

CAN - CSC595/2

**CAN - SPS Kopplermodul
für
S5-90U, S5-95U und S5-100U**

Hardware - Handbuch

Der Inhalt dieses Handbuches wurde mit größter Sorgfalt erarbeitet und geprüft. esd übernimmt jedoch keine Verantwortung für Schäden, die aus Fehlern in der Dokumentation resultieren könnten. Insbesondere Beschreibungen und technische Daten sind keine zugesicherten Eigenschaften im rechtlichen Sinne.

esd hat das Recht, Änderungen am beschriebenen Produkt oder an der Dokumentation ohne vorherige Ankündigung vorzunehmen, wenn sie aus Gründen der Zuverlässigkeit oder Qualitätssicherung vorgenommen werden oder dem technischen Fortschritt dienen.

Sämtliche Rechte an der Dokumentation liegen bei esd. Die Weitergabe an Dritte und Vervielfältigung jeder Art, auch auszugsweise, sind nur mit schriftlicher Genehmigung durch esd gestattet.

esd electronic system design gmbh
Vahrenwalder Str. 205
30165 Hannover

Tel.: 0511/372 98-0
FAX : 0511/372 98-68
E-Mail: info@esd.electronics.com
Internet: www.esd-electronics.com

Dokument-Datei:	I:\TEXTE\DOKU\MANUALS\CAN\CSC595.2\DEUTSCH\CSC0515H.MA6
Datum der Druckvorlagenerstellung:	20.12.99

Platinenversion:	CSC505-2
-------------------------	----------

Änderungen in den Kapiteln

Die hier aufgeführten Änderungen im Anwenderhandbuch betreffen sowohl Änderungen in der Hardware als auch reine Änderungen in der Beschreibung der Sachverhalte.

Kapitel	Änderungen gegenüber Rev. 1.4
5.3	Steckerbelegung für DeviceNet-Option
4.4.2	LED-Anzeige für DeviceNet-Software

Weitere technische Änderungen vorbehalten.

Inhalt	Seite
1. Übersicht	3
1.1 Beschreibung des Moduls	3
1.2 Frontplattenansicht mit LEDs- und Steckern	5
1.3 Zusammenfassung der technischen Daten	6
1.3.1 Allgemeine technische Daten	6
1.3.2 CAN-Schnittstellen der CSC595/2	7
1.3.3 SPS-Baugruppe	8
1.4 Software-Unterstützung	8
1.5 Bestellhinweise	9
2. Installationshinweise	11
2.1 Einbau des CSC595/2-Moduls in ein SIMATIC-Automatisierungsgerät	11
2.2 Anschluß einer Bedienstation	12
2.2.1 Terminal	12
2.2.2 PC oder Laptop mit Terminalprogramm	12
3. Bestückungsdruck, Steckbrücken und Kodierschalter	13
3.1 Bestückungsdruck	13
3.2 Default-Einstellung der Brücken und Kodierschalter	15
3.3 Beschreibung der Brücken und Kodierschalter	16
3.3.1 Betrieb des 82527 mit 10 MHz oder 20 MHz (S202)	16
3.3.2 Tx-Signal des CAN-Controllers 82527 an CAN-Interface (S203)	16
3.3.3 Aktivierung des Bootstrap Loaders (SL3)	17
3.3.4 Einstellung der Module-No. über Kodierschalter S301	18
4. Beschreibung der Baugruppen	19
4.1 SPS-Bus-Interface	19
4.2 CAN-Bus-Interface	20
4.2.1 Bitrate	20
4.2.2 Sende-und Empfangsschaltung des CAN-Interface (Physical Layer)	20
4.3 Spezifikation der seriellen Schnittstelle	21
4.4 LED-Anzeige	23
4.4.1 LED-Anzeige für CAN-Schicht-2- und CANopen-Software	23
4.4.2 LED-Anzeige für DeviceNet-Software	24
5. Steckerbelegungen	25
5.1 SPS-Bus-Stecker P301	25
5.2 Stecker der CAN-Bus-Schnittstelle P3 (9-pol. DSUB Stift)	26
5.3 DeviceNet-Option	27
5.4 Serielle Schnittstelle RS-232 auf P2 (9-pol. DSUB Buchse)	28
5.5 Anschlußleitungen für die RS-232-Schnittstelle zum PC	29
6. Korrekte Verdrahtung galvanisch getrennter CAN-Netze	31
7. Stromlaufpläne	35



1. Übersicht

1.1 Beschreibung des Moduls

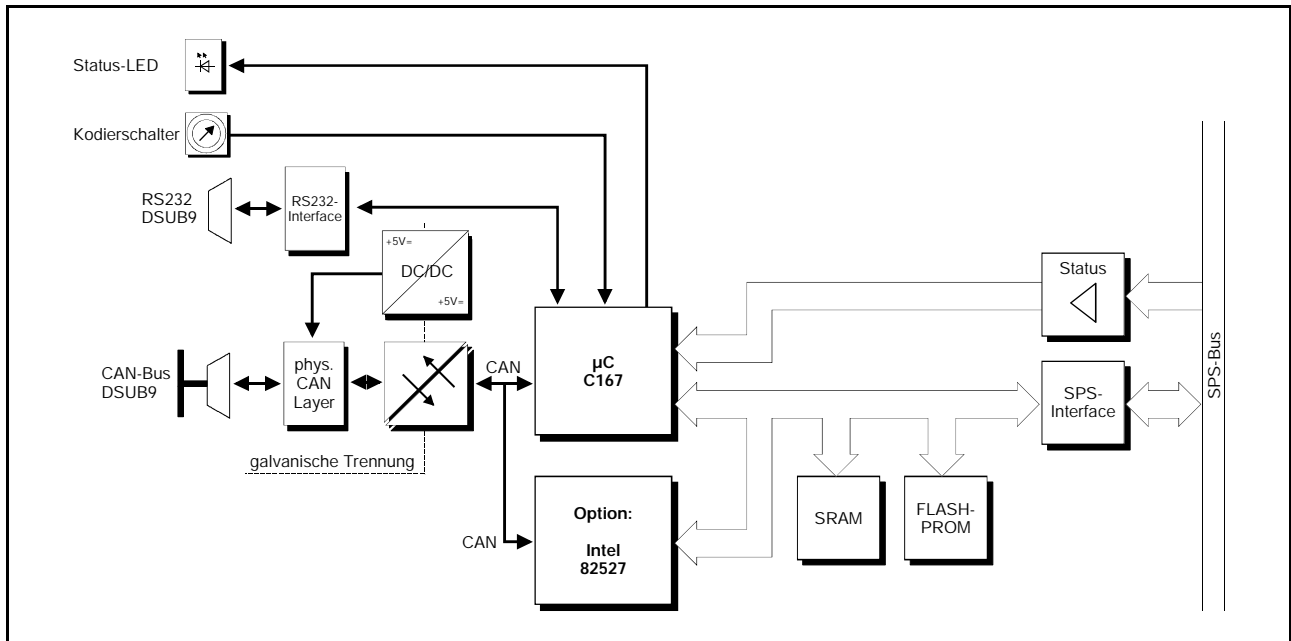


Abb. 1.1.1: Blockschaltbild der CSC595/2

Mit dem Kommunikationsprozessor CAN-CSC595/2 ist eine direkte Kopplung zwischen SIEMENS-SPS der Typen S5-90U, S5-95U oder S5-100U und esd-CAN-I/O-Modulen oder anderen CAN-Teilnehmern möglich. Dadurch ergibt sich unter anderem die Möglichkeit einer schnellen Prozeßdatenkopplung zu VMEbus- oder PC-basierten Systemen oder zu anderen SPS-Einheiten.

Das Modul gewährleistet für den SPS-Programmierer die völlige Transparenz der Prozeßdaten. Es sind keine zusätzlichen Funktions- oder Datenbausteine notwendig, so daß SPS-Programme in gewohnter Weise ablauffähig sind.

Die CAN-CSC595/2 arbeitet mit dem Hochleistungs-Microcontroller C167C mit integriertem CAN-Controller und garantiert eine Bitrate von 1 MBit/s ohne Datenverlust, sogar wenn der C167C als High-Level-Protokoll-Master arbeitet.

Der physikalische CAN-Layer entspricht ISO 11898. Die Baudrate kann, ebenso wie alle CAN-Identifizier, über die lokale RS-232 Schnittstelle mit Hilfe der Software eingestellt werden.



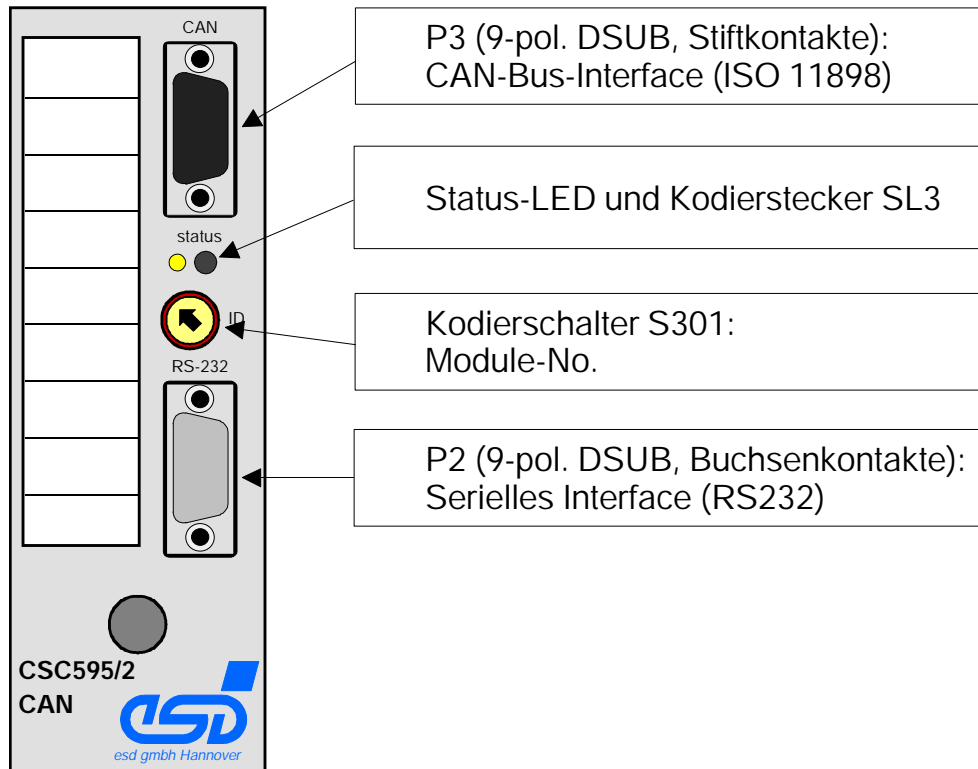
Die Konfiguration - auch anderer Module - kann über die RS-232-Schnittstelle erfolgen. Eine automatische Konfiguration anderer Module (nach dem Kaltstart) ist ebenfalls möglich. Die Einstellungen werden im lokalen EEPROM gespeichert.

