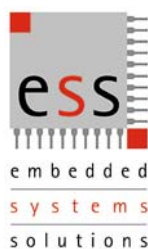


Benutzerhandbuch

CANgine

No. 1

März 2004 / FW 3.0





Die Informationen in diesem Dokument wurden sorgfältig erstellt und geprüft. Trotzdem übernimmt ESS keine Haftung für etwaige Fehler. Ebenso übernimmt ESS keine Haftung für Schäden, die durch die Verwendung dieses Handbuchs oder der hierin beschriebenen Produkte entstehen.

ESS behält sich vor Informationen in diesem Dokument und Eigenschaften der hierin beschriebenen Produkte ohne Ankündigung zu ändern.

Veröffentlichung und Vervielfältigung dieses Dokumentes nur mit schriftlicher Genehmigung von ESS.

Ausgabe	Datum	Status
4	Mär 2004	Inhaltsverzeichnis aktualisiert
3	Dez 2003	Angepasst auf FW 3.0
2	Sep 2003	Technische Daten ergänzt
1	Juli 2003	erste Ausgabe für Hardwarestand 2.0

© Copyright 2003, 2004

ESS Embedded Systems Solutions GmbH
Industriestr. 15
D-76829 Landau
(49) 6341 3487-0
(49) 6341 3487-29
info@ESSolutions.de
www.ESSolutions.de

www.CANgine.com

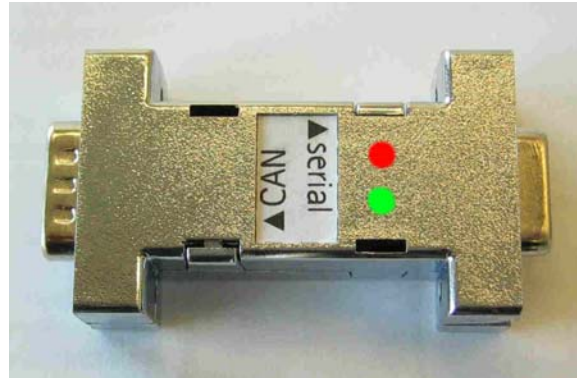
Inhalt

1. Einführung.....	4
2. Inbetriebnahme	4
3. Initialisierung beim Anlegen der Spannung.....	5
4. Befehlsbearbeitung	5
5. CAN Sende und Empfangspuffer.....	6
6. Fehlerbehandlung	6
7. Befehle.....	7
7.1 Übersicht.....	7
7.2 A Befehl	7
7.3 C Befehl	8
7.4 F Befehl.....	8
7.5 M Befehl.....	8
7.6 m Befehl.....	8
7.7 O Befehl	9
7.8 P Befehl	9
7.9 S Befehl	9
7.10 s Befehl	10
7.11 t Befehl	10
7.12 T Befehl.....	11
7.13 U Befehl.....	11
7.14 V Befehl.....	11
7.15 X Befehl.....	12
7.16 Z Befehl.....	12
8. Empfang von CAN Nachrichten	12
8.1 Ausgabeformat.....	12
8.1.1 Standard Frame.....	12
8.1.2 Extended Frame	13
8.2 CAN Akzeptanzfilterung.....	13
9. Inbetriebnahme und Test an einem PC	14
10. Steckerbelegung.....	15
10.1 Serielle Schnittstelle	15
10.2 CAN.....	15
11. Technische Daten.....	15
Life support.....	16
Right to make changes.....	16

1. Einführung

Wir freuen uns, dass Sie sich für ein Produkt der CANgine Familie entscheiden haben. Die CANgine Familie basiert auf leistungsfähigen 8 Bit Controllern mit integriertem Full CAN Interface und internem Flash Speicher. Damit können extrem kleine und trotzdem leistungsfähige CAN Einheiten aufgebaut werden.

Die Spannungsversorgung erfolgt über den CAN Stecker. Die serielle Schnittstelle spricht ASCII Code und kann damit an jedem Gerät betrieben werden, auf dessen serielle Schnittstelle man Zugriff hat. Damit ist die Anbindung nahezu aller Automatisierungskomponenten, auch älterer Geräte, an den CAN Bus möglich.



