
Card12

Hardware Version 1.1

Benutzerhandbuch

25. Juni 2008

Copyright (C)1999-2008 by
ELMICRO Computer GmbH & Co. KG
Hohe Str. 9-13 D-04107 Leipzig
Telefon: +49-(0)341-9104810
Fax: +49-(0)341-9104818
Email: leipzig@elmicro.com
Web: <http://elmicro.com>

Dieses Handbuch wurde sorgfältig erstellt und geprüft. Trotzdem können Fehler und Irrtümer nicht ausgeschlossen werden. ELMICRO übernimmt keinerlei juristische Verantwortung für die uneingeschränkte Richtigkeit und Anwendbarkeit des Handbuchs und des beschriebenen Produktes. Die Eignung des Produktes für einen spezifischen Verwendungszweck wird nicht zugesichert. Die Haftung des Herstellers ist in jedem Fall auf den Kaufpreis des Produktes beschränkt. Eine Haftung für eventuelle Mangelfolgeschäden wird ausgeschlossen.

Produkt- und Preisänderungen bleiben, auch ohne vorherige Ankündigung, vorbehalten.

Die in diesem Handbuch erwähnten Software- und Hardwarebezeichnungen sind in den meisten Fällen auch eingetragene Warenzeichen und unterliegen als solche den gesetzlichen Bestimmungen. Es kann aus dem Fehlen einer besonderen Kennzeichnung nicht darauf geschlossen werden, daß die Bezeichnung ein freier Warenname ist. Gleiches gilt für Rechte aus Patenten und Gebrauchsmustern.

Inhalt

1. Überblick	2
Technische Daten	2
Lieferumfang	3
2. Schnellstart	4
Schaltplan	4
3. Bestückungsplan	5
4. Jumper und Brücken	6
Jumper	6
Lötbrücken	6
5. Mechanische Abmessungen	8
6. Applikationshinweise	9
Verhalten nach Reset	9
Startup-Code	9
Zusatzinformationen im Web	10
7. Monitorprogramm TwinPEEKs	11
Serielle Kommunikation	11
Autostart-Funktion	11
Redirected Interrupt Vektors	11
8. Monitor Befehle	13
9. Memory Map	15
Anhang	16
Literatur	16
S-Record Format	16
EMV Hinweise	18

1. Überblick

Card12 ist ein einfach anzuwendendes Controller Modul im Scheckkartenformat auf Basis der 16-Bit Mikrocontrollerfamilie HC12 von Motorola. Das Card12 Controller Modul erleichtert die Evaluierung des Mikrocontrollerbausteins und ist eine schnell verfügbare, kostengünstige Ausgangsbasis für die Realisierung kleiner bis mittlerer Serienanwendungen.

Bei dem Modul Card12.D60A kommt eine MCU vom Typ MC912D60A zum Einsatz. Das selbe Modul ist erhältlich mit dem Controller MC912DG128A.

Die für den HC12 erhältliche umfassende Softwareunterstützung (Monitor, C-Compiler, BDM-Debugger) erleichtert die Entwicklung von Embedded Systemen.

Technische Daten

- Motorola 16 Bit Mikrocontroller MC912D60A im TQFP-112 Gehäuse
- 16 MHz Oszillatortakt, 8 MHz Bustakt
- 60 KB Flash Speicher
- 2 KB RAM
- 1 KB EEPROM
- 8 Kanal 16 Bit Timer mit Input Capture / Output Compare
- 16 Kanal 10 Bit A/D-Wandler
- 2 Kanal SCI-Schnittstelle mit RS232 Transceiver (MAX232E)
- 4 Kanal PWM
- Synchrones serielles Interface SPI
- COP Watchdog, Clock Monitor
- CAN 2.0A/B Businterface mit CAN-Treiber (PCA82C251)
- Resetcontroller zur Betriebsspannungsüberwachung
- bis zu 80 freie Ein-/Ausgabeleitungen

- alle I/O-Anschlüsse sind auf seitliche Steckverbinder herausgeführt
- Abmessungen: 54mm x 86mm
- 5V Betriebsspannung (keine zusätzliche 12V für Flash-Programmierung mehr erforderlich)
- typ. Stromaufnahme ca. 50 mA

Card12.DG128A (Unterschiede zur D60A-Version):