Das Modul wird in einem Kunststoff-Gehäuse geliefert, das zu SIEMENS S5-Geräten kompatibel ist.



Übersicht

1.2 Frontplattenansicht mit LEDs- und Steckern





1.3 Zusammenfassung der technischen Daten

1.3.1 Allgemeine technische Daten

RS-232-Schnittstelle	RS-232C-Interface auf 9-poliger DSUB-Buchse als Eingabe- und Konfigurationsschnittstelle und zum Laden neuer S-Records für Software-Updates des FLASH-EPROMs
Temperaturbereich	0...50EC Umgebungstemperatur
Luftfeuchtigkeit	max. 90%, nicht kondensierend
Versorgungsspannung CAN-Modul	über SPS-Bus zugeführt, Nennspannung 9V \pm 10%, Stromaufnahme (typisch, bei 20EC): ca. 200 mA (ohne CAN-Controller 82527)
Steckverbinder	P301 - SPS-Anschluß P2 (DSUB9/Buchse) - RS-232-Schnittstelle P3 (DSUB9/Stift) - CAN-Bus-Schnittstelle
Gehäuse	Siemens SPS-Modul-Gehäuse, passend zu SIMATIC S5 Bus-Modul
Gewicht	ca. 250 g

Tabelle 1.3.1: Allgemeine Daten der CSC595/2



1.3.2 CAN-Schnittstellen der CSC595/2

Anzahl der CAN-Interfaces	eine Schnittstelle an Stecker P3
Controller-Bausteine	C167 und 82527(Optional)
Nutzung des optionalen zweiten Controllers	Empfang und Auswertung von RTR-Frames
CAN-Identifizier	über CAN- oder RS-232-Schnittstelle programmierbar
esd-Module-No.	über Kodierschalter in der Frontplatte einzustellen oder über CAN- oder RS-232-Schnittstelle programmierbar
I ² C-EEPROM	zur Speicherung der Parameter
Physical Layer	Physical Layer gemäß ISO 11898, Übertragungsrate programmierbar von 10 kBit/s bis 1 MBit/s
Galvanische Trennung des CAN-Interfaces gegenüber anderen Baugruppen	Trennung über Optokoppler und DC/DC-Wandler, nach VDE 0110b §8, Isolationsgruppe C und Einbau in Schaltschrank): 300 V(DC), 250 V(AC)
DeviceNet-Option	je ein Adapter-Board für jeden CAN-Kanal mit Phoenix Combicon-Steckverbinder (oder äquivalente), Optocoppler und CAN-Treiber gemäß DeviceNet-Spezifikation 'DeviceNet Communication Model and Protocol, Rel. 2.0', DeviceNet-Stecker über Frontplatte zugänglich.

Tabelle 1.3.2: CAN-Schnittstellen der CSC595/2



1.3.3 SPS-Baugruppe

SPS-Ankopplung	kompatible SPS-Baugruppen: SIEMENS S5-90U SIEMENS S5-95U SIEMENS S5-100U SIEMENS S5-102U SIEMENS S5-103U SIEMENS ET-100
Monitoring-Mode	Überwachung des SPS-Bus: Übertragung aller SPS-Daten auf den CAN-Bus

Tabelle 1.3.3: SPS-Baugruppe

1.4 Software-Unterstützung

Die komplette EPROM-residente CAN-Kommunikations-Firmware zum Betrieb des CSC595/2-Moduls ist im Lieferumfang enthalten. Die Software wird im zweiten Teil des Handbuches beschrieben.

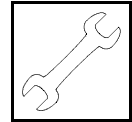


1.5 Bestellhinweise

Typ	Eigenschaften	Bestell-Nr.
CAN-CSC595-2	Koppler CAN/Siemens S5-SPS Lieferumfang: Gerät mit CAN-Controller C167 (82527 nicht bestückt) mit Kunststoffgehäuse, Kodierstift für Bootstrap-Loader, Soft- und Hardware-Handbuch	C.2902.02
CAN-CSC595/2-SDS	Option: SDS Masterfirmware	C.2902.50
CAN-CSC595/2-CoS	Option: CANopen Slavefirmware	C.2902.52
CAN-CSC595/2-CoM	Option: CANopen Masterfirmware	C.2902.54
CAN-CSC595/2-MD	Anwenderhandbuch in deutsch 1*)	C.2902.20
CAN-CSC595/2-ME	Anwenderhandbuch in englisch 1*)	C.2902.21

1*) Wird das Handbuch gemeinsam mit dem Modul bestellt, so wird es kostenlos mitgeliefert.

Tabelle 1.5.1: Bestellhinweise



2. Installationshinweise

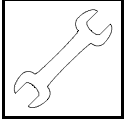
2.1 Einbau des CSC595/2-Moduls in ein SIMATIC-Automatisierungsgerät

Für die Inbetriebnahme des SIMATIC-S5-Automatisierungsgerätes sind die Anweisungen des SIMATIC-S5-Handbuches zu beachten! Die folgenden Schritte beziehen sich nur auf die Installation des CSC595/2-Moduls.

Vorgehensweise:

1. Stromversorgung der SIMATIC Zentral-Erweiterungsgeräte und der Signalgeber und Signalempfänger unterbrechen (ausschalten).
2. Freien Steckplatz im Zentralgerät aussuchen, CSC595/2 auf Baugruppenträger der SIMATIC stecken und mit der über die Frontplatte zugänglichen Schlitzschraube fixieren.
3. CAN-Schnittstelle anschließen.
Die CAN-Schnittstelle wird über den 9-poligen DSUB-Stecker in der Frontplatte angeschlossen. Hinweise zur Verdrahtung des CAN-Netzes können dem Kapitel 'Korrekte Verdrahtung galvanisch getrennter Netze' am Ende dieses Handbuches entnommen werden.
4. Terminal an RS-232-Schnittstelle anschließen.
Es kann ein handelsübliches Terminal (z.B. WYSE, FALCO) oder ein PC oder Laptop mit einem Terminal-Programm verwendet werden. Der Anschluß wird in dem folgenden Kapitel 'Anschluß einer Bedienstation' separat beschrieben.
5. Zentralgerät einschalten, die anderen CAN-Bus-Teilnehmer einschalten, Terminal einschalten (die Reihenfolge ist beliebig)
6. Befindet sich die Treiber-Software bereits im lokalen Flash-EEPROM (Default-Zustand bei Auslieferung des Moduls), dann sollte die Status-LED des Controllers C167 (neben dem Kodierstecker) blinken: 500 ms lang grün und 100 ms lang rot. Sie zeigt damit an, daß der Zustand des Moduls OK ist und daß es mit den Default-Parametern arbeitet.
7. Das CSC595/2-Modul kann jetzt über das Terminal konfiguriert werden. Während der Konfiguration können verschiedene Parameter (z.B. Baudrate, Identifier) verändert werden. Alle Konfigurationsparameter können im lokalen EEPROM abgespeichert werden. Die geänderten und abgespeicherten Parameter werden erst nach einem RESET aktiv.

Die Konfiguration des Moduls wird im Software-Handbuch beschrieben. Dort findet sich auch eine vollständige Auflistung der Default-Parameter, mit denen das Modul nach der Auslieferung arbeitet.



Installation

2.2 Anschluß einer Bedienstation

Die Bedienstation wird zur Konfiguration des CSC595/2-Moduls benötigt. Es kann ein handelsübliches Terminal oder ein PC mit Terminal-Programm verwendet werden. Sollen neue Software-Updates vom Anwender selbst installiert werden, ist ein PC zwingend notwendig.

Die Einstellungsparameter der Schnittstelle (Baudrate etc.) sind in dem noch folgenden Kapitel 'Spezifikation der seriellen Schnittstelle' auf Seite 21 aufgeführt.

2.2.1 Terminal

Während der Verdrahtung sollte das Terminal ausgeschaltet sein. Das Terminal wird über die 9-polige DSUB-Buchse (P2) in der Frontplatte angeschlossen. Die Signalbelegung ist so gewählt, daß ein Terminal ohne Nullmodem direkt angeschlossen werden kann.

2.2.2 PC oder Laptop mit Terminalprogramm

Während der Verdrahtung sollte der PC, bzw. das Laptop ausgeschaltet sein.

An welchem Port das Modul während des Betriebs (Konfiguration) angeschlossen wird, hängt vom eingesetzten Terminal-Programm ab. In der Regel werden auch hier mehrere Ports unterstützt.

Beim Anschluß an ein 9-poliges Maus-Port ist ein *Nullmodem* in die Zuleitung zu schalten. Erfolgt der Abgriff über einen 25-poligen DSUB-Stecker, ist kein Nullmodem erforderlich. Die Signalbelegung geeigneter Anschlußleitungen ist im Anhang aufgeführt.

esd bietet auch fertig konfektionierte Anschlußkabel mit 9-poligen Steckern an, bei denen kein Nullmodem erforderlich ist.



