Inhalt

1. Einleitung ... 2
2. Verfügbarkeit und Art der Spezialfunktionen ... 2
3. Erläuterungen zu den Spezialfunktionen ... 3
3.1.1. Beispiel (Delta) ... 3
3.1.2. Beispiel (Nano-C) ... 3
3.2. Spezialfunktion 1: Achse initialisieren ... 3
3.2.1. Beschreibungsblock ... 4
3.2.2. Hinweis ... 4
3.2.3. Beispiel (steuerungsunabhängig) ... 4
3.3. Spezialfunktion 4: BCD in HEX Wandlung ... 5
3.3.1. Anwendung ... 5
3.3.2. Beispiel (steuerungsunabhängig) ... 5
3.3.3. Hinweis ... 5
3.4. Spezialfunktion 5: HEX in BCD Wandlung ... 6
3.5. Spezialfunktion 7: Netzwerkadresse definieren ... 6
3.6. Hinweis zu den Spezialfunktionen 10, 11 und 12 ... 6
3.7. Spezialfunktion 10: CAN: verbinden ... 7
3.7.1. Beispiel ... 7
3.8. Spezialfunktion 11: CAN: Daten senden ... 8
3.8.1. Beispiel ... 8
3.9. Spezialfunktion 12: CAN: Daten holen ... 9
3.9.1. Beispiel ... 9
3.9.2. Hinweis ... 9
3.10. Spezialfunktion 20: Quadratwurzel ... 10
3.11. Spezialfunktion 21: Sinus ... 10
3.12. Spezialfunktion 22: Cosinus ... 10
3.13. Spezialfunktion 23: Tangens ... 10
3.14. Spezialfunktion 24: Arcus Sinus ... 10
3.15. Spezialfunktion 25: Arcus Cosinus ... 11
3.16. Spezialfunktion 26: Arcus Tangens ... 11
3.17. Spezialfunktion 27: Exponentialfunktion ("e hoch") ... 12
3.17.1. Hinweis ... 12
3.18. Spezialfunktion 28: Natürlicher Logarithmus ("ln") ... 12
3.19. Spezialfunktion 29: Absolutwert ... 12
3.20. Spezialfunktion 30: MOD ... 12
3.20.1. Beispiel (Delta) ... 13
3.20.2. Hinweis ... 13
3.21. Spezialfunktion 40: Multimaster Tokenweitergabe ... 13
3.21.1. Initialisierung der Net1-Schnittstelle ... 13
3.21.2. Initialisierung der Net2-Schnittstelle ... 14
3.21.3. Programmierung ... 14
3.21.4. Master-Master-Kopplung / Multimaster-Betrieb ... 14
3.22. Spezialfunktion 252: Checksumme ermitteln ... 15

Alle Rechte vorbehalten.
Die Firma Jetter AG behält sich das Recht vor, Änderungen an ihren Produkten vorzunehmen, die der technischen Weiterentwicklung dienen. Diese Änderungen werden nicht notwendigerweise in jedem Einzelfall dokumentiert.

Diese Application Note und die darin enthaltenen Informationen wurden mit der gebotenen Sorgfalt zusammengestellt. Die Firma Jetter AG übernimmt jedoch keine Gewähr für Druckfehler oder andere Fehler oder daraus entstehende Schäden.

Die hier genannten Marken und Produktnamen sind Warenzeichen oder eingetragene Warenzeichen der jeweiligen Titelhalter.

Jetter AG
Gräterstrasse 2
D-71642 Ludwigsburg
Germany

Telefon - Zentrale: 07141/2550-0
Telefon - Vertrieb: 07141/2550-530
Telefon - Technische Hotline: 07141/2550-444

Telefax: 07141/2550-425
E-Mail - Vertrieb: sales@jetter.de
E-Mail - Technische Hotline: hotline@jetter.de
Internetadresse: http://www.jetter.de

1. Einleitung

Die Programmiersprache Sympas hat einen begrenzten Befehlssatz. Wenn weitere Befehle notwendig sind, werden dafür Spezialfunktionen verwendet.

2. Verfügbarkeit und Art der Spezialfunktionen

Folgende Sympas Spezialfunktionen sind auf den einzelnen Steuerungen verfügbar (Tabelle 1):

Spezial-funktion	Wirkung	PASE-E	Delta	Nano-A und Nano-B	Nano-C	PC-PPLC
0	Neuerkennung Module					X
1	Initialisierung Slaves	X	X			
4	BCD → HEX	X	X	X	X	X
5	HEX → BCD	X	X	X	X	X
7	Netzwerkadresse definieren	X				
10	CAN: verbinden		X			
11	CAN: Daten senden		X			
12	CAN: Daten holen		X			
20	Quadratwurzel	X	X		X	X
21	Sinus	X	X		X	X
22	Cosinus	X	X		X	X
23	Tangens	X	X		X	X
24	Arcus Sinus	X	X		X	X
25	Arcus Cosinus	X	X		X	X
26	Arcus Tangens	X	X		X	X
27	e^x	X	X		X	X
28	Logarithmus nat.	X	X		X	X
29	Absolutwert	X	X			
30	Modulo	X	X			
40	Multimaster Tokenweitergabe		X			
252	Checksumme ermitteln	X	X			

Tabelle 1

Die Mikro verfügt über keine Spezialfunktionen.

3. Erläuterungen zu den Spezialfunktionen

3.1. Aufruf einer Spezialfunktion

Die `SPEZIALFUNKTION` ist fester Bestandteil des Sympas Befehlssatzes. Eine Spezialfunktion wird immer mit zwei Parametern aufgerufen:

```
SPEZIALFUNKTION [#NUMMER, p1=PARAMETER1, p2= PARAMETER2]
```

Die `NUMMER` der verfügbaren Spezialfunktionen kann aus Tabelle 1 entnommen werden. Der erste Parameter (`PARAMETER1`) stellt die Nummer des Registers dar, das den Operanden enthält (Quellregister). Der zweite Parameter (`PARAMETER2`) stellt die Nummer des Registers dar, in das das Ergebnis der Funktion geschrieben wird (Zielregister).