- Motorola 16 Bit Mikrocontroller MC912DG128A
- 128 KB Flash Speicher
- 8 KB RAM
- 2 KB EEPROM
- zwei CAN-Interface (1x inkl. Treiber, 1x TTL)

Lieferumfang

- Controller Modul mit MC912D60A (Card12.D60A) bzw. mit MC912DG128A (Card12.DG128A)
- TwinPEEKs Monitorprogramm (im Flash Speicher der MCU)
- RS232 Anschlußkabel (Sub-D9)
- zwei 50pol. Stiftleisten sowie Power-Stecker
- Hardwarehandbuch
- CD-ROM mit Assemblersoftware, verschiedene Datenblätter, HC12 Reference Manual, Softwarebeispiele, C-Compiler Demoversion u.v.m.

2. Schnellstart

Kein Mensch liest gern dicke Handbücher. Daher hier die wichtigsten Hinweise in Kürze. Wenn Sie sich jedoch über ein Detail einmal nicht sicher sind, dann informieren Sie sich am Besten in den nachfolgenden Kapiteln.

Und so können Sie beginnen:

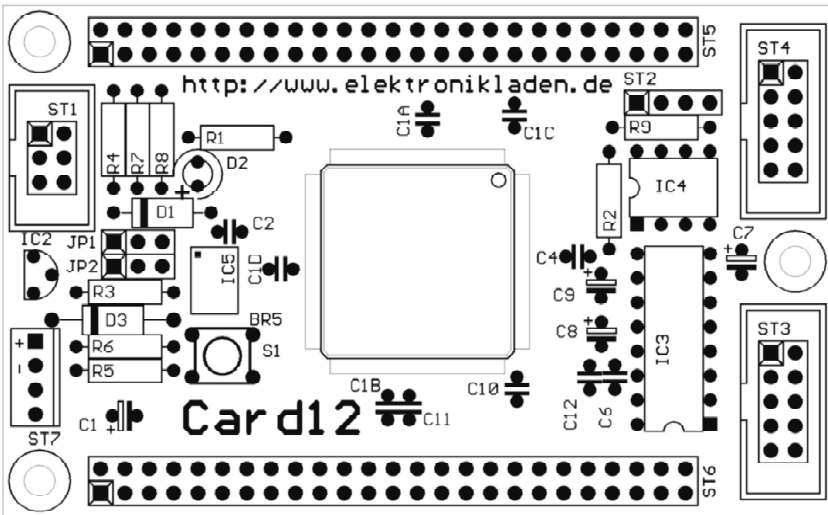
- Überprüfen Sie die Baugruppe zuerst auf offenkundige Transportschäden.
- Verbinden Sie das Controller Modul via RS232 mit Ihrem PC. Die Verbindung zwischen Card12 (Schnittstelle SER0, ST3) und PC erfolgt über das mitgelieferte 10-pol. Flachbandkabel.
- Starten Sie auf dem PC ein Terminalprogramm. Ein einfaches Programm wie OC-Console (kostenlos auf unserer Website!) reicht aus.
- Stellen Sie die Baudrate auf 19200 Baud. Schalten Sie alle zusätzlichen Protokolle (Hard- und Softwarehandshake) aus.
- Schließen Sie die (stabilisierte!) Versorgungsspannung an den Einplatinenrechner an, z.B. hier:
 - Masse an ST7 Pin 2
 - +5V an ST7 Pin 1
- Vergewissern Sie sich **zuvor** von der richtigen Spannung und Polarität!
- Daraufhin startet das Monitorprogramm und zeigt eine kurze Systemmeldung an. Mit Ausgabe des Promptzeichens erwartet es Ihre Anweisungen.

Wir wünschen Ihnen viel Erfolg bei Ihrer Arbeit mit der Card12!