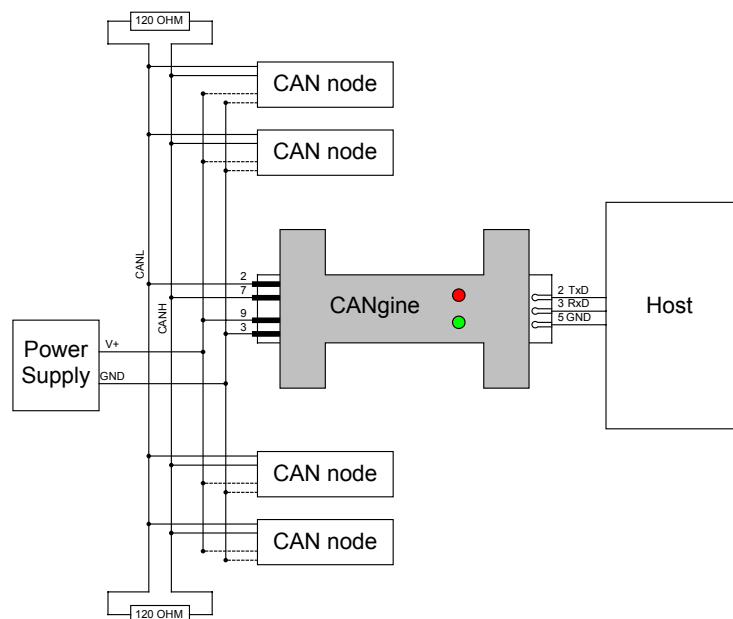
In seinem Standardgehäuse misst CANgine No. 1 nur 53 x 34 mm² bei einer Dicke von 16 mm. Wenn dies bei bestimmten Anwendungen nicht passen sollte, kann CANgine No. 1 schon bei kleinen Stückzahlen in Sonderbauformen auch ohne Gehäuse geliefert werden. Besprechen Sie Ihre Anforderungen mit unserem Vertrieb.

CANgine No. 1 unterstützt an der seriellen Schnittstelle Baudraten von 1200 bis 115200 bps und am CAN die vom CiA empfohlenen Baudraten zwischen 10 kBaud und 1 Mbaud. Inbetriebnahme und Fehleranalyse werden durch 2 LEDs unterstützt.

Softwareupdates oder kundenspezifische Software-Versionen können einfach ohne Öffnen des Gehäuses geladen werden.

2. Inbetriebnahme

Die Grafik zeigt den Anschluss der CANgine an ein CAN Netzwerk. Die Spannungsversorgung der CANgine erfolgt über die Pins 9 und 3 des CAN Steckers, wie vom CiA empfohlen. Die maximal zulässige Spannung beträgt 30V. Eine höhere Spannung führt zur Zerstörung der Elektronik, da die zulässige Verlustleistung innerhalb des Gehäuses überschritten wird. Insbesondere bei hohen Baudraten auf dem CAN ist auf die beiden Abschlusswiderstände von je 120 Ohm zu achten.



Nach dem Anlegen der Spannung blinkt die grüne RUN LED. Der Blinkcode gibt die eingestellte Baudrate wieder:

Anzahl	Bedeutung
2	115200
3	57600
4	38400
5	19200
6	9600
7	4800
8	2400

Dieser Blinkcode wird mit dem Aktivieren des CAN (O Befehl) beendet. Ab diesem Zeitpunkt signalisiert die LED die Betriebsbereitschaft des CAN.

CANgine No. 1 ist jetzt betriebsbereit und bearbeitet die Kommandos, die über die serielle Schnittstelle eingehen.