3.1.1. Beispiel (Delta)

```
SPEZIALFUNKTION [#21, P1=62208, P2=62209]
```

Berechnet den Sinus aus der Zahl im Gleitkommaregister 62208 und speichert das Ergebnis im Gleitkommaregister 62209 ab.

Es ist im Prinzip zugelassen, Integerregister für die Parameterübergabe bzw. für das Ergebnis zu verwenden, jedoch aufgrund des Wertebereichs meist nicht sinnvoll.
Die Parameter `PARAMETER1` und `PARAMETER2` können bei allen Funktionen auch indirekt angegeben werden.

3.1.2. Beispiel (Nano-C)

```
SPEZIALFUNKTION [#20, p1=R(100), p2=R(101)]
```

3.2. Spezialfunktion 0: Neuerkennung Module

Die Spezialfunktion 0 ist nur für die PC-PPLC gedacht. Bei der PC-PPLC handelt es sich um eine Steuerung (Nano-C mit verbesserter Performance) auf einer PC-Einsteckkarte. Wenn der PC, in dem die PC-PPLC sitzt, bereits gebootet ist, ohne daß die Anlage - und damit die Module - an Spannung liegen, muß es die Möglichkeit einer Neuerkennung der angeschlossenen Module geben, ohne den PC neu starten zu müssen. Hierfür wird die Spezialfunktion 0 verwendet. Als Parameter werden beliebige Werte verwendet – sie werden nicht beachtet. Im einfachsten Fall sieht die Spezialfunktion 0 so aus:

```
SPEZIALFUNKTION [#0, p1=0, p2=0]
```

Die Spezialfunktion 0 wird nur einmal am Beginn des Programms gegeben, dann nicht mehr.

3.3. Spezialfunktion 1: Achse initialisieren

```
SPEZIALFUNKTION [#1, p1= rBeschreibungsblock, p2= rBasisregister]
```

Die Spezialfunktion 1 wird zum Kopieren von Registern in Modulregister verwendet. Es können maximal 99 Register kopiert werden. Die Register, in die die Werte geschrieben werden sollen, müssen in diesem Fall jedoch nicht in einem festen Block sein. Es können auch einzelne Register geladen werden.

Die Spezialfunktion 1 wurde für das Initialisieren von Achskarten entwickelt, weil dort sehr viele Register geladen werden müssen. Die Funktion hat zwei Parameter: Der erste (`rBeschreibungsblock`) gibt die Nummer des ersten Registers eines Beschreibungsblocks an, der zweite Parameter (`rBasisregister`) ist ein Basisregister, zu dem ein Offset addiert wird. Die Parameter `rBeschreibungsblock` und

rBasisregister können auch indirekt adressiert werden. Der Beschreibungsblock enthält alle Angaben darüber, wieviele und welche Register kopiert werden sollen und zwar nach folgenden Konventionen.

3.3.1. Erläuterungen zum Beschreibungsblock

3.3.1.1. Aufbau

Registernummer	Inhalt
rBeschreibungsblock	Anzahl zu kopierender Register
rBeschreibungsblock + 1	Erster Offset zur Registernummer rBasisregister
rBeschreibungsblock + 2	Wert, der ins erste Register geschrieben wird
rBeschreibungsblock + 3	Zweiter Offset zur Registernummer rBasisregister
rBeschreibungsblock + 4	Wert, der ins zweite Register geschrieben wird
rBeschreibungsblock + 5	usw.

3.3.1.2. Hinweis

Für jedes zu beschreibende Register werden zwei Register des Beschreibungsblocks gebraucht, beispielsweise Register rBeschreibungsblock + 1 und Register rBeschreibungsblock + 2. Diese beiden Werte haben zur Folge, daß ins Register mit der Nummer

rBasisregister + R(rBeschreibungsblock + 1)

der Wert

R(rBeschreibungsblock + 2)

geschrieben wird. Dieser Sachverhalt kann am besten anhand eines Beispiels erklärt werden.

3.3.1.3. Beispiel (steuerungsunabhängig)

```
SPEZIALFUNKTION [#1, p1=100, p2=1000]
```

Der Beschreibungsblock beginnt also bei Register 100 (rBeschreibungsblock = 100).
Für das Beispiel sollen sich folgende Werte in den Registern 100 und folgende befinden:

Register 100 = 4
Register 101 = 10
Register 102 = 4500
Register 103 = 11
Register 104 = 199
Register 105 = 15
Register 106 = 912
Register 107 = 19
Register 108 = 9999

Nach Ausführung der Spezialfunktion enthalten folgende Register die Werte:

Register 1010 = 4500
Register 1011 = 199
Register 1015 = 912
Register 1019 = 9999

Erstes zu beschreibendes Register:
Registernr. = rBasisregister + R(rBeschreibungsblock+1) = 1000 + R(100+1) = 1000 + 10 = 1010
Wert für Register 1010: R(rBeschreibungsblock+2) = R(100+2) = 4500

Zweites zu beschreibendes Register:
Registernr. = `rBasisregister` + R(`rBeschreibungsblock`+3) = 1000 + R(100+3) = 1000 + 11 = 1011
Wert für Register 1011: R(`rBeschreibungsblock`+4) = R(100+4) = 199

Drittes zu beschreibendes Register:
Registernr. = `rBasisregister` + R(`rBeschreibungsblock`+5) = 1000 + R(100+5) = 1000 + 15 = 1015
Wert für Register 1015: R(`rBeschreibungsblock`+6) = R(100+6) = 912

Viertes und letztes (`rBasisregister` = Register 100 = 4) zu beschreibendes Register:
Registernr. = `rBasisregister` + R(`rBeschreibungsblock`+7) = 1000 + R(100+7) = 1000 + 19 = 1019
Wert für Register 1019: R(`rBeschreibungsblock`+8) = R(100+8) = 9999