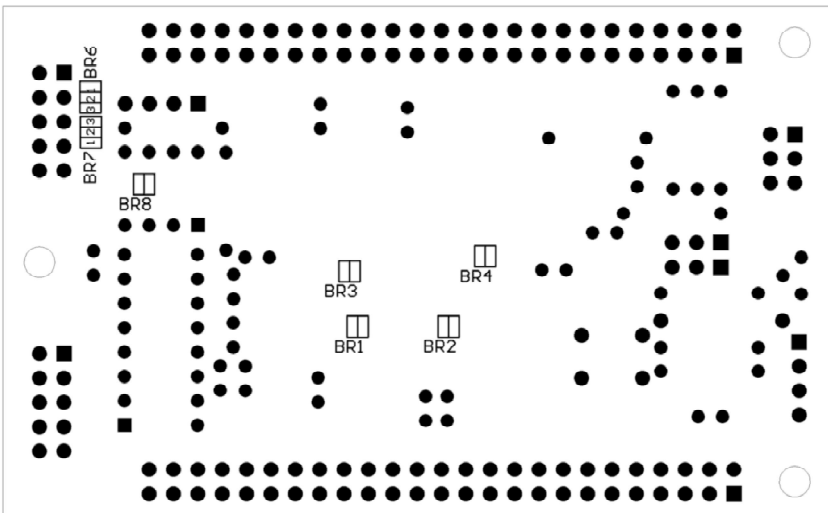
Schaltplan

Damit alle Details gut lesbar bleiben, liegt der Schaltplan im A4-Format separat bei.

3. Bestückungsplan



Lageplan Bestückungsseite



Lötbrücken auf der Platinenrückseite

4. Jumper und Lötbrücken

Jumper

JP1, JP2: MODA, MODB

1-2*	MODA/MODB=L (Single Chip Mode)
2-3	MODA/MODB=H (Expanded Wide Mode)

Lötbrücken

Die folgenden Lötbrücken befinden sich auf der Unterseite der Platine (vergl. Lageplan auf vorhergehender Seite):

BR1: VDDAD

offen	externe Einspeisung VDDAD
geschl.*	onboard mit VCC verbunden

BR2: VRH

offen	externe Einspeisung VRH
geschl.*	onboard mit VCC verbunden

BR3: VFP

offen*	VFP Pin nicht beschaltet (D60A, DG128A!!)
geschl.	VFP Pin verbunden mit externer Programmierspannung (nur für ältere D60 und DG128 Chips)

BR4: VSTBY (nur für DG128(A))

offen*	externe Einspeisung VSTBY möglich
geschl.	onboard mit VCC verbunden

BR5: EXTAL

offen	Takt (EXTAL) wird extern eingespeist
geschl.*	Oszillator IC5 liefert Takt (EXTAL)

* = Standardeinstellung

BR6, BR7: RS232 TxD/RxD Select (SER1)

- 1-2* RS232 als "Device" konfiguriert
(für Verbindung zu einem PC)
- 2-3 RS232 als "Host" konfiguriert
(für Verbindung zu seriellen LCD o.ä.)

BR8: LCD Power Supply (SER1)

- offen* VCC nicht am RS232 Anschluß SER1 verfügbar
(normale Belegung)
- geschl. VCC am RS232 Port SER1 verfügbar
(Pin9 des Sub-D Steckers)

* = Standardeinstellung

5. Mechanische Abmessungen

Die folgende Tabelle gibt die mechanischen Dimensionen des Card12 Moduls wieder. Die Angaben dienen als Orientierung beim Entwurf von Trägerplatten/-baugruppen.

Die südwestliche Ecke der Platine bildet den Koordinatenursprung. Die Lage der Platine ist horizontal, wie im Bestückungsplan (s.o.) dargestellt.

Alle Angaben zu Bohrungen (B) beziehen sich auf die Mitte, bei Steckverbindern (ST) auf die Lage von Pin 1.

	X in Zoll	Y in Zoll
ST1	0,150	1,575
ST2	2,600	1,700
ST3	3,150	0,675
ST4	3,150	1,825
ST5	0,400	1,900
ST6	0,400	0,100
ST7	0,133	0,725
B1	0,150	0,150
B2	0,150	1,950
B3	3,250	1,050
PCB	3,400	2,100

6. Applikationshinweise

In diesem Abschnitt sollen einige Besonderheiten der Programmierung des HC12 im allgemeinen und der Card12 Baugruppe im besonderen genannt werden.

Dieses Hardwarehandbuch kann nur einige *spezifische* Hinweise geben. Die Behandlung *allgemeiner* Techniken zur Programmierung des Controllers in Assembler bzw. Hochsprachen würden Umfang und Ziel dieses Handbuchs sprengen. Die meisten Antworten finden Sie beim (leider unerläßlichen) Studium der Datenblätter und Referenzhandbüchern der Halbleiterhersteller.