Zum Testen der CANgine No. 1 verwendet man am besten einen PC mit einem Terminalprogramm. Wir empfehlen folgende Einstellungen:

- Baudrate entsprechend Einstellung an CANgine
- 8 Datenbits
- no parity
- 1 Stopbit
- kein Protokoll
- Einfügen eines [NL] (0x0A) beim Empfang von [CR] (0x0D)
- lokales Echo

3. Initialisierung beim Anlegen der Spannung

Beim Einschalten der Spannung werden verschiedene Prüfungen durchgeführt, um eine sichere Funktion der CANgine zu gewährleisten. Hierzu gehört die Überprüfung des Programmspeichers anhand einer Checksumme. Um die Überprüfungen sichtbar zu machen, wird zu Beginn die rote Fehler ERR LED eingeschaltet und die grüne RUN LED ausgeschaltet. Tritt ein Firmware Checksummen Fehler auf, blinkt die rote ERR LED 7 mal und die grüne RUN LED ist aus.

Nach Abschluss der internen Überprüfungen werden die Peripheriebausteine initialisiert (CAN und RS232) und die rote ERR LED wird ausgeschaltet. Die grüne LED blinkt mit einer der seriellen Baudrate entsprechenden Anzahl Impulse (siehe Tabelle oben). Das Blinken wird beendet, wenn der CAN Bus freigeschaltet wird. Ab diesem Zeitpunkt zeigt die grüne RUN LED den Zustand des CAN Bus an (an = CAN aktiv, aus = CAN passiv oder Bus Off).

4. Befehlsbearbeitung

Sobald ein Befehl vom Host erkannt wurde, wird er ausgeführt. Jeder Befehl wird durch eine Ausgabe an den Host quittiert. Bevor die Ausgabe erfolgt ist, können keine neuen Befehle angenommen werden.



Die erfolgreiche Ausführung eines Befehls wird mit dem Zeichen [CR] (0x0D) quittiert, ein Fehler mit [BELL] (0x07) signalisiert.

Sowohl die serielle Schnittstelle als auch der CAN Bus werden interruptgesteuert betrieben. Damit hat der Empfang der eingehenden Nachrichten am CAN Bus hohe Priorität. Auch während ein Befehl bearbeitet wird, können Nachrichten am CAN empfangen werden.

5. CAN Sende und Empfangspuffer

CANgine hat zwei Ringpuffer zur Bearbeitung der CAN Nachrichten, die empfangen und gesendet werden. Jeder Puffer kann bis zu 30 Nachrichten aufnehmen.

Der Engpass im Nachrichtendurchsatz ist die serielle Schnittstelle. Sie begrenzt den maximalen Nachrichtendurchsatz bei 115200 kbaud auf ca. 350 standard Frames pro Sekunde bzw. 250 extended Frames pro Sekunde. Diese Werte wurden mit Nachrichtenlängen von 8 Datenbytes gemessen. Mit weniger Datenbytes ergibt sich eine Erhöhung dieser Werte. Durch Einstellen der Akzeptanzfilter so, dass für die an der CANgine arbeitende Anwendung nicht relevante Nachrichten ausgefiltert werden, kann CANgine an die aktuelle Anwendung optimal angepasst und der Datendurchsatz optimiert werden.

Zur Kontrolle der eingehenden Nachrichten besitzt CANgine zwei Zähler mit je 32 Bit Breite (0 bis 4.294.967.295). Einer zählt die korrekt in die Eingangs-Queue eingetragenen Nachrichten, der zweite die Nachrichten, die nicht in die Queue eingetragen werden konnten. Bei einem Überlauf beginnen die Zähler wieder bei 0 zu zählen. Die Zähler können mit dem X Befehl ausgelesen werden. Ein Lesen der Fehlerinformation mit dem F Befehl setzt beide Zähler auf den Wert 0 zurück.

6. Fehlerbehandlung

Folgende Fehler werden von CANgine intern erkannt und in einem Fehlermerker gespeichert:

Fehler	Bit des Fehlermerkers
CAN Empfangspuffer Überlauf	0
CAN Sendepuffer Überlauf	1
CAN "error passive" Zustand	5
CAN "bus error" Zustand	7

Fehler werden über die ERR-LED durch unterschiedliche Blinkzahlen angezeigt.