3. Bestückungsdruck, Steckbrücken und Kodierschalter

3.1 Bestückungsdruck

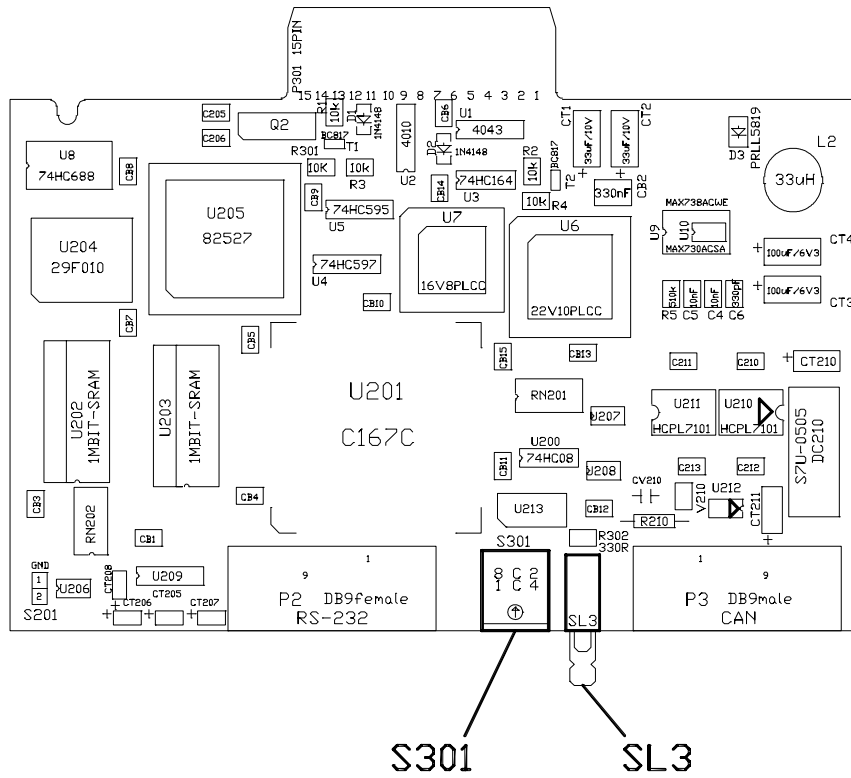


Abb. 3.1.1: Lage der Konfigurationselemente auf der Bestückungsseite der Platine

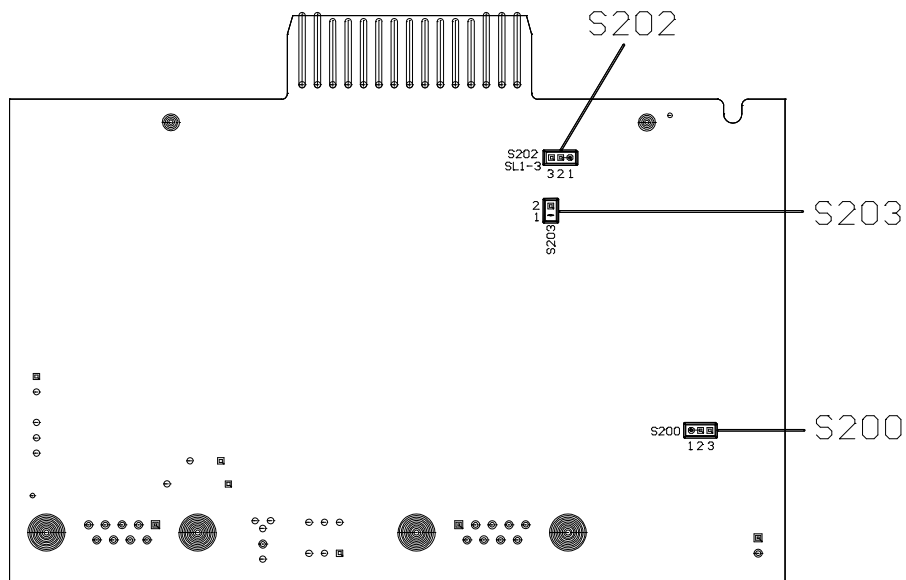
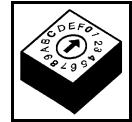


Abb. 3.1.2: Lage der Lötbrücken auf der Lötseite der Platine



3.2 Default-Einstellung der Brücken und Kodierschalter

Die jeweilige Default-Einstellung der Brücken, Kodierschalter und des Steckkontaktes bei Auslieferung der Karte ist in den nachfolgenden Abbildungen eingetragen.

Die Anordnung der Bauteile auf der Bestückungsseite ist Abbildung 3.1.1 zu entnehmen. Die Bauteile sind in den anschließenden Beschreibungen so dargestellt, wie sie der Anwender betrachtet, wenn er die Platine mit den CAN-Bus-Steckern nach links vor sich liegen hat.

Die Lage der Lötbrücken auf der Lötseite ist Abbildung 3.1.2 zu entnehmen. Die Lötbrücken sind in den anschließenden Beschreibungen so dargestellt, wie sie der Anwender betrachtet, wenn er die Platine mit den CAN-Bus-Steckern nach rechts vor sich liegen hat (Ansicht Lötseite).

Zusammenfassung der Default-Einstellungen bei Auslieferung des Moduls:

Lötbrücke	Funktion	Einstellung
S200	Speicherkapazität der SRAMs	256 kByte (2 x 128 kByte)
S202	Betrieb des 82527 mit 10 MHz oder 20 MHz	82527 wird mit 20 MHz getaktet
S203	Tx-Signal des 82527 an CAN-Interface	Board ohne 82527: offen Board mit 82527: gesetzt, d.h. Tx-Signal ist mit CAN-Bus-Interface verbunden

Hinweis: Die Lötbrücke S200 wird im folgenden nicht weiter beschrieben, da die Stellung der Brücke von der Bestückung der SMD-Speicherbausteine (SRAM) abhängt. Die Bestückung der SMD-Speicher wird jedoch einmalig im Werk festgelegt und danach nicht mehr verändert. Der Anwender darf die Lötbrücke S200 daher nicht umsetzen!

Steckkontakt/ Kodierschalter	Funktion	Einstellung
Steckkontakt SL3	Bootstrap-Loader aktivieren	nicht gesteckt, d.h. Bootstrap-Loader nicht aktiv
Kodierschalter S301	Module-No.	die Module-No. muß in jedem Fall vom Anwender an vorhandenes CAN-Netz angepaßt werden, daher keine definierte Default-Stellung

Tabelle 3.2.1: Default-Einstellung der Brücken und Kodierschalter

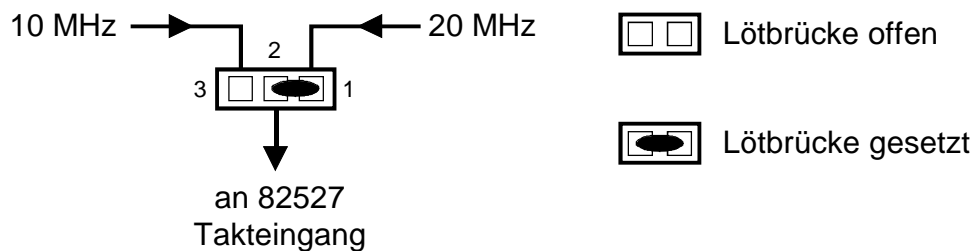


3.3 Beschreibung der Brücken und Kodierschalter

3.3.1 Betrieb des 82527 mit 10 MHz oder 20 MHz (S202)

Über diese Lötbrücke kann die Taktfrequenz des CAN-Controllers 82527 auf 10 MHz oder 20 MHz eingestellt werden. Bei Auslieferung ist die Brücke gesetzt, d.h. der Controller wird mit 20 MHz betrieben.

Die Stellung dieser Lötbrücke sollte vom Anwender nicht verändert werden.

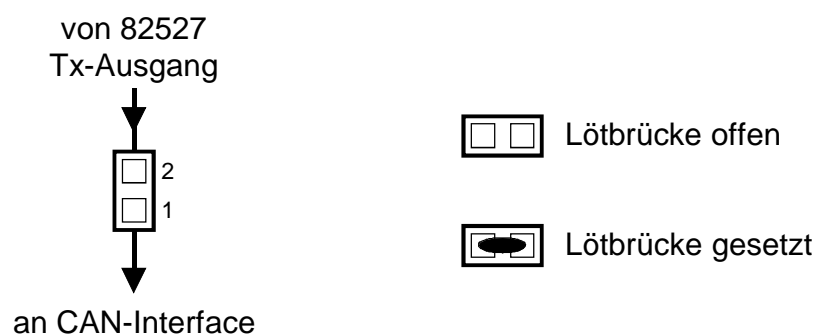


Dargestelltes Beispiel: Einstellung der Lötbrücke für 20 MHz-Betrieb des Controllers 82527

3.3.2 Tx-Signal des CAN-Controllers 82527 an CAN-Interface (S203)

Über diese Lötbrücke wird das Tx-Signal des CAN-Controllers 82527 mit dem CAN-Interface verbunden. Ist der Controller nicht bestückt, so ist die Lötbrücke offen.

Die Stellung dieser Lötbrücke sollte vom Anwender nicht verändert werden.



Dargestelltes Beispiel: CAN-Controller 82527 nicht bestückt



3.3.3 Aktivierung des Bootstrap Loaders (SL3)

Um ein Software-Update über die serielle Schnittstelle in den lokalen Speicher herunterladen zu können, muß der Bootstrap-Loader freigeschaltet werden. Die Verriegelung soll dem Schutz vor versehentlichem Überschreiben des lokalen Programmcodes dienen.

Um den Bootstrap-Loader freizuschalten, ist der mitgelieferte Kodierstift in die Buchse SL3 zu stecken. Der Stift schließt einen internen Kontakt und schaltet dadurch den Bootstrap-Loader frei. Der Ladevorgang kann jetzt über die Bedien-Software gestartet werden.

Der Kodierstift schließt den Kontakt korrekt, wenn er bis zum Anschlag hineingesteckt wird.



3.3.4 Einstellung der Module-No. über Kodierschalter S301

Die Module-No., mit der das CSC595/2-Modul beim Betrieb mit den Default-Parametern über den CAN-Bus selektiert wird, ist 8 Bit breit. Sie dient der Firmware zur Identifikation des Moduls.

Mit dem vierpoligen Kodierschalter S301 in der Frontplatte werden die Bits 0 bis 3 der CAN-Module-No. gesetzt. Die Bits 4 bis 7 der Module-No. sind fest auf '0' gelegt.

Die Zuordnung der Kodierschalter-Stellung zur Module-No. ist somit wie folgt:

Kodierschalter-Stellung	Module-No.-Bit [HEX]
0 (*)	00 (*)
1	01
2	02
:	:
E	0E
F	0F

Tabelle 3.3.4: Zuordnung der Kodierschalterstellung zur Module-No.

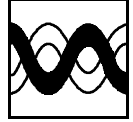
(*) Wird der Kodierschalter auf '00' gestellt und ein RESET ausgelöst (über Power Down), so arbeitet das Modul nach dem Wiedereinschalten mit den Default-Parametern weiter. Alle vormals veränderten Parameter, auch wenn sie im lokalen I²C-EEPROM gespeichert wurden, gehen verloren.

Arbeitet das Modul mit den Default-Parametern, so ist die an dem Kodierschalter eingestellte Module-No. aktiv. Über die Firmware kann die komplette 8 Bit breite Module-No. frei programmiert werden. Die programmierte Module-No. ersetzt sofort die über den Kodierschalter eingestellte Module-No..

Die Programmierung der Module-No. wird im Software-Handbuch des Moduls beschrieben.

Achtung: Das Modul kann nicht parametrisiert werden, solange der Kodierschalter auf '0' eingestellt ist! Nach dem Reset, der notwendigerweise der Programmierung folgen muß, würden die geänderten Parameter sofort wieder mit den Default-Werten überschrieben werden.

Die Stellung des Kodierschalters bei Auslieferung des Moduls ist nicht festgelegt, da sie generell vom Anwender mit den anderen Modul-Nummern im CAN-Netz abgeglichen werden muß.