Verhalten nach Reset

Sobald die Resetleitung des Controllers freigegeben wird, holt sich die MCU die Information, an welcher Adresse das Programm des Anwenders beginnt. Der Controller liest hierzu den Resetvektor von den Speicherzellen \$FFFE und \$FFFF und springt dann an die dort angegebene Programmadresse.

Im Auslieferungszustand der Card12 ist im Flash-Bootblock (\$E000-\$FFFF) das Monitorprogramm TwinPeeks abgelegt. Der Resetvektor verweist auf den Beginn dieses Monitorprogramms. In Folge dessen startet nach jedem Reset automatisch TwinPeeks.

Um jedoch statt dessen ein im Flashspeicher (außerhalb des Bootblocks) abgelegtes Anwenderprogramm ohne Benutzereingriff starten zu können, ist in TwinPeeks ein Mechanismus eingebaut, der es ermöglicht, unter bestimmten Bedingungen automatisch zu dem Anwenderprogramm zu verzweigen. Die Details dieses Verfahrens sind im Kapitel Monitorprogramm beschrieben.

Startup-Code

Jede Controllerfirmware beginnt mit einer Reihe von Anweisungen zur Initialisierung der Hardware. Im Fall der Card12 beschränken sich die *unbedingt notwendigen* Initialisierungen auf zwei Aspekte, nämlich das Setzen des Stackpointers und die Abschaltung (bzw. ggf. die

geeignete Initialisierung) des Watchdog. Hier ein Assemblerbeispiel, welches durch seine besondere Schlichtheit besticht:

```
main    lds  #StackPtr    ; Init Stack Pointer
        clr  COPCTL      ; Disable Watchdog
```

Wichtiger Hinweis zum VFP-Pin des Controllers

Die Controller der ersten HC12 Generation (D60, DG128 etc.) benötigen eine Flash Programmierspannung von 12V (siehe Datenblatt!). Diese wird bei Card12 entweder über den Stromversorgungsanschluß ST7 oder über den BDM-Connector ST1 zugeführt. Die Lötbrücke BR3 muß geschlossen sein.

Die neueren A-Typen (D60A, DG128A), welche erstmals Mitte des Jahres 2001 verfügbar wurden, besitzen ein modifiziertes Flashmodul und erzeugen die Programmierspannung intern. **Bei den A-Typen muß BR3 geöffnet sein, um auszuschließen, daß am Pin 97 versehentlich 12V zugeführt werden - dies würde sonst zur Zerstörung des Controllerbausteins führen!**

Zusatzinformationen im Web

Sobald zusätzliche Informationen zu Hard- und Software der Card12 vorliegen, veröffentlichen wir diese auf unserer Website:

<http://elmicro.com/de/card12.html>

7. Monitorprogramm TwinPEEKs

Software Version 1.6

Serielle Kommunikation

TwinPEEKs kommuniziert über die erste RS232 Schnittstelle ("SER0") mit **19200 Baud**. Weitere Einstellungen: 8N1, kein Hardware- oder Softwarehandshake, kein Protokoll.

Autostart Funktion

Der TwinPEEKs Monitor überprüft nach Reset, ob die Port Pins PH6 und PH7 miteinander verbunden sind. Ist das der Fall, springt der Monitor zur Adresse \$8000 (*nur für DG128(A): \$4000*). Durch dieses Feature wird es möglich, ein Anwenderprogramm automatisch zu starten, ohne den Resetvektor im geschützten Flash Boot Block ändern zu müssen.

Redirected Interrupt Vectors

Die Interruptvektoren des HC12 liegen am Ende des 64 KB umfassenden Adreßraumes, d.h. innerhalb des schreibgeschützten Monitorcodes. Um dennoch Interruptfunktionen in einem Anwenderprogramm zu ermöglichen, leitet der Monitor alle Interruptvektoren (außer den Resetvektor) auf Adressen im internen RAM um. Das Verfahren entspricht der Vorgehensweise des HC11 im Special Bootstrap Mode.