ERR (Blinkzahl)	Bedeutung
8	CAN bus error (bus off)
7	reserviert
6	CAN error passive state
5	reserviert
4	reserviert
3	reserviert
2	CAN transmit queue full
1	CAN receive queue full

Mit dem F Befehl kann der Fehlermerker ausgelesen werden. Ein Lesen des Fehlermerkers setzt ihn zurück.

Die ERR LED zeigt nur den letzten schwerwiegendsten Fehler an, während der Fehlermerker auch alte, nicht mehr anliegende Fehler speichert. Durch das Lesen des Fehlermerkers wird die LED ausgeschaltet und der Fehlermerker zurückgesetzt. Steht der Fehler nach wie vor an (z.B. CAN error passive), wird er sofort wieder gesetzt und die LED geht nicht aus. Dann muss erst der Fehler auf dem CAN Bus behoben werden (meistens falsche Baudrate bei einem CAN Knoten bevor der Fehler an der CANgine erfolgreich zurückgesetzt werden kann.

Steht beim Erkennen des F Befehls ein Bus Error an, so wird der CAN Controller neu initialisiert.

7. Befehle

7.1 Übersicht

A[CR]	Alle empfangenen CAN Frames ausgeben
C[CR]	CAN schließen
F[CR]	Fehlerinformation ausgeben
mxxxxxxxx[CR]	Akzeptanz Masken Register setzen
Mxxxxxxxx[CR]	Akzeptanz Tag Register setzen
O[CR]	CAN öffnen
P[CR]	Einen empfangenen CAN Frame ausgeben
Sn[CR]	CAN Baudrate auf n = 0..8 setzen (0 = 10 kbit, 8 = 1Mbit)
saabcde[CR]	CAN Baudrate Register einstellen
tiiidd..[CR]	Einen CAN Frame im Standardformat (11 Bit ID) senden
Tiiiiidd..[CR]	Einen CAN Frame im erweiterten Format (29 Bit ID) senden
Un[CR]	RS232 Baudrate auf n = 1..8 setzen (1 = 115200 baud, 8 = 1200 baud)
V[CR]	Versions Informationen ausgeben
X[CR]	CAN Frame Zählerinformationen ausgeben
Zn[CR]	Kontinuierliche Ausgabe der empfangenen Nachrichten umschalten

7.2 A Befehl

Alle empfangenen CAN Nachrichten an der seriellen Schnittstelle ausgeben. Das Ausgabeformat entspricht dem Eingabeformat (siehe T bzw. t Befehl). Nach jeder Nachricht wird ein [CR] ausgegeben, nach der letzten Nachricht zusätzlich ein A und [CR] als Endekennung.

Format:

A[CR]

Antwort:

Alle empfangenen Nachrichten

A[CR], wenn keine Nachrichten vorhanden sind

[BELL] wenn CAN nicht aktiv oder Befehlsformat ungültig

7.3 C Befehl

Inaktiviert den CAN Kanal der CANgine. Nach Ausführung dieses Befehls geht CANgine vom CAN Bus und beeinflusst diesen nicht mehr. Es wird auch kein ACK Bit mehr gesendet. Die grüne RUN LED erlischt.

Format:

C[CR]

Antwort:

[CR] wenn Befehl ausgeführt

[BELL] wenn CAN nicht aktiv oder Befehlsformat ungültig

7.4 F Befehl

Auslesen der Fehlerinformation und Zurücksetzen der CAN Nachrichtenzähler.

Format:

F[CR]

Antwort:

Fxx[CR]

[BELL] wenn Befehlsformat ungültig

F ist ein zweistelliger Hexadezimalwert mit folgender Bitbelegung

7	6	5	4	3	2	1	0
CAN Bus Off	res	CAN error passive	res	res	res	Transmit Puffer voll	Receive Puffer voll

7.5 M Befehl

Setzen des 32 Bit Akzeptanz Registers zur Filterung empfangener Nachrichten. Dieser Befehl ist nur bei deaktiviertem CAN zulässig.

Format:

Mcccccccc[CR]

wobei

cccccccc Akzeptanzregister (8 Hexadezimalzeichen)

Antwort:

[CR] bei erfolgreicher Ausführung

[BELL] wenn CAN aktiv oder Befehlsformat fehlerhaft

Zum Format und zur Bedeutung siehe Kapitel Akzeptanzfilterung.