4. Beschreibung der Baugruppen

4.1 SPS-Bus-Interface

Das CSC595/2-Modul ist mit einem SPS-Interface ausgestattet, das für den Anschluß an Siemens SIMATIC-S5-Baugruppen vorgesehen ist. Die Steuerung des Interface wird von programmierbaren Logikbausteinen übernommen. Die Sende- und Empfangsdaten werden in SRAM-Speichern zwischengespeichert. Die folgende Abbildung zeigt den Aufbau der Interface-Steuerung.

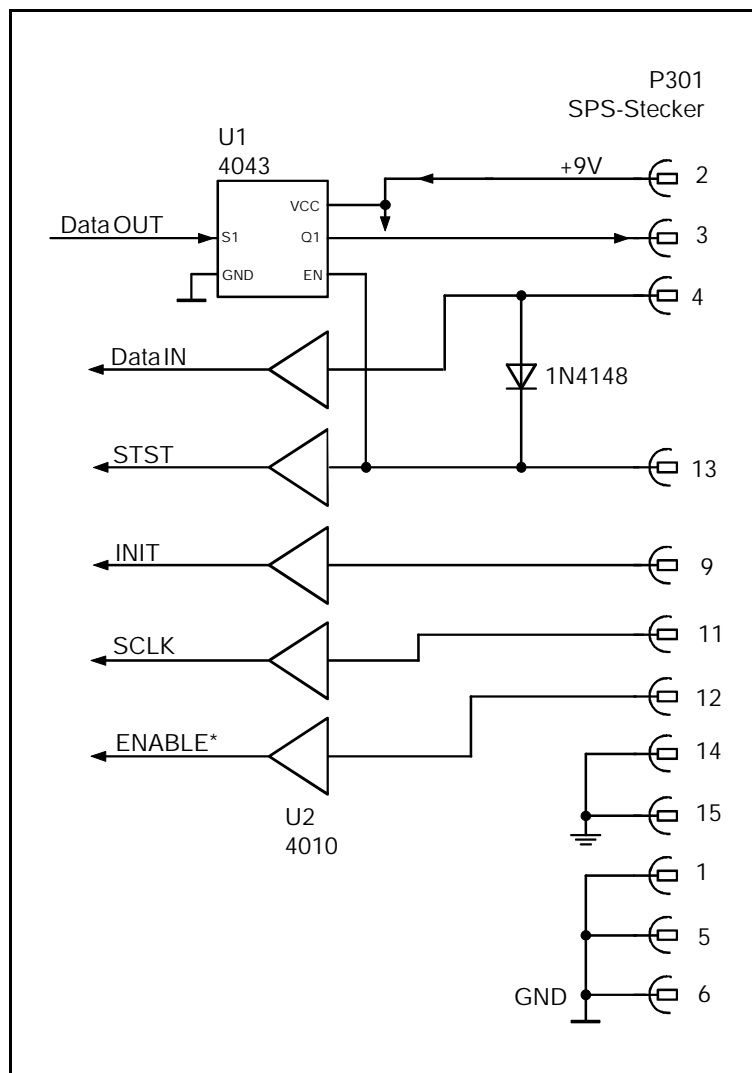
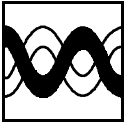


Abb. 4.2.1: Blockschaltbild der SPS-Interface-Steuerung



4.2 CAN-Bus-Interface

4.2.1 Bitrate

Die Übertragungsgeschwindigkeit des CAN-Interface kann von 10 KBit/s bis 1,0 MBit/s variiert werden. Die Einstellung der Bitrate erfolgt über die lokale Software. Weitere Informationen hierzu finden sich im Software-Handbuch des Moduls.

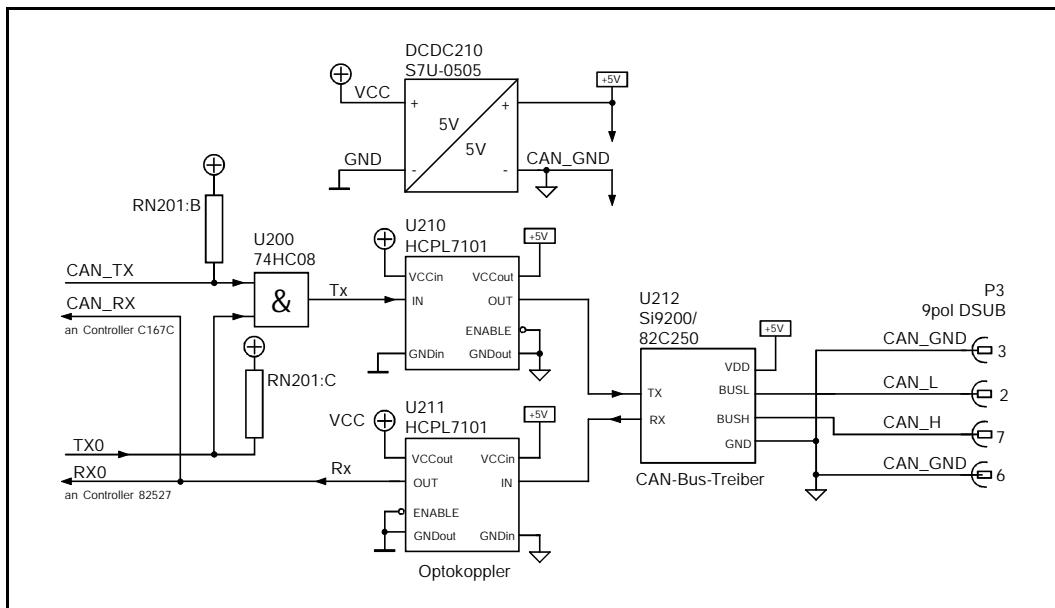
4.2.2 Sende-und Empfangsschaltung des CAN-Interface (Physical Layer)

Als CAN-Controller wird der C167C eingesetzt. Das physikalische Interface des CAN-Bus entspricht der Norm ISO 11898. Als CAN-Bus-Transceiver werden auf dem Modul z.B. der Si9200 oder der 82C250 eingesetzt.

Die Spannungsversorgung des CAN-Interface erfolgt über einen DC/DC-Wandler aus der lokalen +5V-Versorgungsspannung. Die galvanische Trennung der Signale zum CAN-Bus erfolgt über Optokoppler.

Anmerkung zur Verdrahtung des CAN-Netzes:

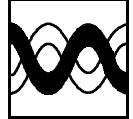
Das Bezugspotential des CAN-Bus (CAN_GND) muß an genau *einem* Punkt des CAN-Netzes mit dem Erdpotential verbunden werden.



+5V, VCC, GND, CANGND...	lokale Versorgungsspannungen
CAN_TX, CAN_RX...	Signale des CAN-Controllers C167C (U201)
TX0, RX0...	Signale des optionalen CAN-Controllers 82527 (U205)
CAN_L, CAN_H...	CAN-Bus-Signalleitungen

Abb. 4.3.1: Funktionsschaltbild des CAN-Bus-Interface beim Einsatz der Interface-Bausteine Si9200 oder 82C250

Die Beschreibung der Steckerbelegung befindet sich im Anhang.



4.3 Spezifikation der seriellen Schnittstelle

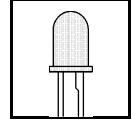
Die RS-232-Schnittstelle ist wie folgt spezifiziert:

Parameter	Einstellungen
Baudrate	19200 Baud
Daten-Bits	8 Bits/Character
Stop-Bits	1 Stop-Bit
Parity-Bit	No Parity
Handshake	No Handshake (oder, falls dies nicht einstellbar: XON/XOFF)

Tabelle 4.4.1: Parameter der seriellen Schnittstelle

Das angeschlossene Terminal, bzw. der PC mit Terminal-Programm ist auf die oben angegebenen Werte einzustellen.

Die Steckerbelegung der Schnittstelle auf P2 ist im Anhang aufgeführt.



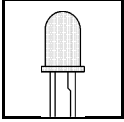
4.4 LED-Anzeige

4.4.1 LED-Anzeige für CAN-Schicht-2- und CANopen-Software

Die Status-LED befindet sich neben dem Bootstrap-Stecker.

Leuchtzustand	Bedeutung des Leuchtzustandes		
	Modul ist in RESET-Zustand	Modul ist im Bootstrap-Modus	Modul ist in 'normalem' Betrieb
konstant grün leuchtend	-	-	Allgemeiner Zustand des Moduls ist Ok
LED blinkt: kurz grün, kurz rot (etwa 100 ms grün, 100 ms rot)	-	-	EEPROM-Error
LED blinkt: kurz grün, kurz rot (etwa 200 ms grün, 200 ms rot)	-	Anzeige des Bootstrap-Modes	-
LED blinkt: lang grün, kurz rot (etwa 500 ms grün, 100 ms rot)	-	-	Default-Status (Modul arbeitet mit Default-Parametern)
LED blinkt: kurz grün, kurz rot, kurz aus (etwa 330 ms grün, 330 ms rot, 330 ms aus)	-	-	Modul ist als CANopen- Master konfiguriert, das CANopen-Netz ist aber noch nicht gestartet
LED aus	Modul ist im RESET	-	-

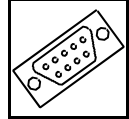
-... Diese Kombination von Betriebszustand des Moduls und Leuchtzustand der LED tritt nicht ein.



LED-Anzeigen

4.4.2 LED-Anzeige für DeviceNet-Software

Leuchtzustand	Bedeutung des Leuchtzustandes
LED aus	Prüfung auf Vergabe eines doppelten Identifiers; 'duplicate MAC-ID request'
LED leuchtet 1x kurz grün beim Einschalten und ist dann aus	Nicht an DeviceNet angeschlossen
LED blinkt: grün	System hochgelaufen und 'duplicate MAC-ID request' durchgeführt, aber die Kommunikation der Teilnehmer besteht noch nicht.
konstant grün leuchtend	Die Kommunikation ist aufgebaut und arbeitet fehlerfrei.
LED blinkt: rot	Unterbrechung in der Kommunikation zwischen einem DeviceNet- Teilnehmer und CSC595/2.
konstant rot leuchtend	'fatal error', unbehebbarer Fehler aufgetreten! Bitte überprüfen Sie die Einstellungen und Schaltungen der angeschlossenen DeviceNet-Geräte.
LED blinkt: 50 ms rot 50 ms grün	'CAN-error' aufgetreten.