Das Anwenderprogramm setzt den benötigten Interruptvektor, indem es einen Sprungbefehl in den RAM-Pseudovektor einträgt. Um z.B. den Timer Channel 4 (TC4-) Interrupt nutzen zu können, muß ein Anwenderprogramm folgende Schritte ausführen:

```
ldaa #$06           ; JMP Opcode
staa $07DC          ; TC4 Pseudo Vector
ldd #isrFunc        ; Jump Address
std $07DD           ; TC4 Pseudo Vector + 1
```

Als C-Programm läßt sich folgende Codesequenz verwenden:

```
// install TC4 pseudo vector in RAM
// (if running with TwinPEEKs monitor)
*(unsigned char *)0x07dc) = 0x06; // JMP opcode
*((void (**)(void))0x07dd) = isrFunc;
```

Der folgende Ausschnitt aus dem Assemblerlisting des Monitorprogramms dokumentiert die Adressen der umgeleiteten Interruptvektoren:

```
FFC2 :                org $ffc2                ; 912D60 *and* 912DG128
FFC2 : 07A6           dc.w RAMTOP-90           ; CGM LLH
FFC4 : 07A9           dc.w RAMTOP-87           ; MSCAN Tx
FFC6 : 07AC           dc.w RAMTOP-84           ; MSCAN Rx
FFC8 : 07AF           dc.w RAMTOP-81           ; MSCAN Err
FFCA : 07B2           dc.w RAMTOP-78           ; Pulse Accu B Overflow
FFCC : 07B5           dc.w RAMTOP-75           ; MDCU
FFCE : 07B8           dc.w RAMTOP-72           ; KWUG, KWUH
FFD0 : 07BB           dc.w RAMTOP-69           ; MSCAN Wake Up
FFD2 : 07BE           dc.w RAMTOP-66           ; ATD
FFD4 : 07C1           dc.w RAMTOP-63           ; SCI1
FFD6 : 07C4           dc.w RAMTOP-60           ; SCI0
FFD8 : 07C7           dc.w RAMTOP-57           ; SPI
FFDA : 07CA           dc.w RAMTOP-54           ; Pulse Accu Input Edge
FFDC : 07CD           dc.w RAMTOP-51           ; Pulse Accu Overflow
FFDE : 07D0           dc.w RAMTOP-48           ; Timer Overflow
FFE0 : 07D3           dc.w RAMTOP-45           ; TC7
FFE2 : 07D6           dc.w RAMTOP-42           ; TC6
FFE4 : 07D9           dc.w RAMTOP-39           ; TC5
FFE6 : 07DC           dc.w RAMTOP-36           ; TC4
FFE8 : 07DF           dc.w RAMTOP-33           ; TC3
FFEA : 07E2           dc.w RAMTOP-30           ; TC2
FFEC : 07E5           dc.w RAMTOP-27           ; TC1
FFEE : 07E8           dc.w RAMTOP-24           ; TC0
FFF0 : 07EB           dc.w RAMTOP-21           ; RTI
FFF2 : 07EE           dc.w RAMTOP-18           ; IRQ
FFF4 : 07F1           dc.w RAMTOP-15           ; XIRQ
FFF6 : 07F4           dc.w RAMTOP-12           ; SWI
FFF8 : 07F7           dc.w RAMTOP-9           ; Illegal Opcode
FFFA : 07FA           dc.w RAMTOP-6           ; COP Fail
FFFC : 07FD           dc.w RAMTOP-3           ; Clock Monitor Fail
FFFE : F000           dc.w main                ; Reset
```

In der DG128(A) Version des Monitors gilt o.a. Tabelle äquivalent, statt der Adresse \$07FF gilt hier lediglich \$3FFF als höchste (mit den Pseudovektoren belegte) RAM-Adresse. Der DG128(A) verfügt darüber hinaus über fünf zusätzliche Interruptvektoren:

```
FFB8 :                org $ffb8                ; 912DG128 *only!*
FFB8 : 3F97           dc.w RAMTOP-105          ; MSCAN1 Tx
FFBA : 3F9A           dc.w RAMTOP-102          ; MSCAN1 Rx
FFBC : 3F9D           dc.w RAMTOP-99           ; MSCAN1 Err
FFBE : 3FA0           dc.w RAMTOP-96           ; MSCAN1 Wake Up
FFC0 : 3FA3           dc.w RAMTOP-93           ; IIC
```

8. Monitor Befehle

TwinPEEKs Command Summary:

Dump: D (adr1 (adr2))

- Anzeige des Speicherinhaltes ab adr1 bis adr2

Edit: E (addr (byte))

- Speicher editieren ab Adresse addr
- Bei Angabe von byte wird gleich geschrieben, statt in den interaktiven Mode zu wechseln