7.6 m Befehl

Setzen des 32 Bit Akzeptanz Maskenregisters zur Filterung empfangener Nachrichten. Dieser Befehl ist nur bei deaktiviertem CAN zulässig.

Format:

mcccccccc[CR]
wobei
cccccccc Akzeptanzmaskenregister (8 Hexadezimalzeichen)

Antwort:

[CR] bei erfolgreicher Ausführung
[BELL] wenn CAN aktiv oder Befehlsformat fehlerhaft

Zum Format und zur Bedeutung siehe Kapitel Akzeptanzfilterung.

7.7 O Befehl

Aktiviert den CAN Bus. Mit diesem Befehl geht der Controller an den Bus und empfängt die CAN Nachrichten entsprechend dem eingestellten Akzeptanzfilter. Die grüne RUN LED geht an.

Format:

O[CR]

Antwort:

[CR] bei erfolgreicher Ausführung
[BELL] wenn CAN bereits aktiv ist oder bei Fehler im Befehlsformat

7.8 P Befehl

Ausgabe einer empfangenen CAN Nachricht. Liegen mehrere Nachrichten im Empfangspuffer, wird die älteste Nachricht an der seriellen Schnittstelle ausgegeben.

Format:

P[CR]

Antwort:

Eine CAN Nachricht im gleichen Format wie beim t bzw. T Befehl beschrieben, abgeschlossen mit [CR]
[CR] wenn keine Nachrichten im Empfangspuffer vorhanden sind
[BELL] wenn CAN inaktiv oder bei Fehler im Befehlsformat

7.9 S Befehl

Einstellen der CAN Baudrate auf einen vom CiA empfohlenen Wert. Der Befehl ist nur bei deaktiviertem CAN zulässig.

S0	10 kBit
S1	20 kBit
S2	50 kBit
S3	100 kBit
S4	125 kBit
S5	250 kBit
S6	500 kBit
S7	800 kBit
S8	1 MBit

Format:

Sn[CR]

Antwort:

[CR] bei erfolgreicher Ausführung

[BELL] wenn CAN aktiv oder bei Fehler im Befehlsformat

Antwort:

[CR] Befehl erfolgreich ausgeführt

[BELL] unzulässiger Wert für n oder CAN Kanal offen.

7.10 s Befehl

Einstellen des CAN Bus Timing über die internen Register des CAN Controllers. Siehe hierzu das Datenblatt des T89C51CC01.

Format:

saabcde[CR]

aa X2 Bit und BTR Register in zweistelliger hexadezimaler Darstellung (00..7F)

7	6	5	4	3	2	1	0
0	X2 Bit	BTR					

b PRS Register in einstelliger Darstellung (0..7)

c PHS1 Register in einstelliger Darstellung (0..7)

d PHS2 Register in einstelliger Darstellung (0..7)

e SJW Register in einstelliger Darstellung (0..3)

Antwort:

[CR] bei erfolgreicher Ausführung

[BELL] wenn CAN aktiv oder bei Fehler im Befehlsformat

7.11 t Befehl

Eintragen einer CAN Nachricht mit Standard Identifier (11 Bit) in den Sendepuffer. Der Sendepuffer wird kontinuierlich bearbeitet und sobald der CAN Controller zum Senden bereit ist, die nächste zu sendende Nachricht ausgelesen und dem CAN Controller zum Senden übergeben. Die Nachrichten gehen in der Reihenfolge des Eintragens in den Sendepuffer auf den Bus.

Format:

tiiiidd...[CR]

wobei

t Befehlszeichen

iii CAN Identifier mit fester Länge (3 Hexadezimalzeichen {000..7FF})

l Anzahl der Datenbytes der Nachricht (1 Zeichen {0...8})

dd... Datenbytes (Anzahl Zeichen 2 x l, jeweils {00..FF})

Antwort:

[CR] wenn Nachricht in Sendepuffer übernommen wurde

[BELL] wenn CAN inaktiv, Sendepuffer voll oder Befehlsformat ungültig

7.12 T Befehl

Eintragen einer CAN Nachricht mit Extended Identifier (29 Bit) in den Sendepuffer. Der Sendepuffer wird kontinuierlich bearbeitet und sobald der CAN Controller zum Senden bereit ist, die nächste zu sendende Nachricht ausgelesen und dem CAN Controller zum Senden übergeben. Die Nachrichten gehen in der Reihenfolge des Eintragens in den Sendepuffer auf den Bus.