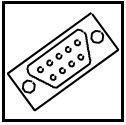


5. Steckerbelegungen

5.1 SPS-Bus-Stecker P301

Pin	Signal
1	GND
2	+9V
3	Data OUT
4	Data IN
5	GND
6	GND
7	-
8	-
9	INIT
10	-
11	SCLK
12	ENABLE*
13	STST-Bus
14	PE
15	PE

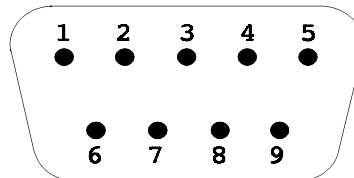
+9V, GND.....	Spannungsversorgung
Data IN, Data OUT.....	Datenleitungen
STST-Bus.....	Start/Stop-Signal von Steuereinheit
INIT.....	Signal für Initialisierungssequenz
SCLK.....	synchroner Shift-Takt
ENABLE*.....	Freigabeeingang für Bus
PE.....	Schutzerde-Anschluß
'-'.....	nicht angeschlossen



Steckerbelegung

5.2 Stecker der CAN-Bus-Schnittstelle P3 (9-pol. DSUB Stift)

Pin-Zuordnung:



Pin-Belegung:

Signal	Pin		Signal
CAN_GND	6	1	reserviert
		2	CAN_L
CAN_H	7	3	CAN_GND
reserviert	8	4	reserviert
reserviert	9	5	reserviert

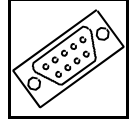
9-poliger DSUB-Stecker

Signalbeschreibung:

CAN_L, CAN_H ... CAN-Signalleitungen

CAN_GND ... Bezugspotential des lokalen CAN-Physical Layers

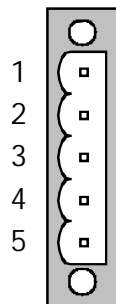
reserviert ... Pins, die für zukünftige Anwendungen reserviert sind.



5.3 DeviceNet-Option

Ist das Modul mit dem DeviceNet-Interface bestückt, stehen die DSUB-Stecker nicht zur Verfügung. Als Steckverbinder werden 5-polige Phoenix-Combicon-Stecker MSTB 2.5/-GF-5.08 (oder äquivalente) eingesetzt.

Pin-Zuordnung:

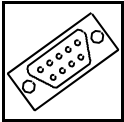


Pin-Belegung:

Pin	Signal
1	V-
2	CAN-
3	Shield
4	CAN+
5	V+

Signalbeschreibung:

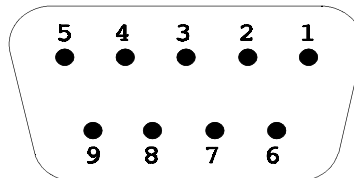
V+...	Spannungszuführung für das CAN-Interface ($U_{VCC} = 24 \text{ V} \pm 4\%$)
V-...	Bezugspotential für V+ und CAN+/CAN-
CAN+, CAN-...	CAN-Signale
Shield...	Abschirmung (über hochohmige RC-Kombination (1MS, 10nF/500V) mit Erde (Frontplatte) verbunden)



Steckerbelegung

5.4 Serielle Schnittstelle RS-232 auf P2 (9-pol. DSUB Buchse)

Pin-Zuordnung:



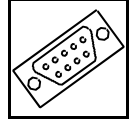
Pin-Belegung:

Signal	Pin		Signal
-	1	6 7 8 9	-
RxD	2		-
TxD	3		-
DTR	4		-
GND	5		-

9-polige DSUB-Buchse

Signalbeschreibung:

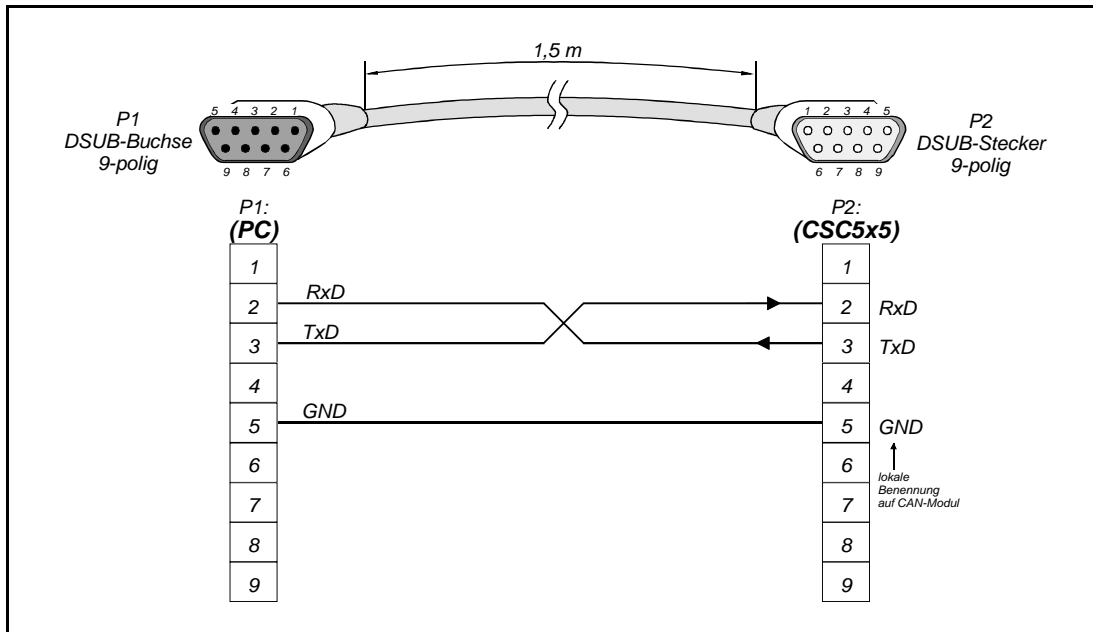
- TxD ... Signalleitung: Datenausgang der CSC595/2
- RxD ... Signalleitungen: Dateneingang der CSC595/2
- DTR ... Handshake-Signal (Ausgang)
- GND ... Bezugspotential
- '-' ... nicht angeschlossen



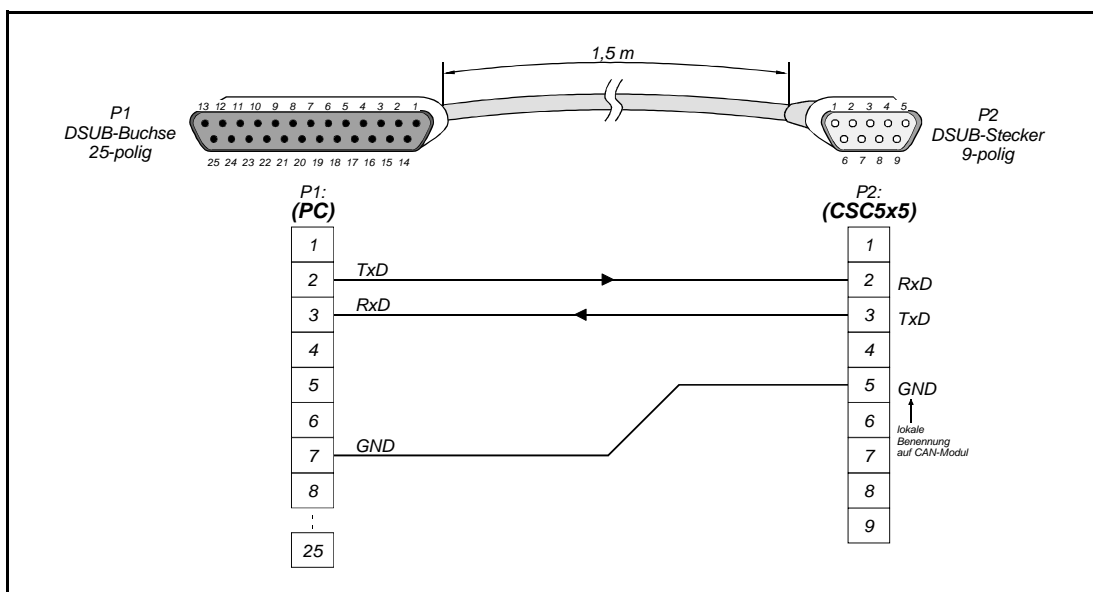
5.5 Anschlußleitungen für die RS-232-Schnittstelle zum PC

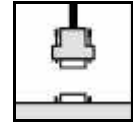
Die folgenden beiden Abbildungen zeigen die notwendige Belegung der RS-232-Anschlußleitungen zwischen CSC595/2 und einem PC.

Adapterkabel 9-pol. DSUB Buchse auf 9-pol. DSUB Stecker



Adapterkabel 25-pol. DSUB-Buchse auf 9-pol. DSUB-Stecker





6. Korrekte Verdrahtung galvanisch getrennter CAN-Netze

Generell sind bei der Verdrahtung sämtliche gültigen Richtlinien (DIN, VDE) bzgl. EMV-gerechtem Aufbau, Leitungsführung, Leiterquerschnitte, zu verwendende Materialien, Mindestabstände, Blitzschutz etc. zu beachten.

Die folgenden **Grundregeln** für die CAN-Bus Verdrahtung sollten unbedingt beachtet werden:

1.	Ein CAN-Netz darf sich nicht verzweigen (Ausnahme: kurze Stichleitungen) und muß an beiden Enden mit dem Wellenwiderstand der Leitung (in der Regel $120\ \Omega \pm 10\%$) abgeschlossen werden (zwischen den Signalen CAN_L und CAN_H und nicht gegen GND)!
2.	Eine CAN-Datenleitung benötigt zwei verdrehte Adern (Twisted Pair) und eine Leitung zur Mitführung des Bezugspotentials (CAN_GND)! Hierzu sollte die Abschirmung des Kabels verwendet werden!
3.	Das mitgeführte Bezugspotential CAN_GND muß an einem Punkt mit dem Erdpotential (PE) verbunden werden. Es muß genau eine Verbindung mit Erde hergestellt werden!
4.	Die Baudrate muß an die Leitungslänge angepaßt werden.
5.	Stichleitungen sind so kurz wie möglich zu halten ($l < 0,3\ \text{m}$)!
6.	Bei doppelt abgeschirmten Leitungen muß der äußere Schirm an einem Punkt mit dem Erdpotential (PE) verbunden werden. Es darf nicht mehr als einen Anschluß an Erde geben.
7.	Es ist ein geeigneter Leitungstyp (Wellenwiderstand ca. $120\ \Omega \pm 10\%$) zu verwenden und der Spannungsabfall auf der Leitung ist zu beachten!
8.	Die CAN-Leitungen sollten nicht in unmittelbarer Nähe von Störquellen verlegt werden. Läßt sich dies nicht vermeiden, so sind doppelt abgeschirmte Leitungen vorzuziehen.