3.4. Spezialfunktion 4: BCD in HEX Wandlung

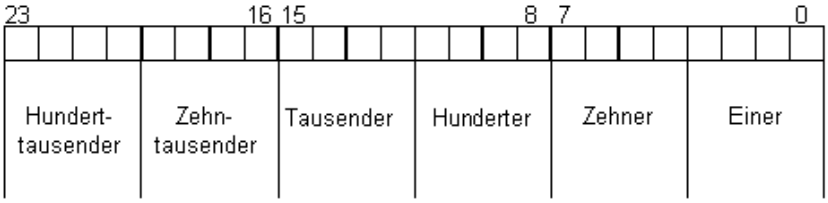
Die Spezialfunktion 4 dient der Umwandlung von **b**inär **c**odierten **D**ezimalzahlen (BCD) in Dezimalzahlen.

3.4.1. Anwendung

Werden beispielsweise BCD-Schalter an digitale Eingänge einer Steuerung angeschlossen, dann liegen die Zahlen, die über die Schalter eingestellt werden, im überlagerten Register dieser Eingänge im BCD-Format vor. Um die Eingabe als Dezimalzahl nutzen zu können, muß sie zunächst umgewandelt werden. Dies geschieht mit der Spezialfunktion 4.

```
SPEZIALFUNKTION [#4, p1=Parameter1, p2=Parameter2]
```

Die Bits des Quellregisters werden als BCD-Zahl interpretiert, in eine Binärzahl umgewandelt und in das Zielregister geschrieben. Dabei stellen jeweils vier Bits des Quellregisters eine Dezimalstelle dar. Es können maximal sechs Dezimalstellen gewandelt werden.



3.4.2. Beispiel (steuerungsunabhängig)

Das Register 100 habe den Wert: 0101 1000 0011 0110 = 22582

Die entsprechende BCD-Zahl lautet: 5 8 3 6 = 5836

Der Befehl

```
SPEZIALFUNKTION [#4, p1=100, p2=101]
```

hat nun zur Folge, daß in das Register 101 der Wert 5836 geschrieben wird.

3.4.3. Hinweis

Für die Spezialfunktion 4 können nur Integerregister, keine Gleitkommaregister verwendet werden.

3.5. Spezialfunktion 5: HEX in BCD Wandlung

Die Spezialfunktion 5 dient der Umwandlung von Dezimalzahlen in **b**inär **c**odierte **D**ezimalzahlen (BCD). Sie entspricht somit der Umkehrung der Spezialfunktion 4.

3.6. Spezialfunktion 7: Netzwerkadresse definieren

Hinweis: Die Anwendung der Spezialfunktion 7 ist nur bei der PASE-Eplus sinnvoll.

Bei der PASE-Eplus wird die JetWay-Netzwerkadresse über vier DIL-Schalter auf der E-CPUplus eingestellt. Folglich können nur Netzwerkadressen von 1 bis 15 vergeben werden. Für die Vergabe höherer Netzwerkadressen wird die Spezialfunktion 7 verwendet.

```
SPEZIALFUNKTION [#7, p1=Quellregister, p2=Zielregister]
```

Das Quellregister enthält die neue Netzwerkadresse (Wertebereich: 1 bis 42) und das Zielregister die neue Baudrate, die für den JetWay verwendet werden soll. Dabei bedeuten die Werte 30 = 19200 Baud und 5 = 115200 Baud.

3.7. Spezialfunktionen 10, 11 und 12

3.7.1. Hinweis zu den Spezialfunktionen 10, 11 und 12

Die Anwendung der Spezialfunktionen 10, 11 und 12 ist nur bei der Delta sinnvoll.

Ab der D-CPU-Version 1.05 ist es möglich, über das D-CAN1 Submodul auf Submodulsteckplatz eins oder zwei auf der D-CPU, Registerwerte an eine oder mehrere andere Delta oder PASE-E Steuerungen über einen CAN-Bus zu übertragen. Dabei können einzelne Registerwerte oder ganze Registerblöcke mit bis zu 64 Werten gesendet werden.

Die von einer Steuerung gesendeten Werte werden von allen Teilnehmern am Bus empfangen und zwischengespeichert. Über das Anwenderprogramm kann der empfangene Registerblock vom D-CAN1 Submodul gelesen werden. Von Busteilnehmern empfangene Registerwerte, die nicht benötigt werden, brauchen auch nicht gelesen zu werden.

Es können maximal 12 Teilnehmer angeschlossen sein. Auf jedem Modul ist für jeden Busteilnehmer ein Registerblock von 64 Registern reserviert, in denen die empfangenen Registerwerte abgelegt werden.
Die Kommunikation zwischen D-CPU und D-CAN1 Submodul wird über Register direkt auf dem Submodul und die drei Spezialfunktionen 10, 11 und 12 organisiert. Bei jeder der drei Spezialfunktionen enthält der erste Parameter die Nummer des ersten Registers eines Parameterregisterblocks, der die weiteren Funktionsparameter enthält. Der zweite Parameter enthält die Nummer des ersten Registers des Ergebnisregisterblocks.

Da das flexible Konzept der Delta unterschiedlichste Bestückungsvarianten - auch mit Modulen von Fremdherstellern, die nicht automatisch erkannt werden können - zulässt, muß dem Betriebssystem der Delta vor dem ersten Zugriff (Register oder Spezialfunktion) auf das D-CAN1 Submodul mitgeteilt werden, welches Submodul sich auf welchem Submodulsteckplatz befindet. Dazu wurde für jeden der beiden Submodulsteckplätze ein Register vergeben:

Steckplatz 1: Register 61688
Steckplatz 2: Register 61689

Je nachdem, auf welchem Submodulsteckplatz das D-CAN1 Submodul montiert ist, muss das entsprechende Register zu Beginn des Anwenderprogramms geladen werden.
Für das D-CAN1 Submodul mit Master-Master-Kopplung in der Version 1.xx muss der Wert **1** geladen werden.