Format:

Tiiiiiiiidd...[CR]

wobei

T Befehlszeichen

iiiiiii CAN Identifier mit fester Länge
(8 Hexadezimalzeichen {00000000..1FFFFFFF})

I Anzahl der Datenbytes der Nachricht (1 Zeichen {0...8})

dd... Datenbytes (Anzahl Zeichen 2 x I, jeweils {00..FF})

Antwort:

[CR] wenn Nachricht in Sendepuffer übernommen wurde

[BELL] wenn CAN inaktiv, Sendepuffer voll oder Befehlsformat ungültig

7.13 U Befehl

Einstellen der Baudrate der seriellen Schnittstelle. Die Befehlsantwort erfolgt noch in der alten Baudrate, damit die Antwort vor dem Umstellen der Baudrate erkannt werden kann.

U1	115200
U2	57600
U3	38400
U4	19200
U5	9600
U6	4800
U7	2400

Format:

Un[CR]

Antwort:

[CR] Befehl erfolgreich ausgeführt

[BELL] wenn Fehler im Befehlsformat

7.14 V Befehl

Dient zum Auslesen der Hardware und Software Versionsinformation.

Format:

V[CR]

Antwort:

Vhhss[CR]

wobei hh = Hardwareversion und ss = Softwareversion

[BELL] bei Fehler im Befehlsformat

7.15 X Befehl

Auslesen der CAN Nachrichtenzähler. Es wird zuerst der Zähler für die erfolgreich in den Empfangspuffer eingetragenen Nachrichten ausgegeben, dann ein [CR] und danach der Zähler für die Nachrichten, die aufgrund von Pufferüberlauf nicht eingetragen werden konnten. Abgeschlossen wird mit einem [CR].

Format:

X[CR]

Antwort:

s..s[CR]f..f[CR] wobei s..s = {0.. 4294967295} und f..f = {0.. 4294967295}
[BELL] bei Fehler im Befehlsformat

Die Zähler werden mit dem Auslesen der Fehlerinformation (F Befehl) zurückgesetzt.

7.16 Z Befehl

Mit diesem Befehl kann eingestellt werden, ob CANgine einen empfangen CAN Frame sofort auf dem Bildschirm ausgibt, oder auf ein P bzw. A Kommando wartet. Der Befehl ist nur bei deaktiviertem CAN zulässig.

Format:

Za[CR] wobei a=1 für ein (1) und a=0 für aus

Antwort:

[CR] bei erfolgreicher Ausführung
[BELL] wenn CAN aktiv, Zustand bereits eingestellt oder Fehler im Befehlsformat.

8. Empfang von CAN Nachrichten

Nachrichten, die auf dem CAN eingehen unterliegen der Akzeptanzfilterung (siehe Kap. 8.2). Passieren die Nachrichten die Akzeptanzfilterung werden sie in den Empfangspuffer eingetragen und je nach Einstellugn der CANgine entweder auf Anforderung (P oder A Kommando) oder kontinuierlich (Z1 Kommando) ausgegeben.

8.1 Ausgabeformat

Das Ausgabeformat entspricht dem Format für den Transmit Befehl, jedoch wird der empfangenen Nachricht das Zeichen 'r' für Frames mit Standard Identifier (11 Bit) bzw. 'R' für Frames mit extended Identifier (29 Bit) vorangestellt.