Fill: F adr1 adr2 byte

- Füllt den Speicherbereich von adr1 bis (excl.) adr2 mit dem Wert byte

Goto: G (addr)

- Springt zum Anwenderprogramm ab Adresse addr

Help: H

- Command Summary

Load: L (offs)

- Lädt eine S-Record Datei in den Speicher
- Es können S1-Records (D60(A)) bzw. S1- und S2-Records (DG128(A)) geladen werden
- Das sendende Terminal muß nach jeder übertragenen S-Record Zeile auf die Empfangsbestätigung (*) warten!
- Bei Angabe von offs wird der zu ladende Block um den 16-Bit Wert offs verschoben.

Move: M adr1 adr2 adr3

- Kopiert den Speicherbereich von addr1 bis (excl.) addr2 nach addr3 ff.

Page: P (page)

- Selektiert eine Page (page=0..7). Die Page wird daraufhin im Speicher von \$8000 bis \$BFFF (16 KB) eingeblendet.
Nur für DG128(A).

Bulk: X (array)

- Löscht den Flash Memory
- Ohne Angabe von array: löscht gesamten Flash (außer oberen Bootblock)
- array = 0: löscht das untere 28 KB Flash Array
- array = 1: löscht das obere 32 KB Flash Array
- Der Monitor (im oberen Bootblock) wird in keinem Fall gelöscht! (zum Überschreiben des Monitors ist ein BDM-Lader erforderlich) *Nur für DG128(A): array = 0..3 (je 32 KB)*

9. Memory Map

Card12.D60A

Start	Ende	Belegung
\$0000	\$01FF	Steuerregister
\$0200	\$07FF	1,5KB (von 2KB) RAM (die unteren 0,5KB sind durch Steuerreg. verdeckt) TwinPEEKs verwendet die oberen 512 Bytes
\$0C00	\$0FFF	1KB EEPROM
\$1000	\$7FFF	28KB (unteres) Flash Memory Array
\$8000	\$FFFF	32KB (oberes) Flash Memory Array TwinPEEKs verwendet die oberen 8KB

Card12.DG128A

Start	Ende	Belegung
\$0000	\$03FF	Steuerregister
\$0800	\$0FFF	2KB EEPROM
\$2000	\$3FFF	8KB RAM TwinPEEKs verwendet die oberen 512 Bytes
\$4000	\$7FFF	16KB Flash (identisch mit Page 6)
\$8000	\$BFFF	16KB Flash Page 0 (Page 0..7 mittels PPAGE frei wählbar)
\$C000	\$FFFF	16KB Flash (identisch mit Page 7) TwinPEEKs verwendet die oberen 8KB

Anhang

Literatur

- [1] Kreidl, Kupris, Thamm: Mikrocontroller-Design
Hardware- und Software-Entwicklung mit dem 68HC12/HCS12;
Carl Hanser Verlag; 2003

S-Record Format

Das von Freescale publizierte S-Record Format ist ein Dateiformat zur Definition von Objektdateien (Maschinencode, Executables) unter Verwendung einer textuellen (ASCII-) Notation, die es erlaubt, diese Objektdateien mit jedem beliebigen Texteditor zu betrachten oder zu ändern. Eine S-Record Datei besteht aus einer beliebigen Anzahl S-Records bzw. Zeilen. Eine jede Zeile hat die folgende logische Struktur:

ID	LEN	ADDR	DATA	CS	<EOL>
----	-----	------	------	----	-------

Das Feld ID gibt den S-Record Typ an. Relevant sind die Typen "S1", "S2", "S8", "S9" und gelegentlich "S0" (Kommentarrecord). Außer dem ID Feld bestehen alle weiteren Felder aus Paaren von Hexziffern, beispielsweise "A9", "55" oder "0F".

Das Feld LEN besteht aus einem derartigen Hexziffernpaar und bestimmt die Anzahl der folgenden Ziffernpaare (enthält die Ziffernpaare der Felder ADDR, DATA und CS).

ADDR ist die Anfangsadresse der Datenbytes dieser Zeile. Das Feld besteht bei S1-Records aus zwei Byte (erst H-, dann L-Byte), d.h. aus zwei Ziffernpaaren.