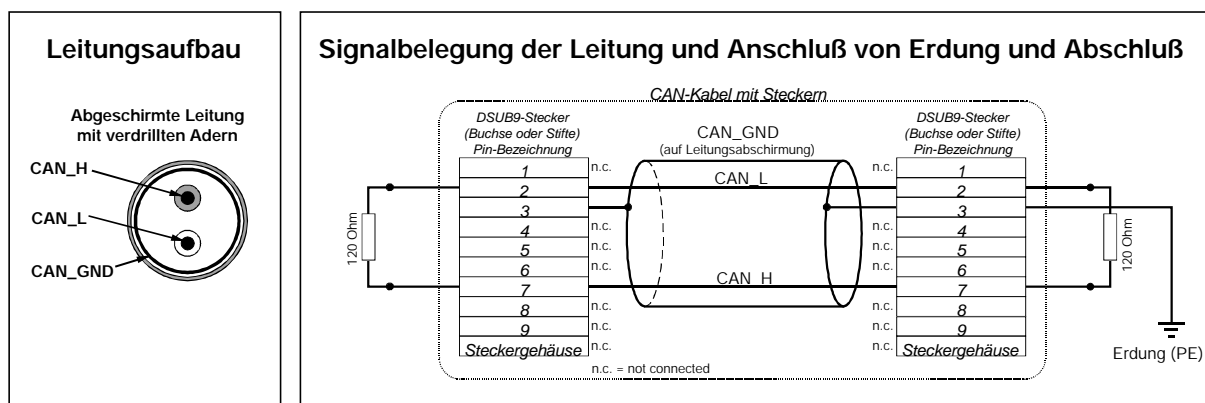
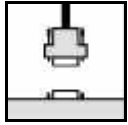


Abb.: Aufbau und Anschluß der Leitung



Verdrahtungshinweise

Verkabelung

- bei Geräten, die nur einem CAN-Stecker besitzen, T-Stück und Stichleitung (kürzer als 0,3 m) verwenden (als Zubehör lieferbar)

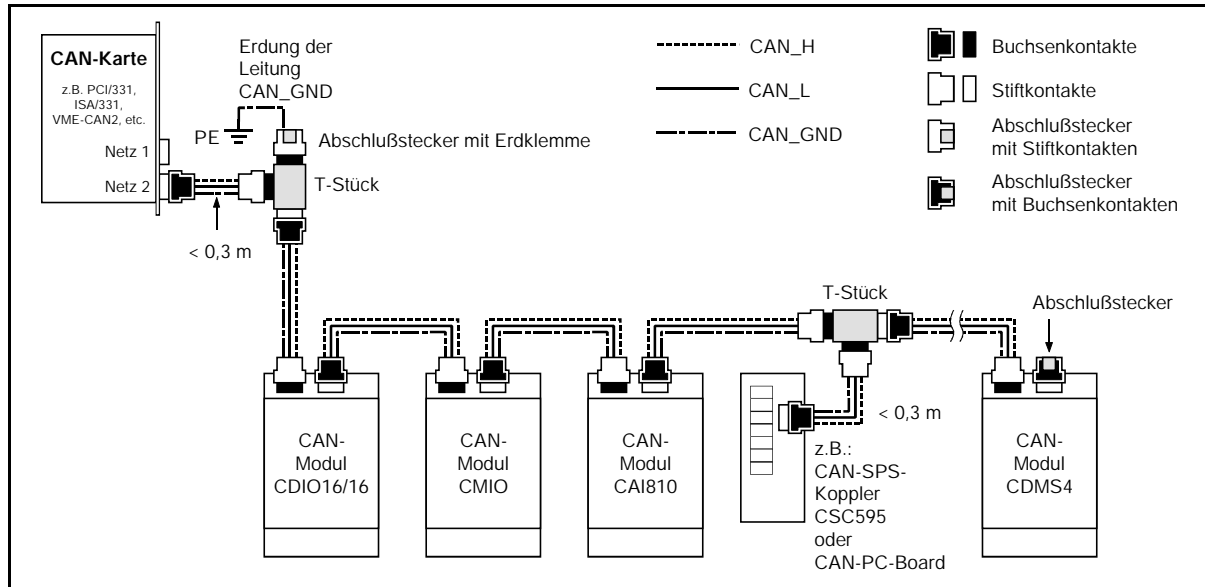


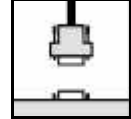
Abb.: Beispiel für korrekte Verdrahtung (bei Verwendung einfach abgeschirmter Leitungen)

Abschlußwiderstand

- **externen** Abschlußstecker verwenden, weil dieser später leichter auffindbar ist!
- 9-polige DSUB-Abschlußstecker mit Stift- oder Buchsenkontakten und Erdungsklemme sind als Zubehör erhältlich

Erdung

- CAN_GND muß in der CAN-Leitung mitgeführt werden, weil die einzelnen esd-Module galvanisch voneinander getrennt sind!
- CAN_GND muß an **exakt einem** Punkt im Netz mit dem Erdpotential (PE) verbunden werden!
- jeder CAN-Teilnehmer ohne galvanisch getrenntes Interface wirkt wie eine Erdung, darum: maximal einen Teilnehmer ohne Potentialtrennung anschließen!
- Erdung kann z.B. an einem Abschlußstecker vorgenommen werden



Leitungslänge

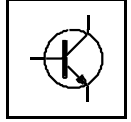
- Optokoppler verzögern die CAN-Signale. Durch den Einsatz schneller Optokoppler und den Test jedes Boards bei 1 MBit/s kann esd jedoch eine erreichbare Länge von 37 m bei 1 MBit/s garantieren. Voraussetzung hierfür ist ein abgeschlossenes Netz ohne Impedanzstörungen, wie z.B. längere Stichleitungen. (Ausnahme: CANbloc-Mini-DIO8 und -AI4, hier nur 10 m bei 1 MBit/s.)

Bit-Rate [kBit/s]	typische Werte der erreichbaren Leitungslänge mit esd-Interface l_{\max} [m]	CiA-Empfehlungen (07/95) für erreichbare Leitungslängen l_{\min} [m]
1000	37	25
800	59	50
666.6	80	-
500	130	100
333.3	180	-
250	270	250
166	420	-
125	570	500
100	710	650
66.6	1000	-
50	1400	1000
33.3	2000	-
20	3600	2500
12.5	5400	-
10	7300	5000

Tabelle: Erreichbare Leitungslängen in Abhängigkeit von der Bitrate beim Einsatz von esd-CAN-Interfaces

Beispiele für geeignete Leitungstypen

Hersteller	Leitungstyp
U.I. LAPP GmbH & Co. KG Schulze-Delitzsch-Straße 25 70565 Stuttgart	UNITRONIC ®-BUS LD, UNITRONIC ®-BUS FD P LD
metrofunk KABEL-UNION GmbH Postfach 410109 12111 Berlin	LiYCY 2 x 0,38 mm ² , LiYCY 2 x 0,5 mm ² , LiYCY 2 x 0,75 mm ² , LiYCY 2 x 1,0 mm ² , 1P x AWG 22 C, 1P x AWG 20 C
Alcatel Kabelmetal Kabelkamp 20 30179 Hannover	DUE 4401, DUE 4001, DUE 4402



7. Stromlaufpläne

CAN - CSC595/2

**CAN - PLC Interface Module
for
S5-90U, S5-95U and S5-100U**

DeviceNet

Software Manual

NOTE

The information in this document has been carefully checked and is believed to be entirely reliable. **esd** makes no warranty of any kind with regard to the material in this document, and assumes no responsibility for any errors that may appear in this document. **esd** reserves the right to make changes without notice to this, or any of its products, to improve reliability, performance or design.

esd assumes no responsibility for the use of any circuitry other than circuitry which is part of a product of **esd gmbh**.

esd does not convey to the purchaser of the product described herein any license under the patent rights of **esd gmbh** nor the rights of others.

esd electronic system design gmbh
Vahrenwalder Str. 205
D-30165 Hannover
Germany

Tel: +49-511-372-980
Fax: +49-511-633-650
Email: pm@esd.h.eunet.de

Manual File:	I:\TEXTE\DOKU\MANUALS\CAN\CSC595.2\SC0512SD.EN6
Date of Print:	28/10/97

Described Software:	DeviceNet driver for CSC595/2	CSC595/2 Kernel
Revision/Date:	V0.1f	V0.2b

Changes in the chapters

The changes in the user's manual listed below affect changes in the software, as well as changes in the description of the facts only.

Firmware Manual Version	Chapter	Alterations as compared to version 1.0
CSC595/2 V1.2	1.3	Commands DM, SM, and SY deleted in document, because they are designed for service and programming only.
	4.2	PLC address in example changed.
	-	-

Technical details are subject to change without notice.

1. Configuration Commands via RS-232	3
1.1 Introduction	3
1.2 Common Commands	3
1.2.1 Save Configuration	3
1.2.2 Set Bit Rate, Display Bit Rate	4
1.2.3 Wakeup Time	5
1.2.4 Module Number	6
1.2.5 Reset Module	6
1.2.6 CAN Default	6
1.3 Additional Common Commands	7
1.3.1 Help	7
1.4 Configuring the DeviceNet interface	8
1.4.1 Scanner Parameter	8
1.4.2 Installing COMM_ID (with PAM-scanner only)	8
1.4.3 Slave Parameter	9
1.5 Configuration of the PLC Interface	10
1.5.1 Slotnumber	10
1.5.2 Data Window	11
1.5.3 PLC - DeviceNet Mapping Table	12
2. The Command Page	13
2.1 Module State	13
2.2 Command Interface	14
3. Meaning of the CAN/DvN-LED in Front of the Module	16
4. Examples	17
4.1 Configuration Of A <i>FESTO</i> Pneumatic Valves Unit As A DeviceNet Slave	17
4.2 Configuration Of A <i>KUKA</i> Robot As A DeviceNet Master	22

1. Configuration Commands via RS-232

1.1 Introduction

The RS-232 interface is used for configuring and debugging. It can be used by a terminal or a PC with a terminal program (i.e. Windows terminal) with the following parameters:

Bit rate: 19200 baud
 Data: 8 bit
 Parity: no parity
 Stop bit: 1 stop bit
 Handshake: no handshake

All written parameters are interpreted as HEX-values !!!! (In this document HEX values are marked with an '\$' in front of the given value.)

The configuration parameters are only valid after storing them into the internal EEPROM ('CS') and resetting the module (e.g. by command 'RS' or by power_up).

An individual help to each command is printed by pressing the command with a following '?'.

1.2 Common Commands

1.2.1 Save Configuration

Name:	Save configuration
Input command syntax:	CS
Input parameter syntax:	> ok > error

Description:

All actual configuration parameters are stored into the internal EEPROM. The module answers with an 'ok', if well done, or with 'error', if an error occurred. The parameters are only valid until the next reset of the module.