8.1.1 Standard Frame

riiIdd...[CR]

- r Zeichen für Empfangsframe im Standardformat (11 Bit Identifier)
- iii CAN Identifier mit fester Länge
(3 Hexadezimalzeichen {000..7FF})
- I Anzahl der Datenbytes der Nachricht (1 Zeichen {0...8})
- dd... Datenbytes (Anzahl Zeichen 2 x I, jeweils {00..FF})

8.1.2 Extended Frame

RiiiiiiiIdd...[CR]

wobei

- R Zeichen für Empfangsframe im extended Format (29 Bit Identifier)
- iiiiiii CAN Identifier mit fester Länge (8 Hexadezimalzeichen {00000000..1FFFFFFF})
- I Anzahl der Datenbytes der Nachricht (1 Zeichen {0..8})
- dd... Datenbytes (Anzahl Zeichen 2 x I, jeweils {00..FF})

8.2 CAN Akzeptanzfilterung

Die Akzeptanzfilterung erlaubt es nur die Nachrichten in einem CAN Knoten zu verarbeiten, die für die angeschlossene Anwendung interessant sind. Damit erlauben die CAN Controller eine erhebliche Entlastung der Prozessoren, insbesondere bei hohen Buslasten und / oder hohen Baudraten.

Zur Akzeptanzfilterung werden in CANgine zwei Register verwendet:

- Akzeptanzregister
- Maskenregister

Im Akzeptanzregister werden die Bits eingestellt, die bei den gewünschten Nachrichten-Identifiern auf 0 oder 1 stehen müssen. Im Maskenregister werden mit einer 1 die Bitpositionen markiert, bei denen ein Vergleich durchgeführt werden soll.

Beispiel für Standard-Identifizier: Sie wollen alle Nachrichten mit Identifiern zwischen 0x100 und 0x1FF empfangen, alle anderen Nachrichten sollen vom CAN Controller ausgefiltert werden. Hierzu werden die Bits 10..8 im Maskenregister mit einer 1 markiert und im Akzeptanzregister mit 001 gekennzeichnet. Was in der folgenden Tabelle anstelle von x eingetragen wird, ist gleichgültig, da im Maskenregister hier 0 steht und diese Bitpositionen nicht zum Vergleich herangezogen werden.

Bit	10	9	8	7	6	5	4	3	2	1	0
Akzep	0	0	1	x	x	x	x	x	x	x	x
Mask	1	1	1	0	0	0	0	0	0	0	0

Um CANgine flexible zu gestalten, wurden die beiden Register sowohl für Standard- als auch für Extended-Identifizier verwendet und um zwei weitere Bits ergänzt, das RTR- und das IDE-Bit.

Damit können nicht nur Identifier Filter gesetzt werden, es kann ausserdem auf RTR Nachrichten und auf Extended bzw. Standard Nachrichten gefiltert werden. In den 32 Bit Registern werden die Identifier Bits linksbündig auf Bit 31 geschoben und die Bits 2 für RTR und 0 für IDE vorgesehen. Die nachfolgende Tabelle zeigt die Positionen der Identifier-Filterbits für Standard und Extended Format.

S	10	9	8	7	6	5	4	3	2	1	0	-	-	-	-	-
Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
E	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13

S	-	-	-	-	-	-	-	-	-	-	-	-	-	-	RTR	-	IDE
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0	
E	12	11	10	9	8	7	6	5	4	3	2	1	0	RTR	-	IDE	

Mit Hilfe des Bit 0 in Akzeptanz- und Maskenregister wird entschieden, welches Format empfangen wird:

M[0]	m[0]	Verhalten
0	0	empfängt beide Frame Typen
0	1	empfängt nur Standard Frames
1	0	empfängt beide Frame Typen
1	1	empfängt nur Extended Frames

Beispiel:

Um nur Standard-Frames mit Identifiern zwischen 0x100 und 0x1FF zu empfangen, stellt man ein:

$$\begin{aligned} \text{Akzeptanzregister (M):} & \quad 0x100 * 2^{21} + 0 = 0x20000000 \\ \text{Maskenregister (m):} & \quad 0x700 * 2^{21} + 1 = 0xE0000001 \end{aligned}$$

Um nur Extended-Frames mit Identifiern zwischen 0x100 und 0x1FF zu empfangen, stellt man ein:

$$\begin{aligned} \text{Akzeptanzregister (M):} & \quad 0x100 * 2^3 + 1 = 0x00000801 \\ \text{Maskenregister (m):} & \quad 0x700 * 2^3 + 1 = 0x00003801 \end{aligned}$$

9. Inbetriebnahme und Test an einem PC

Stecken Sie die CANgine mit dem als RS232 gekennzeichneten Steckverbinder an den COM Port, den Sie im Terminalprogramm verwenden. Verbinden Sie CANgine mit dem CAN Bus, aus dem CANgine die Spannungsversorgung bezieht (7...30VDC).