Hinweis: Zum Betrieb des D-CAN1 Submodul ist eine D-CPU mit der Platinenrevision **03** nötig!

3.7.2. Spezialfunktion 10: CAN: verbinden

Nach dem Einschalten der Steuerung nimmt das D-CAN1 Submodul noch nicht am Busverkehr teil. Es können also noch keine Werte gesendet oder empfangen werden (die rote LED blinkt). Erst über die Spezialfunktion 10 wird die Verbindung zum Bus hergestellt (rote LED erlischt).

```
SPEZIALFUNKTION [#10, p1=ParamRegNum, p2=ErgRegNum]
```

Der Parameterregisterblock (Beginn bei p1) umfasst 5 Register:

Registernummer	Bedeutung
p1 + 0	Submodulsteckplatz des D-CAN1 Submoduls auf der D-CPU, 1 oder 2
p1 + 1	Teilnehmernummer 2 bis 13 Es ist darauf zu achten, dass jede Nummer nur einmal in diesem Verbund vorkommt
p1 + 2	Baudrate, 1 bis 4 1 = 125 kBaud 2 = 250 kBaud 3 = 500 kBaud 4 = 1000 kBaud Alle Teilnehmer müssen mit der gleichen Baudrate arbeiten. Die maximal mögliche Baudrate ist von der Gesamtleitungslänge abhängig.
p1 + 3	reserviert, sollte -1 enthalten
p1 + 4	reserviert, sollte 0 enthalten

Der Ergebnisregisterblock (Beginn bei p2) umfasst 2 Register:

Registernummer	Bedeutung
p2 + 0	Softwareversionsnummer * 100, (z.B. Registerinhalt 105 → Version 1.05)
p2 + 1	Modultyp / Hardwareversion

Die Anbindung kann mehrere Millisekunden dauern. Während dieser Zeit ist das Busy-Bit im Statusregister (Register 63x00, x = 3 für Submodulsteckplatz 1 und x = 4 für Submodulsteckplatz 2, Bit 16) des D-CAN1 Submoduls gesetzt. Bevor mit den nachfolgend beschriebenen Funktionen auf das Modul zugegriffen werden kann, muss abgewartet werden bis das Busy-Bit gelöscht ist.

3.7.2.1. Beispiel

Vor dem Aufruf der Spezialfunktion 10 werden die Register 100 bis 104 folgendermaßen beschrieben:

Register 100 = 1	Submodulsteckplatz
Register 101 = 5	Teilnehmernummer
Register 102 = 3	Baudrate
Register 103 = -1	reserviert
Register 104 = 0	reserviert

```
SPEZIALFUNKTION [#10, p1=100, p2=110]
```

Nachdem die Spezialfunktion 10 erfolgreich abgearbeitet wurde, nimmt das D-CAN1 Submodul auf Submodulsteckplatz 1 (Register 100) mit der Teilnehmernummer 5 (Register 101) und der Baudrate 500 kBaud (Register 102) am Canbus teil. Register 110 enthält jetzt die Versionsnummer der Firmware des D-CAN1 Submoduls und Register 111 die Hardwareversion.

3.7.3. Spezialfunktion 11: CAN: Daten senden

Nachdem mit der Spezialfunktion 10 die Verbindung zum CAN-Bus hergestellt worden ist, können mit der Spezialfunktion 11 Registerinhalte über den CAN-Bus an die anderen Steuerungen geschickt werden.

```
SPEZIALFUNKTION [#11, p1=ParamRegNum, p2=ErgRegNum]
```

Der Parameterregisterblock (Beginn bei p1) hat folgenden Aufbau:

Registernummer	Bedeutung
p1 + 0	Submodulsteckplatz des D-CAN1 Submoduls auf der D-CPU, 1 oder 2
p1 + 1	Anzahl zu sendender Register: 1 bis 64
p1 + 2	erste Registernummer im Registerblock der D-CAN1 Submodule der anderen Steuerungen: 0 bis 63
p1 + 3	erstes zu sendendes Register
p1 + 4	zweites zu sendendes Register
p1 + ...	weitere zu sendende Register

Der Ergebnisregisterblock wird zur Zeit nicht benutzt. Für zukünftige Erweiterungen sollten 2 Register reserviert werden.

Nach Beendigung dieser Funktion ist das Busy-Bit im Statusregister (Register 63x00, x = 3 für Submodulsteckplatz 1 und x = 4 für Submodulsteckplatz 2, Bit 16) des D-CAN1 Submoduls gesetzt und es beginnt mit dem Senden. Durch Abfragen auf Rücksetzen des Busy-Bits kann überprüft werden, wann der Registerblock vollständig übertragen worden ist.

3.7.3.1. Beispiel

Über das D-CAN1 Submodul auf dem zweiten Submodulsteckplatz sollen 3 Register übertragen werden. Diese Werte sollen in den D-CAN1 Submodulen der anderen Steuerungen ab der Speicherplatznummer 20 abgelegt werden.

Register 200 = 2	Submodulsteckplatz
Register 201 = 3	Anzahl Werte
Register 202 = 20	Zielregisternummer
Register 203 = 123456	Wert 1
Register 204 = 40404	Wert 2
Register 205 = -98765	Wert 3

```
SPEZIALFUNKTION [#11, p1=200, p2=210]
```

Nachdem die Spezialfunktion 11 erfolgreich abgearbeitet wurde, enthalten die Register 20 bis 22 des Registerblocks auf den D-CAN1 Submodulen der anderen Steuerungen die gesendeten Werte:

Register 20 = 123456
Register 21 = 40404
Register 22 = -98765

Die Register 210 (p2) und folgende werden zur Zeit noch nicht beeinflusst (die Register 210 und 211 sollten jedoch reserviert werden).