1.2.2 Set Bit Rate, Display Bit Rate

Name:	Set bit rate
Input command syntax:	SB
Input parameter syntax:	< index(0-\$F)/register-format

Name:	Display bit rate
Input command syntax:	DB
Output parameter syntax:	> index(0-\$F)/register-format

The default bit rate is set by the coding switch at the front panel:

Setting of coding switch	Default-bit rate [Kbit/s]
0	125
1	250
2	500

Table 1.2.1: Setting of default bit rate

This bit rate can be overwritten by the command 'SB', with any value except \$FFFF. The new value can be displayed by calling the command 'DB'.

The bit rate can be given as index (0...\$F) or as the direct register format (intel CAN controller, 20 MHz):

Index [HEX]	Register [HEX]	Bit rate [bit/s]
0	0x1600	1 M
1	0x1b00	666 K
2	0x2f00	500 K
3	0x1b01	333 K
4	0x2f01	250 K
5	0x2f02	166 K
6	0x1c04	125 K
7	0x2f04	100 K
8	0x1b09	66 K
9	0x2f09	50 K
A	0x2f0e	33 K
B	0x2f18	20 K
C	0x2f27	12.5 K
D	0x2f31	10 K
E	0x7f7f	6 K
F	0x1200	1.6 M

Table 1.2.2: Setting bit rate by command

1.2.3 Wakeup Time

Name:	Wakeup time
Input command syntax:	WU
I/O parameter syntax:	<> time(ms)

Description:

After wakeup the module waits for the duration of the given wakeup time before it starts with any action at the CAN.

Acceptable values are 0...\$7FFF, according to 0...32767 ms. If the command is called without a parameter value, the actual value of the parameter is displayed at the monitor.

1.2.4 Module Number

Name:	Module number
Input command syntax:	MN
I/O parameter syntax:	<> number(1-\$3F)

Description:

The MAC_ID of the module is set by this command. If the command is called without a parameter value, the actual value of the MAC_ID is displayed at the monitor.

1.2.5 Reset Module

Name:	Reset module
Input command syntax:	RS
I/O parameter syntax:	No parameters

Description:

This commands is an easy way to reset the module (instead of a power_up).

1.2.6 CAN Default

Name:	CAN default
Input command syntax:	CD
I/O parameter syntax:	No parameters

Description:

With this command, all parameter are defaulted (but not actually stored in the EEPROM).

1.3 Additional Common Commands

1.3.1 Help

Name:	Help
Input command syntax:	??
Output parameter syntax:	> commands

Description:

This command returns a list of all possible commands (without following supervisor commands).

1.4 Configuring the DeviceNet interface

1.4.1 Scanner Parameter

Name:	Scanner parameter
Input command syntax:	SC
I/O parameter syntax:	<> scan_nr MAC_ID rxlen txlen rate

Parameter description:

scan_nr: table index (0...\$1F)
MAC_ID: MAC_ID of the module. that should be scanned
rxlen: Length of the produced data of that module (0...\$FE)
txlen: Length of the consumed data of that module (0...\$FE)
rate: Expected package rate of the connection (0 - \$FFFF)

Command description:

This command installs a module to the internal scanner. The 'scan_nr' means the table number of this entry. The scanner polls all modules written in this table up to the first not written entry. If an entry should be deleted, just type 'SC nr -' and the scanner stops at this point. The 'rxlen' and 'txlen' parameters are checked against the remote module and if a difference is recognized, a configuration error is shown to the PLC. The expected package rate is written to the remote module and the module is scanned by the half rate to reduce CAN access.

If the command is called without parameters, the actual values of the parameters are displayed at the monitor.

1.4.2 Installing COMM_ID (with PAM-scanner only)

Name:	COMM_ID install
Input command syntax:	CI
Input parameter syntax:	< scan_nr R/T COMM_ID < scan_nr
Output parameter syntax:	>RX_COMM_ID TX_COMM_ID

Parameter description:

can_nr: The entry-number from the scanner table above
R/T: 'R' for a receiving COMM_ID,
'T' for a transmitting COMM_ID
COMM_ID: COMM_ID (0...\$FFFE)

Command description:

This command installs a COMM_ID for a given scanner entry (refer to 1.5.1). If at least one COMM_ID is given for a scanner entry, the data length of this entry means the pure net data length without any COMM_ID. The scanner switches automatically to a special COMM_ID handler, who builds the correct transmit-frames with COMM_ID resp. extract raw data from received frames.

COMM_ID can only be added to a given scanner entry, but overwriting a scanner entry clears installed COMM_ID's of the old entry.

1.4.3 Slave Parameter

Name:	Slave parameter
Input command syntax:	SL
I/O parameter syntax:	<> rxlen txlen

Parameter description:

rxlen: Consumed_Data_Length (0...\$FE)

txlen: Produced_Data_Length (0...\$FE)

Command description:

The module can also be accessed by another scanner. You just have to define the Consumed_Data_Length (rxlen) and the Produced_Data_Length (txlen) with this command.

If the command is called without a parameter value, the actual value of the parameters are displayed at the monitor.

1.5 Configuration of the PLC Interface

The interface between the module and the PLC is located in the peripheral address range of the PLC. The absolute address of the module depends on the slotnumber of the module. Every slot occupies only 8 byte data of the PLC. To raise the performance of this interface, the module is able to simulate more than one slot. Due to this simulation the addresses of the following modules on the PLC bus are shifted to slots behind the last simulated slot.

There are generally two ways to access the data of DeviceNet:

1. Direct access through I/O-addresses is an easy and quick way to write and read data. It is just necessary to make a mapping table between I/O-addresses and DeviceNet data. The disadvantage is the small I/O data area of the PLC and the larger PLC cycle time simulating more data.
2. Indirect access through a command and a data window.
The command window is fixed in the first 4 bytes of the first simulated slot, input and output. So the module must be inserted in an 'analog slot', i.e. one of the first 8 slots!!! The data window is definable in address and range.

1.5.1 Slotnumber

Name:	Slot number
Input command syntax:	SN
I/O parameter syntax:	<> slot

Parameter description:

slot: slotnumber (0...7)

Command description:

Calling this command stores the slotnumber of the module. The module needs this number, to recalculate the relative address positions from absolute addresses. The absolute addresses are set by the commands described in the following chapters.

The slot number is necessary to communicate with the PLC.

1.5.2 Data Window

Name:	Window
Input command syntax:	WD
I/O parameter syntax:	< I/O addr len > I addr len O addr len

Parameter description:

I: Input Window, PLC can read data from DeviceNet

O: Output Window, PLC can write data to DeviceNet

addr: Absolute address of the window

len: Length of the window (0,2,4,8)

Command description:

This command defines the address and the range of the data window. Using this data window and additional the command window the PLC can access to the whole data area of the DeviceNet interface (chapter 2).

If the command is called without parameter values, the actual values of the parameters are displayed at the monitor.

1.5.3 PLC - DeviceNet Mapping Table

Name:	PLC parameter
Input command syntax:	PP
I/O parameter syntax:	<pre> < nr M P I/O M/I addr len MAC_ID offs < nr M P I/O M/I addr len MAC_ID offs COMM_ID < nr S P I/O M/I addr len offs < nr E > nr M/S/E P I/O M/I addr len [MAC_ID] offs [COMM_ID] </pre> <p style="text-align: center;"> </p> <p style="text-align: center;"> MOTOROLA / INTEL notation Input / Output Peripheral area access to Master / Slave, or table-End </p>

Parameter description:

nr: Table index (0...\$3F)
 addr: Absolute PLC address of this entry
 len: Data length of this entry
 MAC_ID: DeviceNet MAC_ID of the module, that should be accessed (Master only)
 offs: Offset inside the data stream of the device
 COMM_ID: COMM_ID of the data object (only on devices with special COMM_ID handling)

Command description:

This command installs a data mapping table between PLC and DeviceNet to perform direct data exchange. The link is made between PLC peripheral input or output ('I/O') address 'addr' with data length 'len' and DeviceNet data from the remote module 'MAC_ID' (via Master 'M') or from the own module (via Slave interface 'S') after position 'offs' of the data stream depending on the direction (I/O).

If it is a remote device with the special COMM_ID handling, the COMM_ID in this entry is necessary to distinguish between several objects.

SPECIAL:

A special feature is made with the 'M/I' letter: The PLC accesses to words in MOTOROLA notation, means a high-byte/low-byte orientation inside the PLC. Because most DeviceNet data is orientated in INTEL notation (low-byte/high-byte), it is possible to change this orientation with the 'I' letter. This is only possible with a even data length, otherwise it is switched back automatically.

The table ends at a special entry written with 'PP nr E'. If no parameter is written, this command shows the complete table.

2. The Command Page

The easiest and quickest data access is done with the mapping table (refer to 1.5.1). But it maybe possible that not all data is directly accessible due to the small peripheral address range of the PLC. Therefore it is possible to perform an indirect access with a command and a data window.

The data window is definable with the 'WD' command (refer to 1.5.2). The command window occupies the first 4 bytes of the module, input and output.

In input direction it holds some state information of the module. In output direction the PLC program can send a command with several specifier to the module. The module recognize a new command only, if at least one bit of this window is changed. So, if nobody writes to the command window, no command is executed and the old command has not to be reset.

2.1 Module State

The command window shows in input direction the state of the module. The memory layout with 'n' as the base address of the module is:

IB n:	state bits
IB n+1:	error MAC_ID
IB n+2 \	
IB n+3 - IW n+2:	special information, command dependent

The state bits in the first input byte are:

I n.7 :	can_off
I n.6 :	can_warn
I n.5 :	scan_ok
I n.4 :	slave_ok
I n.3 :	config_error
I n.2 :	future use
I n.1 :	success
I n.0 :	toggle

The bits 'can_off' and 'can_warn' show the actual state of the CAN chip.

'scan_ok' shows, if the internal scanner is connected to all slaves of the scan-table.

The bit 'slave_ok' means, that the internal slave interface is connected by another scanner. If the data length of the slave is zero, this bit is always set.

The bit 'config_error' shows a configuration error like 'Duplicate_MACID' or 'wrong data size' in scan-table.

The bits 'success' and 'toggle' belongs directly to the command interface: 'toggle' toggles after each finished command. 'success' shows, if this command was finished successfully.

The second input byte shows the MAC_ID of the module, which caused the last error like 'timeout' or 'configuration error'.

The default value is \$0xFF.

The second word of the command window is reserved for command dependent return values (future use).