Kontrollieren sie anhand der Anzahl von Blinkzyklen der grünen RUN-LED die eingestellte Baudrate der seriellen Schnittstelle und starten Sie das von Ihnen bevorzugte Terminalprogramm mit der entsprechenden Baudrate. Weitere Einstellungen 8 Datenbits, no parity, 1 Stopbit, kein Protokoll. Wir empfehlen außerdem, lokales Echo und Einfügen eines LF (0x0A) bei CR (0x0D) einzustellen, da CANgine kein Echo ausgibt und als Zeilenende nur ein CR schickt.

Geben Sie 'V' (es wird zwischen Groß- und Kleinbuchstaben unterschieden) ein, gefolgt von der Eingabetaste. CANgine antwortet mit Vhhff, wobei hh die Hardwareversion (z.B. 20) und ff die Firmwareversion (z.B. 21) anzeigt.

Stellen Sie jetzt die Baudrate am CAN mit dem Kommando S ein. Der Default Wert beträgt 125 kbps. Geben Sie dann Z1[CR] ein, damit CANgine alle empfangenen CAN Frames sofort an die serielle Schnittstelle ausgibt. Dann geben Sie O[CR] ein, um den CAN Kanal zu aktivieren.

Jetzt sehen Sie alle eingehenden Nachrichten sofort auf dem Terminalprogramm. Die ERR LED gibt an, ob alle Nachrichten in den Empfangspuffer eingetragen werden konnten, oder ob es einen Pufferüberlauf gegeben hat.

10. Steckerbelegung

10.1 Serielle Schnittstelle

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

10.2 CAN

Pin	Signal	Pin	Signal
1	nc	6	GND
2	CANL	7	CANH
3	GND	8	nc
4	nc	9	+Vcc
5	nc		

11. Technische Daten

Spannungsversorgung	7 .. 30	VDC
Stromaufnahme	35 (typ.)	mA
Interner Mikrocontroller	Clock intern: 40 Full CAN 2.0B Interface	MHz
CAN Transceiver	82C251	
CAN Schnittstelle	Sub-D 9 polig Stecker	
CAN Baudrate	10 .. 1.000	kBit
Serielle Schnittstelle	Sub-D 9 polig Buchse	
Serielle Baudrate	2.400 .. 115.200	baud
Anzeigen	LED RUN (grün) LED ERR (rot)	
Größe	53 x 54 x 16	mm ³
Gewicht	22	g
Temperaturbereich	-40 .. +80	°C

Disclaimers

Life support

These products are not designed for use in life support appliances, devices or systems where malfunction of these products can reasonably be expected to result in personal injury. ESS Embedded Systems Solutions customers using or selling these products for use in such applications do so at their own risk and agree to fully indemnify ESS Embedded Systems Solutions for any damages resulting from such application.

Right to make changes

ESS Embedded Systems Solutions reserves the right to make changes, without notice, in the products, and/or software, described or contained herein in order to improve design and/or performance. ESS Embedded Systems Solutions assumes no responsibility or liability for use of any of these products, conveys no license or title under any patent, copyright, or mask work to right to these products, and makes no representations or warranties that these products are free from patent, copyright, or mask work right infringement, unless otherwise specified.

ESS Embedded Systems Solutions GmbH
Industriestraße 15
D-76829 Landau
Germany
Phone +49 (0) 6341 34870
info@ESSolutions.de

Copyright 2002, 2003 ESS Embedded Systems Solutions GmbH
All rights reserved. Printed in Germany