3.7.4. Spezialfunktion 12: CAN: Daten holen

Ist ein Registerblock von einer anderen Steuerung vollständig empfangen worden, so wird dies durch Setzen eines entsprechenden Bits im Statusregister (Register 63x00, x = 3 für Submodulsteckplatz 1 und x = 4 für Submodulsteckplatz 2) angezeigt. Für jeden der maximal zwölf Busteilnehmer ist ein Bit im Statusregister vorhanden, so dass aus der Bitposition ermittelt werden kann, von welcher Steuerung gerade Werte empfangen worden sind.

In zwei weiteren Registern (63x10 und folgende, x = 3 für Submodulsteckplatz 1 und x = 4 für Submodulsteckplatz 2) pro Teilnehmer kann gelesen werden, in welchem Bereich des Registerblocks für diesen Teilnehmer die empfangenen Werte abgespeichert worden sind.
Über die Spezialfunktion 12 kann ein Registerblock des gewünschten Teilnehmers gelesen werden:

```
SPEZIALFUNKTION [#12, p1=ParamRegNum, p2=ZielRegNum]
```

Der Parameterregisterblock (Beginn bei p1) umfasst 4 Register:

Registernummer	Bedeutung
p1 + 0	Submodulsteckplatz des D-CAN1 Submoduls auf der D-CPU, 1 oder 2
p1 + 1	Anzahl zu lesender Register: 1 bis 64
p1 + 2	Nummer des ersten zu lesenden Registers aus dem Registerblock: 0 bis 63
p1 + 3	Nummer des Teilnehmers, dessen Werte gelesen werden sollen: 2 bis 13

Im Zielregisterblock werden die gelesenen Werte abgelegt. Dabei wird im ersten - durch p2 bestimmten - Register die Information gespeichert, ob sich der Registerblock seit dem letzten Auslesen geändert hat. Es spiegelt damit den Zustand des entsprechenden Bits im Statusregister wider. Wurden keine neuen Werte empfangen, so enthält das Register den Wert 0. Sind Werte empfangen worden, so enthält das Register den Wert 255. In den folgenden Registern sind die empfangenen Registerwerte abgelegt.

Zu Beginn des Auslesens setzt das D-CAN1 Submodul das 'Block geändert'-Bit im Statusregister zurück.

3.7.4.1. Beispiel

Vom D-CAN1 Submodul auf dem ersten Submodulsteckplatz sollen 3 Register aus dem Registerblock des Teilnehmers mit der Nummer 7 ab der Wertenummer 10 gelesen und ab dem Register 310 in Anwenderregistern der D-CPU gespeichert werden.

Register 300 = 1	Submodulsteckplatz
Register 301 = 3	Anzahl Werte
Register 302 = 10	Quellregisternummer
Register 303 = 7	Teilnehmernummer

```
SPEZIALFUNKTION [#12, p1=300, p2=310]
```

Nachdem die Spezialfunktion 12 erfolgreich abgearbeitet wurde, enthält das Register 310 der D-CPU den Wert 0, wenn vom Busteilnehmer mit der Nummer 7 seit dem letzten Lesen keine neuen Werte übertragen worden sind, oder den Wert 255, wenn seither neue Werte angekommen sind. Die Register 311 bis 313 enthalten die empfangenen Werte.

3.7.4.2. Hinweis

Ist ein Registerblock empfangen worden, so wird zwar das 'Block geändert'-Bit im Statusregister gesetzt, der entsprechende Registerblock jedoch nicht gesperrt. Das bedeutet, dass die Daten gelöscht werden, wenn der sendende Teilnehmer nach der ersten Übertragung sofort eine zweite Übertragung anschließt und der Empfänger die Werte der ersten Übertragung noch nicht abgeholt hat. Es ist sogar möglich, dass während die D-CPU einen Registerblock ausliest, über den CAN-Bus ein neuer Registerblock ankommt und damit die D-CPU einen Teil der Werte der ersten Übertragung und einen Teil der zweiten Übertragung liest. Die Daten innerhalb eines Registers sind jedoch konsistent.

3.8. Spezialfunktion 20: Quadratwurzel

Wertebereich Argument: 0 und positive Zahlen
Wertebereich Ergebnis: 0 und positive Zahlen
Mögliche Fehler: Negative Zahl als Argument
Rechenzeit: ca. 1,1 ms

3.9. Spezialfunktion 21: Sinus

Wertebereich Argument: -1000 bis +1000 (Bogenmaß!)
Wertebereich Ergebnis: -1 bis +1
Mögliche Fehler: keine
Rechenzeit: ca. 4,5 bis 5 ms

3.10. Spezialfunktion 22: Cosinus

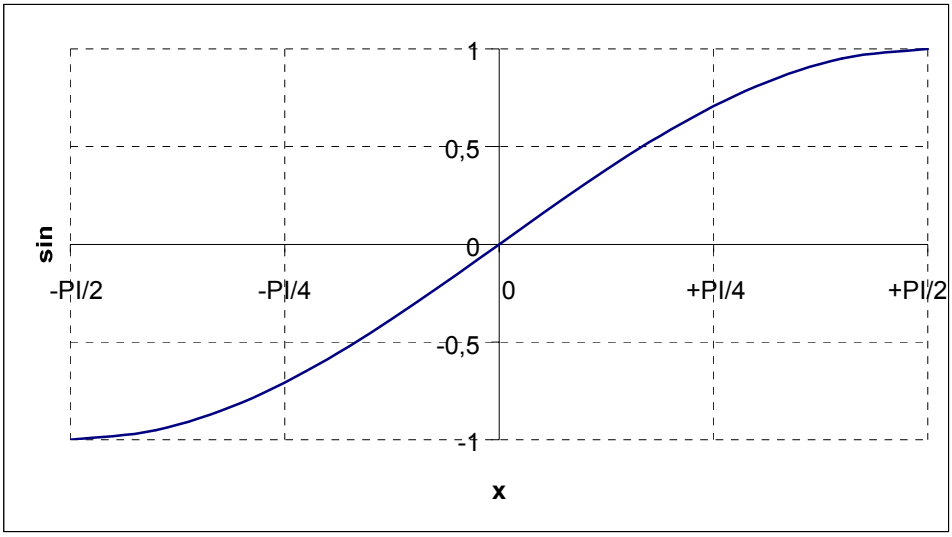
Wertebereich Argument: -1000 bis +1000 (Bogenmaß!)
Wertebereich Ergebnis: -1 bis +1
Mögliche Fehler: keine
Rechenzeit: ca. 5 ms