2.2 Command Interface

Via the command window the PLC can send special commands to the module. A new command is only recognized by the module, if at least one bit of the command window is changed. So, if nobody writes to the command window, no command is executed, and the last written command has not to be reset. A recognized command is finished by set or reset the 'success' bit and by an inverse 'toggle' bit. The memory layout with 'n' as the base address of the module is:

QB n:	command
QB n+1:	subcmd
QB n+2 :	\
QB n+3:	-QW n+2 : command dependent information

Following commands are implemented:

command	subcmd	QB n+2	QB n+3	description
		QW n+2		
0x00	x	x		No command is executed.
0x01	x	x		Resets error: The error MACID (IB n+1) is reset to his default value \$0xFF.
0x02 +0x08= 0x0A	MACID	len	off	Reads 'len' data bytes from device 'MACID' after data offset 'off'. If 'MACID' > \$0x3F, data is read from the internal slave interface. Converts INTEL to MOTOROLA notation (refer to 1.5.3).
0x03 +0x08= 0x0B +0x10= 0x13 +0x08 +0x10= 0x1B	MACID	len	off	Writes 'len' data bytes to device 'MACID' after data offset 'off' and updates mirror_image to process image. If 'MACID' > \$0x3F, data is written to the internal slave interface. Converts INTEL to MOTOROLA notation (refer to 1.5.3). Writes only to mirror_image to have consistent data with the following writes. Converts INTEL to MOTOROLA notation and writes only to mirror_image.
0x04 +0x08= 0x0C	MACID +off	COMM_ID		Reads data from object 'COMM_ID' from PAM device 'MACID'. The data size is always equal to the whole data window size, the 2 MSB's in 'MACID+off' determine the offset in steps of the data-window size. Example: Is the read-data window size 8 bytes and the 2 MSB's are '01' (=1), the command reads from offset 8. Converts INTEL to MOTOROLA notation (refer to 1.5.3).
0x05 +0x08= 0x0D +0x10 0x15 +0x08 +0x10= 0x1D	MACID +off	COMM_ID		Writes data to object 'COMM_ID' from PAM device 'MACID' and updates mirror_image to process_image. The data size is always equal to the whole data window size, the 2 MSB's in 'MACID+off' determine the offset in steps of the data-window size. Converts INTEL to MOTOROLA notation (refer to 1.5.3). Writes only to mirror_image to have consistent data with the following writes. Converts INTEL to MOTOROLA notation and writes only to mirror_image.

3. Meaning of the CAN/DvN-LED in Front of the Module

At start-up the LED indicates the wakeup procedure by changing the colour periodically. During the wakeup time the module is passive on the CAN. The wakeup time can be set by the command 'WU'.

LED State	Meaning
LED off	The LED is turned off while sending the duplicate MAC-ID request.
LED green flashing	After wakeup and duplicate MAC-ID request the LED turns to flashing green. The communication between DeviceNet participants has not yet been established.
LED lights continuously green	There is no error and communication has been established
LED red flashing	A time-out has been detected in the communication between one DeviceNet participant and the CSC595/2.
LED lights continuously red	A fatal error has occurred. Please check the settings and wiring of attached DeviceNet nodes.

Table 3.1.1: LED states

4. Examples

4.1 Configuration Of A *FESTO* Pneumatic Valves Unit As A DeviceNet Slave

- Set the CAN bit rate of the CSC595 to the same value as all other CAN participants at this CAN net.
- Additionally it is necessary to define a MAC-ID for each device.

Use the according manuals for the above required settings .

For the following example the *MAC_ID* of the *FESTO* pneumatic valves has to be set to 5. All numbers in the following example has to be set as HEX values.

All commands used in this example are described in the DeviceNet software manuals of these devices

1. Use the command 'SC', to setup the slaves that shall be scanned by the master with the following parameters:

```

SC   0    5    3    2    64
    *    *    *    *    )   expected package rate is 100 ms (=0x64)
    *    *    *    *    )   -> polling cycle 50 ms
    *    *    *    *
    *    *    *    )   the master shall send 2 bytes to the module *)
    *    *    *
    *    *    )   the module shall answer with 3 bytes *)
    *    *
    *    )   the MAC-ID of the device has to be set to 5
    *
    )   indicates the first entry of the scan table

```

*) The number of bytes that has to be transferred is device-specific.

The end of the table has to be defined by the same command but without any parameters:

```

SC           1    -
    *        )   end of table
    *
    )   2nd entry in the scan table

```

2. Set bit rate to 500 kbit/s

```

SB           2
    )   bit rate index (2 = 500 kbit/s)

```

3. Use the command 'PP' to create the following PLC mapping table (two entries):

PP	0	M	P	O	I	4C	2	5	0	
*	*	*	*	*	*	*	*	*)	0 byte offset
*	*	*	*	*	*	*	*	*		
*	*	*	*	*	*	*	*)		MAC-ID of slave
*	*	*	*	*	*	*	*	*		
*	*	*	*	*	*	*)			length = 2 (data)
*	*	*	*	*	*	*	*			
*	*	*	*	*	*)				PLC address \$0x4C =
*	*	*	*	*	*	*				decimal 76 (AW 76)
*	*	*	*	*)					intel format
*	*	*	*	*						
*	*	*)							output
*	*	*	*							(view from PLC)
*	*	*	*							
*	*)								P-area
*	*	*								
*)									master (scanner)
*	*									
)										first table entry

PP	1	M	P	I	I	4C	2	5	0	
*	*	*	*	*	*	*	*	*)	0 byte offset
*	*	*	*	*	*	*	*	*		
*	*	*	*	*	*	*	*)		MAC-ID of slave
*	*	*	*	*	*	*	*	*		
*	*	*	*	*	*	*)			length = 2 (data)
*	*	*	*	*	*	*	*			
*	*	*	*	*	*)				PLC address \$0x4C =
*	*	*	*	*	*	*				decimal 76 (EW 76)
*	*	*	*	*)					intel format
*	*	*	*	*						
*	*	*)							input
*	*	*	*							(view from PLC)
*	*	*	*							
*	*)								P-area
*	*	*								
*)									master (scanner)
*	*									
)										2. table entry

4. If you wish to determine the state of the pneumatic valves, you have to set the PLC data mapping table to the following values:

PP	2	M	P	I	M	4E	1	5	2	
	*	*	*	*	*	*	*	*)	0 byte offset
	*	*	*	*	*	*	*	*		
	*	*	*	*	*	*	*)		MAC-ID of slave
	*	*	*	*	*	*	*			
	*	*	*	*	*	*)			length = 1 (status)
	*	*	*	*	*)				PLC address \$0x4C =
	*	*	*	*	*					decimal 78 (EB 78)
	*	*	*	*)					motorola format
	*	*	*	*						
	*	*	*)						input
	*	*	*							(view from PLC)
	*	*	*							
	*	*)							P-area
	*	*								
	*)								master (scanner)
	*									
)									3. table entry
PP	3	E								
	*)								end of table
	*									
)									4. entry in PLC data mapping table

5. The MAC-ID of the CSC595 shall be set to 17 in this example with the following command:

```
MN  11
*   )           MAC-ID of CSC595 ($0x11 = decimal 17)
*
)           module number
```

Important: Do not miss to save all settings with the command 'CS' and to trigger a RESET with the command 'RS' or with the PLC at the CSC595!

Additionally you must consider, in which slot the CSC595 is inserted:

SN 0

The module is inserted in slot 1 --> base address of this module is 64 (EW und AW)

If more PLC modules shall be inserted in the following slots of the PLC, the simulated I/O data length of the CSC595 has to be considered (unless, if the following modules are additional CSC595 modules or 'standard' PLC devices).

Does the first CSC595 simulate a maximum of 8 data bytes in the slot, the slot number for a second CSC595 has to be set to

SN 1

If the first CSC595 extends the data length of one slot, i.e. if it simulates more than 8 bytes, the slot number for a second CSC595 has to be set to

SN 2

This is necessary because the PLC must be able to address each data byte, even if it is one of a standard PLC device or if it is a simulated data byte of the CSC595. With each set of 8 simulated data bytes the slot number of the following PLC devices has to be incremented by '1':

If the first CSC595 is inserted in slot one and uses the data length of 2 slots, the second CSC595 must use the data bytes of the third slot.

Example for direct access:

As an option it is possible to access data directly without using the above listed table. The data bytes that are used with the direct access must be different to those which are used in the PLC data mapping table!

WD	I	44	8	data window EW 68 - EW 74 for common reading of data
WD	O	44	8	data window AW 68 - EW 74 for writing data (outside the data mapping table !)

4.2 Configuration Of A KUKA Robot As A DeviceNet Master

The robot acts as a master. The data of the robot shall be received using the PLC's peripheral data input bytes 80 to 85. Data that shall be transmitted to the robot using the PLC's peripheral data output bytes 80 to 83.

PP	00	S	P	I	M	50	6	0	
	*	*	*	*	*	*	*	*	
	*	*	*	*	*	*	*	*	
	*	*	*	*	*	*	*)	0 byte offset
	*	*	*	*	*	*	*		
	*	*	*	*	*	*)		length = 6 bytes
	*	*	*	*	*	*			
	*	*	*	*	*)			PLC address \$0x50 =
	*	*	*	*	*				decimal 80 (EY 80)
	*	*	*	*)				motorola format
	*	*	*	*					(no data swapping)
	*	*	*)					input
	*	*	*						(view from PLC)
	*	*	*						
	*	*)						P-area
	*	*							
	*)							slave
	*								
)								1. table entry
PP	01	S	P	O	M	50	4	0	
	*	*	*	*	*	*	*	*	
	*	*	*	*	*	*	*	*	
	*	*	*	*	*	*	*)	0 byte offset
	*	*	*	*	*	*	*		
	*	*	*	*	*	*)		length = 4 bytes
	*	*	*	*	*	*			
	*	*	*	*	*)			PLC address \$0x50 =
	*	*	*	*	*				decimal 80 (AY 80)
	*	*	*	*)				motorola format
	*	*	*	*					
	*	*	*)					input
	*	*	*						(view from PLC)
	*	*)						P-area
	*	*							
	*)							slave
	*								
)								2. table entry

```

PP  2    E
      *    )          end of table
      *
      )          3. entry of data mapping table

```

Setting the MAC-ID (module number) of the CSC595.

```

MN  11
      *    )          MAC-ID of the CSC595 ($0x11 = decimal 17)
      *
      )          module number

```

Setting the slave parameters of the CSC595:

```

SL  6    4
      *    )          produced data length of the internal slave (CSC595 ->robot)
      *
      )          consumed data length of the internal slave
                  (robot -> CSC595)

```

Setting the scanner parameters of the CSC595:

```

SC  0    -          no scan table, no master functionality

```

Now just save the data at the CSC595 with the command 'CS' and trigger RESET using the command 'RS'.