIXED(15) IDENT) GLOBAL;	Liefert die Größe eines Zeichens des gewählten Zeichensatzes in den Variablen xhoehe und ybreite zurück.
GRAPH_CURSOR: PROC((x, y) FIXED(15),	Positioniert den Graphik-Cursor auf den Punkt (x, y). Mit status = 1 wird der Graphik-Cursor sichtbar, mit sta-

<pre> status FIXED(15)) GLOBAL; </pre>	<p>tus = 0 wird er unsichtbar.</p>												
<pre> GRAPH_CURSOR_SELECT: PROC(Cursor FIXED(31) IDENT, mode FIXED(15)) GLOBAL; </pre>	<p>Gibt eine Bitmap zur Darstellungsform des Graphik-Cursors vor. <code>Cursor</code> muß das erste Element eines <code>FIXED(31)</code>-Feldes mit 32 Elementen sein; damit wird ein 32x32 Bit-Cursor beschrieben. Gesetzte Bits entsprechen darzustellenden Pixeln. <code>mode</code> gibt die Darstellungsart des Cursors an.</p>												
<pre> HIDE_CURSOR: PROC GLOBAL; </pre>	<p>Schaltet den Graphik-Cursor aus.</p>												
<pre> HISTOGRAMM: PROC((XS, YS) FIXED(15) IDENT, count FIXED(15), col FIXED(15), b_col FIXED(15), (offs, width) FIXED(15), ySave FIXED(15) IDENT) GLOBAL; </pre>	<p>Zeichnet ein Histogramm aus <code>count</code> nicht gefüllten Balken der Breite <code>width</code> in der Farbe <code>col</code>. Die Y-Grundlinie des Histogramms wird in <code>offs</code> gegeben. Die einzelnen Histogrammbalken werden durch ihre linke, obere Ecke (<code>XS</code>, <code>YS</code>) und die Breite <code>width</code> bestimmt. <code>XS</code> und <code>YS</code> müssen in getrennten <code>FIXED(15)</code>-Feldern gegeben werden. Zur Erhöhung der Zeichengeschwindigkeit bei wiederholter Darstellung eines Histogramms wird nur die Änderung der Balkenhöhe gegenüber der im Feld <code>ySave</code> gespeicherten Höhe neu gezeichnet. Bei der erstmaligen Darstellung eines Histogramms muss <code>ySave</code> daher vollständig mit dem Wert <code>offs</code> initialisiert werden.</p>												
<pre> PEN: PROC(col FIXED(15)) GLOBAL; </pre>	<p>Setzt die angegebene Farbe <code>col</code> als Schreibfarbe. Bei folgenden Operationen, die keine eigene HinSchreibfarbe spezifizieren, wird <code>col</code> als Schreibfarbe verwendet.</p>												
<pre> PLANE: PROC(Farbnummer FIXED(15), Palettenwert FIXED(31)) GLOBAL; </pre>	<p>Setzt den Wert eines Paletteneintrags. Für Farbnummer sind Werte von 0 bis 15 zulässig. Folgende Bits vom Palettenwert werden genutzt (abhängig vom Grafiksysteem):</p> <table border="1"> <thead> <tr> <th>Farbe</th> <th>Bit (PEARL)</th> <th>Bit(Assembler)</th> </tr> </thead> <tbody> <tr> <td>Rot</td> <td>12 ... 4</td> <td>18 ... 20</td> </tr> <tr> <td>Grün</td> <td>20 ...22</td> <td>10 ... 12</td> </tr> <tr> <td>Blau</td> <td>28 ...30</td> <td>2 ... 4</td> </tr> </tbody> </table> <p>Der Aufbau des Langwortes Palettenwert läßt sich durch die Einzelbit-Darstellung [---- ---- ---r rr-- ---g gg-- ---b bb--] veranschaulichen.</p>	Farbe	Bit (PEARL)	Bit(Assembler)	Rot	12 ... 4	18 ... 20	Grün	20 ...22	10 ... 12	Blau	28 ...30	2 ... 4
Farbe	Bit (PEARL)	Bit(Assembler)											
Rot	12 ... 4	18 ... 20											
Grün	20 ...22	10 ... 12											
Blau	28 ...30	2 ... 4											
<pre> PLINIT: PROC GLOBAL; </pre>	<p>Initialisiert den Display-Prozessor. Muß vor Benutzung der Grafik-Routinen einmal aufgerufen werden.</p>												
<pre> POLYGON: PROC((x, y) FIXED(15) IDENT, cnt FIXED(15), col FIXED(15)) GLOBAL; </pre>	<p>Zeichnet einen (nicht geschlossenen) Linienzug mit <code>cnt</code> Stützpunkten in der Farbe <code>col</code>. <code>x</code> und <code>y</code> müssen in getrennten, eindimensionalen Feldern abgelegt sein.</p>												
<pre> SELECT_FONT: PROC(Font FIXED(15)) GLOBAL; </pre>	<p>Wählt den angegebenen Font als aktuellen Textfont aus. Anzahl und Art der verfügbaren Fonts sind abhängig</p>												