3.11. Spezialfunktion 23: Tangens

Wertebereich Argument: -1000 bis +1000 (Bogenmaß!)
Wertebereich Ergebnis: -1E13 bis +1E13
Mögliche Fehler: keine
Rechenzeit: ca. 10 ms

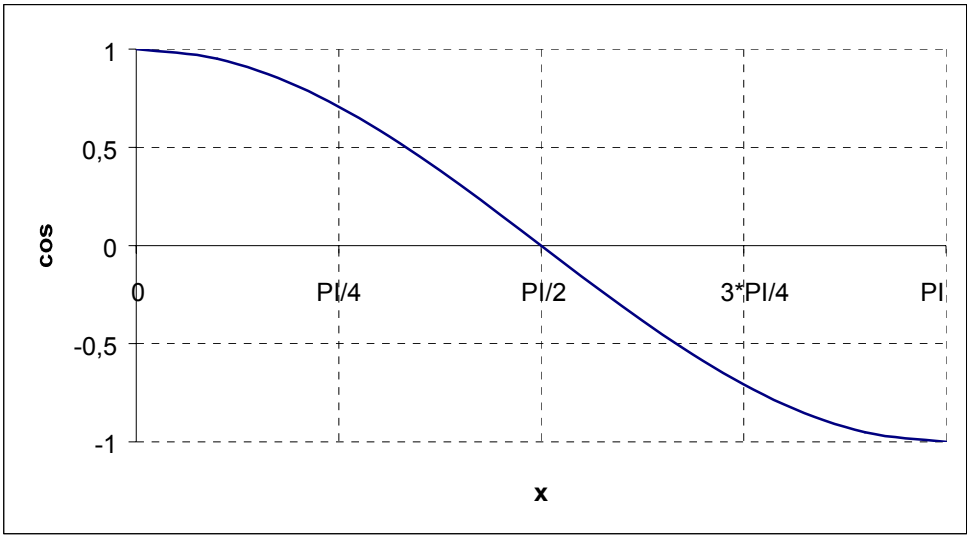
3.12. Spezialfunktion 24: Arcus Sinus

Wertebereich Argument: -1 bis +1
Wertebereich Ergebnis: $-\pi/2$ bis $+\pi/2$
Mögliche Fehler: Argument außerhalb -1 bis +1
Rechenzeit: ca. 6,5 ms



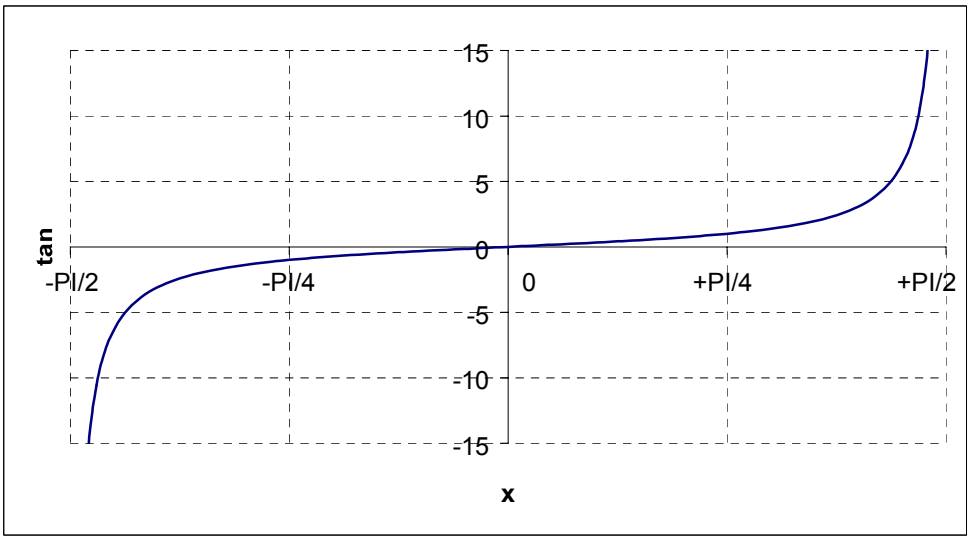
3.13. Spezialfunktion 25: Arcus Cosinus

Wertebereich Argument: -1 bis +1
Wertebereich Ergebnis: 0 bis π
Mögliche Fehler: Argument außerhalb -1 bis +1
Rechenzeit: ca. 6,5 ms



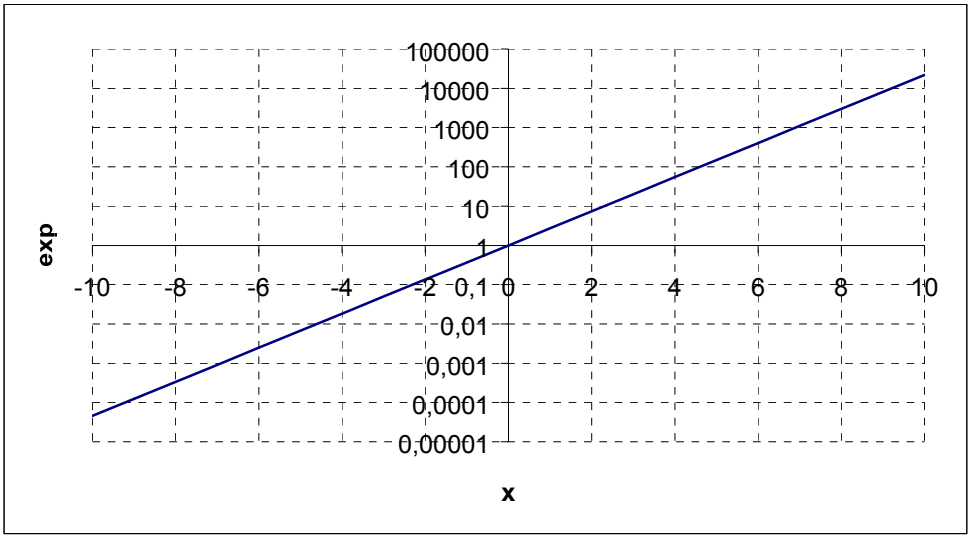
3.14. Spezialfunktion 26: Arcus Tangens

Wertebereich Argument: -1E13 bis +1E13
Wertebereich Ergebnis: -π/2 bis +π/2
Rechenzeit: ca. 5 ms



3.15. Spezialfunktion 27: Exponentialfunktion ("e hoch")

Wertebereich Argument: -30 bis +30
Wertebereich Ergebnis: 1E-13 bis 1E13
Mögliche Fehler: Argument außerhalb -30 bis +30
Ergebnis im Fehlerfall: ABSTURZ!!!!
Rechenzeit: ca. 4,5 ms



3.15.1. Hinweis

Vorsicht bei Verwendung dieser Spezialfunktion. Eine Mißachtung des Wertebereichs des ersten Parameters kann zum Absturz der Steuerung führen.

3.16. Spezialfunktion 28: Natürlicher Logarithmus ("ln")

Wertebereich Argument: 1E-13 bis 1E13
Wertebereich Ergebnis: -30 bis +30
Mögliche Fehler: Argument < 1E-13
Ergebnis im Fehlerfall: ABSTURZ !!!!
Rechenzeit: ca. 2 bis 5 ms

3.16.1. Hinweis

Vorsicht bei Verwendung dieser Spezialfunktion. Eine Mißachtung des Wertebereichs des ersten Parameters kann zum Absturz der Steuerung führen.

3.17. Spezialfunktion 29: Absolutwert

Wertebereich Argument: beliebig
Wertebereich Ergebnis: Betrag des Arguments
Mögliche Fehler: keine
Rechenzeit: ca. 0,5 ms

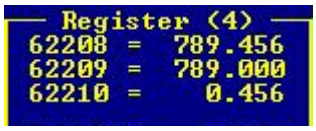
3.18. Spezialfunktion 30: MOD

Wertebereich Argument: beliebig
Wertebereich Ergebnis: Vorkommastellen des Arguments
Wertebereich Ergebnis 2: Nachkommastellen des Argument
Mögliche Fehler: keine

3.18.1. Beispiel (Delta)

```
SPEZIALFUNKTION [#30, p1=62208, p2=62209]
```

Das Registerfenster des Sympas Inbetriebnahmebildschirms sieht nach der Ausführung der Spezialfunktion 30 folgendermaßen aus:



3.18.2. Hinweis

Für die Spezialfunktion 30 können nur Gleitkommaregister, keine Integerregister verwendet werden.

3.19. Spezialfunktion 40: Multimaster Tokenweitergabe

Beide Netzwerkschnittstellen der Delta (Net1 und Net2) sind JetWay- und multimasterfähig; sowohl als Master als auch als Slave.
Es ist jedoch darauf zu achten, daß immer nur eine Schnittstelle Master in einem Netzwerk sein kann!
Master wird eine Schnittstelle dann, wenn die Netzwerkadresse 1 eingestellt ist oder wenn sie im Multimaster-Betrieb den Master-Token zugeteilt bekommt.
Hiermit ist es möglich, die Delta in einen Steuerungsverbund aus PASE-E mit E-INT5-Modulen mit Master-Master-Firmware zu integrieren.

3.19.1. Initialisierung der Net1-Schnittstelle

Die Schnittstelle Net1 kann nicht als freiprogrammierbare, sondern nur nur als Netzwerk-Schnittstelle eingesetzt werden.

Netzwerkadresse:	Register 62995 (Wertebereich: 1 bis 126)
Baudrate:	Register 62996

Wertebereich für Register 62996

0	150
1	300
2	600
3	1200
4	2400
5	4800
6	9600 default
7	19200
8	38400
9	57600
10	76800
11 bis 14	115200

Die beiden Register können beim Einschalten der Steuerung automatisch aus dem EEPROM initialisiert werden:

Netzwerkadresse:	Register 63241 (Wertebereich wie Register 62995)
Baudrate:	Register 63242 (Wertebereich wie Register 62996)

Stehen im EEPROM keine gültigen Werte, so wird die Schnittstelle auf 115200 Baud und Netzwerkadresse 2 gesetzt.

Damit kann die Schnittstelle Net1 als Master oder als Slave in einem Single-Master Netzwerk eingesetzt werden. Der Multimaster-Betrieb wird über den Merker 2152 aktiviert:

Merker 2152 rückgesetzt ➔ Singlemaster-Betrieb
Merker 2152 gesetzt ➔ Multimaster-Betrieb

Nach dem Einschalten der Steuerung ist der Merker 2152 immer gelöscht.

3.19.2. Initialisierung der Net2-Schnittstelle

Nach dem Einschalten der Steuerung ist die Net2-Schnittstelle auf PRIM (freiprogrammierbare Schnittstelle) eingestellt. Zur Verwendung als JetWay-Schnittstelle, sind 3 Register zu beschreiben:

Netzwerkadresse:	Register 62991 (Wertebereich: 1 bis 126)
Baudrate:	Register 62990 (Wertebereich wie Register 62996, siehe Kapitel 3.19.1)
Konfiguration:	Register 62989

Die Net2-Schnittstelle arbeitet als JetWay-Schnittstelle, wenn Bit 0 im Register 62989 gesetzt ist. Die Bits 1 bis 7 in diesem Register werden dann ignoriert.