<code>) GLOBAL;</code>	vom eingesetzten Grafiksystem.
<code>SELECT_TEXT_FONT: PROC(Font FIXED(15)) GLOBAL;</code>	Wählt den angegebenen Font als aktuellen Textfont aus. Anzahl und Art der verfügbaren Fonts sind abhängig vom eingesetzten Grafiksystem.
<code>SET_TEXT_WIDTH: PROC(Xhöhe FIXED(15) IDENT, Ybreite FIXED(15) IDENT) GLOBAL;</code>	Setzt die Größe eines Zeichens für die Funktionen <code>BIN_TEXT</code> und <code>TEXT</code> .
<code>SET_VGA_SCREEN: PROC(Screen FIXED(15)) RETURNS(FIXED(15)) GLOBAL;</code>	Wählt den angegebenen Screen als Ziel der folgenden Graphikoperationen aus. Der Rückgabewert ist <code>-1</code> , falls der angegebene Screen im aktuellen Grafiksystem nicht vorhanden ist.
<code>TEXT: PROC((x, y) FIXED(15), col FIXED(15), string STRUCT [str CHAR(255), len FIXED(15)]) GLOBAL;</code>	Schreibt die in <code>string.str</code> abgelegte Zeichenkette beginnend auf der Position <code>(x, y)</code> (linke, obere Ecke des ersten Zeichens) in der Farbe <code>col</code> auf den Bildschirm. Die Ausgabe endet, wenn <code>string.len</code> Zeichen ausgegeben wurden. Die Schreibrichtung ist horizontal (x-Richtung).
<code>VGA_GET_PLANE: PROC(Plane FIXED(15), (R, G, B) FIXED(15) IDENT) GLOBAL;</code>	Liest aus der Palette <code>Plane</code> den enthaltenen RGB-Wert aus.
<code>VGA_LOCK_TEXT: PROC RETURNS(FIXED(15)) GLOBAL;</code>	Sperrt das Grafiksystem gegenüber den Aktionen der Terminalemulation. Diese Funktion muß jeweils vor einem Block von zusammenhängenden Grafikoperationen aufgerufen werden, falls gleichzeitig die Terminalemulation genutzt werden soll. Der Rückgabewert ist <code>1</code> , falls die Operation erfolgreich war.
<code>VGA_SET_PLANE: PROC(Plane FIXED(15), (R, G, B) FIXED(15)) GLOBAL;</code>	Setzt die Palette <code>Plane</code> auf einen angegebenen RGB-Wert. <code>Plane</code> wird beim Zeichnen über die <code>col</code> angesprochen
<code>VGA_UNLOCK_TEXT: PROC RETURNS(FIXED(15)) GLOBAL;</code>	Gibt das Grafiksystem nach einem Aufruf von <code>VGA_LOCK_TEXT</code> wieder für Aktivitäten der Terminalemulation frei. Der Rückgabewert ist <code>1</code> , falls die Operation erfolgreich war.
<code>WIDTH: PROC(Xwidth FIXED(15) IDENT, Ywidth FIXED(15) IDENT) GLOBAL;</code>	Gibt in den Variablen <code>Xwidth</code> und <code>Ywidth</code> die Größe des Bildschirms in Pixel zurück.

7.4.4 Zeichensatz

Der Zeichensatz für die Funktion TEXT enthält folgende Zeichen oder den eingebauten Zeichensatz des Displays:

0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
	☺	☹	♥	♦	♣	♠	●	■	○	◼	♂	♀	♪	♫	*
16	17	18	19	20	21	22	23	24	25	26	27	28	29	30	31
▶	◀	↕	!!	¶	§	■	↕	↑	↓	→	←	⊥	↔	▲	▼
32	33	34	35	36	37	38	39	40	41	42	43	44	45	46	47
	!	"	#	\$	%	&	'	()	*	+	,	-	.	/
48	49	50	51	52	53	54	55	56	57	58	59	60	61	62	63
0	1	2	3	4	5	6	7	8	9	:	;	<	=	>	?
64	65	66	67	68	69	70	71	72	73	74	75	76	77	78	79
@	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O
80	81	82	83	84	85	86	87	88	89	90	91	92	93	94	95
P	Q	R	S	T	U	V	W	X	Y	Z	[\]	^	_
96	97	98	99	100	101	102	103	104	105	106	107	108	109	110	111
`	a	b	c	d	e	f	g	h	i	j	k	l	m	n	o
112	113	114	115	116	117	118	119	120	121	122	123	124	125	126	127
p	q	r	s	t	u	v	w	x	y	z	{		}	~	△
128	129	130	131	132	133	134	135	136	137	138	139	140	141	142	143
	ú	é	ξ	ä	à	á	ç	ê	ë	è	ï	î	ì	Ä	Ã
144	145	146	147	148	149	150	151	152	153	154	155	156	157	158	159
É	æ	Æ	ô	ö	ò	û	ù	ÿ	Ö	Ü	ø	£	¥	Pls	f
160	161	162	163	164	165	166	167	168	169	170	171	172	173	174	175
á	é	í	ú	ñ	Ñ			¿	Γ	γ	½	¼	i	«	»
176	177	178	179	180	181	182	183	184	185	186	187	188	189	190	191
☼	☼	☼		┌	┐	└	┘	π	τ	∥	∥	π	∥	∥	γ
192	193	194	195	196	197	198	199	200	201	202	203	204	205	206	207
L	⊥	⊥	┌	—	┐	└	┘	⊥	⊥	⊥	⊥	⊥	⊥	⊥	⊥
208	209	210	211	212	213	214	215	216	217	218	219	220	221	222	223
⊥	⊥	⊥	⊥	⊥	⊥	⊥	⊥	⊥	⊥	⊥	■	■	■	■	■
224	225	226	227	228	229	230	231	232	233	234	235	236	237	238	239
α	β	Γ	π	Σ	σ	μ	Τ	Φ	Θ	Ω	δ	θ	ø	€	Π
240	241	242	243	244	245	246	247	248	249	250	251	252	253	254	255
≡	±	≥	≤			÷	≈	°	▪	·	√	n	²	■	