Der Multimaster-Betrieb wird über den Merker 2153 aktiviert:

Merker 2153 rückgesetzt ➔ Singlemaster-Betrieb
Merker 2153 gesetzt ➔ Multimaster-Betrieb

Nach dem Einschalten der Steuerung ist der Merker 2153 immer gelöscht.

3.19.3. Programmierung

Netzwerk-Kommandos können ausschließlich vom Netzwerk-Master gegeben werden. Tritt bei einem Netzwerk-Zugriff ein Fehler auf, so werden die Merker 2110 (Timeout letzter Netzwerkzugriff) und 2111 (Netzwerk-Timeout seit Reset) gesetzt.

Der Timeout in Millisekunden wird eingestellt über Register 61588 für die Net1- und Register 61511 für die Net2-Schnittstelle. Die Resetwerte betragen jeweils 250 [ms]. Der Timeout sollte bei allen Teilnehmern auf den gleichen Wert eingestellt werden.

3.19.4. Master-Master-Kopplung / Multimaster-Betrieb

Beim Multimaster-Betrieb wechselt die Masterfunktion (Register lesen und schreiben können) zwischen den beteiligten Steuerungen hin und her. Die Vergabe der Netzwerkadresse erfolgt wie beim Singlemaster-Betrieb:

- jede Netzwerkadresse darf nur einmal vergeben werden
- die JetWay-Slaves erhalten eine Netzwerkadresse ungleich 1
- der Master bekommt die Netzwerkadresse 1.

Die Weitergabe des Masters wird durch das Steuerungsprogramm eingeleitet. Dazu wird die Spezialfunktion 40 benutzt:

```
SPEZIALFUNKTION[#40, p1=MasterÜbernahme, p2=MasterErhalt]
```

p1: Netzwerkadresse des Teilnehmers, der den Netzwerk-Master übernehmen soll (indirekte Adressierung möglich)
p2: Netzwerkadresse des Teilnehmers, der den Master tatsächlich bekommen hat (indirekte Adressierung möglich)

Die Übernahme der Masterfunktion durch einen Slave vom Netzwerk-Master erfolgt durch das Betriebssystem, sofern der jeweilige Freigabe-Merker (2152 / 2153) gesetzt und die andere Netzwerkschnittstelle nicht Master in ihrem Netz ist.
Ob eine Schnittstelle gerade Master oder Slave ist, kann in Registern festgestellt werden:

Net1: Register 61587
Net2: Register 61509

Ein Registerinhalt von 0 bedeutet dabei, daß die Schnittstelle als JetWay-Slave arbeitet, ein Wert ungleich 0 (255) bedeutet, daß die Schnittstelle als JetWay-Slave arbeitet.

Sind die Freigabe-Merker nicht gesetzt, so reagiert die Delta nicht auf die Masterübergabe und im Netzwerk-Master wird ein Timeout-Fehler ausgelöst. Ist die Freigabe erteilt, das Anwenderprogramm jedoch gestoppt, so wird vom Betriebssystem der Master sofort an den nächsten Teilnehmer weitergegeben.

Ist eine Delta mit einer Netzwerkadresse ungleich 1 Master geworden, so müssen vom Anwenderprogramm Netzwerkzugriffe durchgeführt werden. Nach Beendigung der Netzwerkaktivitäten muß der Master mit Spezialfunktion 40 weitergegeben werden.
Wird der Master am Netzwerk nicht aktiv, so wird nach einer einstellbaren Überwachungszeit der Teilnehmer mit der Nummer '1' wieder automatisch Master.
Diese Überwachungszeit wird in Register 61512 in Vielfachen des Netzwerk-Timeouts (61588 / 61511) eingestellt. Der Wert nach Reset beträgt 3 (750 ms).

Register 61512 wird auch dafür benutzt, zu definieren, wie oft ein Master maximal versucht, die Masterfunktion an einen anderen Teilnehmer weiterzugeben. Dabei wird bei jedem Versuch die Adresse des Teilnehmers, der den Mastertoken erhalten soll, um eins inkrementiert. Konnte nach dieser Anzahl von Versuchen der Token nicht weitergegeben werden, so stellt sich ein Master mit einer Adresse ungleich 1 automatisch auf Slave um und überläßt dem Teilnehmer mit der Nummer 1 das Netzwerk.

Zur Netzwerkverwaltung ist es noch notwendig, in einem Register die höchste Adresse des im Netzwerk vorkommenden Teilnehmers anzugeben:

Register 61515 höchste Netzwerkadresse

Dieses Register muss in allen beteiligten Steuerungen auf den gleichen Wert gesetzt werden.

Zur optimierten Datenübertragung von Registerblöcken im Multimaster-Betrieb kann auch der Sympas-Befehl `KOPIERE` verwendet werden. Dabei können bis zu 21 Register mit einem Netzwerkbefehl übertragen werden. Für das Quell- oder das Zielregister ist dabei eine 50000er-Registernummer anzugeben, das andere Register muß dabei ein Anwenderregister sein. Werden diese Beschränkungen nicht eingehalten, so werden die Registerwerte einzeln übertragen.

3.20. Spezialfunktion 252: Checksumme ermitteln

```
SPEZIALFUNKTION [#252, p1=Checksumme, p2=Startregister]
```

Mit der Spezialfunktion 252 können Registerwerte zu einer Checksumme aufsummiert werden. Sie wird in das Register eingetragen, das als erster Parameter eingegeben wurde (`Checksumme`).
Die Aufsummierung wird bei dem Register begonnen, das als zweiter Parameter (`Startregister`) angegeben wurde. Sie endet beim letzen Anwenderregister.

PASE-E ➔ Register 8191
Delta ➔ Register 20479