DATA enthält die eigentlichen Codebytes, die das Maschinenprogramm bilden. DATA umfaßt (LEN - 3) Bytes bzw. Zeichenpaare bei S1-Records, (LEN-4) bei S2-Records.

Im Feld CS ist eine Prüfsumme enthalten. Sie wird gebildet aus den Werten der Zeichenpaare der Felder LEN, ADDR und DATA. CS ist das (niederwertigste Byte des) Einerkomplement der Summe aller vorgenannten Werte. EOL schließlich steht symbolisch für den durch CR, LF (\$0D, \$0A) gebildeten Zeilenvorschub.

Ein Beispiel soll die Handhabung verdeutlichen:

S1	13	2000	13A400262741010167CC10FF05C7A501	D1	<EOL>
----	----	------	----------------------------------	----	-------

Dieser S1-Record definiert \$13-3 = \$10 Bytes ab Adresse \$2000 des Zielsystems. Die Ziffernpaare des DATA Feldes ergeben eine Summe von \$04FB. Addiert man die \$13 aus dem LEN Feld sowie \$20 und \$00 aus dem ADDR Feld hinzu, ergibt sich ein Wert von \$052E. Das Einerkomplement des LSB (\$2E) ergibt \$D1. Dies ist der korrekte Wert für das Prüfsummenfeld.

Records vom Typ S2 enthalten ebenfalls Daten. Im Gegensatz zu S1-Records kommen bei S2-Records 24-Bit Adressen zum Einsatz. Demzufolge umfaßt das Adressfeld 6 statt 4 Stellen. Werden S2-Records verwendet, um Pagingdaten für den HC(S)12 zu definieren, errechnet sich die lineare 24-Bit Adresse wie folgt:

$$\text{ADDR24} = \text{PAGE} * 0\text{x}4000 + \text{OFFSET}$$

Neben den S1- und S2-Records, welche die eigentlichen Daten enthalten, werden S9- bzw. S8-Records als End-of-File Markierung verwendet. Abgesehen von dieser Terminierungs-Funktion kann in diesen Records die Startadresse des Programms vermerkt werden. Der Aufbau des S9-Records entspricht dem S1 Typ (S8 analog S9), wobei jedoch das Feld DAT leer bleibt. Das Feld ADDR spezifiziert die Startadresse des Programms. Ein typischer S9-Record sieht wie folgt aus:

S9	3	B600		46	<EOL>
----	---	------	--	----	-------

EMV Hinweise

Die Baugruppe entspricht den EMV-Vorschriften. Zur Stromversorgung ist sie an einer Batteriespannungsquelle mit 5,0 Volt (Einhaltung der Spannungsgrenzwerte beachten!) oder an ein Netzteil mit CE-Kennzeichnung anzuschließen. Der Einsatz einer Mikrocontrollerplatine geht stets einher mit einer mehr oder minder umfangreichen Modifikation der Baugruppe (spezielle Firmware, angeschlossene Peripheriebauteile). Der Hersteller kann den vom Kunden geplanten Einsatz der Baugruppe nicht vorhersehen und daher auch keine Vorhersagen über die EMV-Eigenschaften der modifizierten Baugruppe machen. Anwender ohne Zugriff auf ein EMV-Prüflabor sollten die folgenden Richtlinien beachten, die in der Regel eine einwandfreie Funktion der modifizierten Baugruppe gewährleisten:

Um sicherzustellen, daß die Baugruppe auch dann den EMV-Vorschriften entspricht, wenn Verbindungsleitungen zu anderen Geräten (z.B. Personalcomputer) angeschlossen werden oder die Baugruppe vom Kunden selbst mit weiteren Bauteilen nachgerüstet wird (z.B. Meßadapter oder Leistungsendstufen), empfehlen wir, die komplette Baugruppe in ein allseitig geschlossenes Metallgehäuse einzusetzen.

Wird ein LC-Display angeschlossen (ebenfalls auf CE-Kennzeichnung achten), so darf das Verbindungskabel nicht länger als 10 cm sein; hier ist auf jeden Fall ein Metallgehäuse vorzusehen. Wenn für die Programmentwicklung oder die spätere Anwendung die RS232 Schnittstelle benötigt wird, so ist ein max. 10cm langes Kabel zur Verbindung mit der Anschlußbuchse zu verwenden. Die geschirmte Anschlußbuchse ist fest mit dem Metallgehäuse zu verschrauben. Extern zur Verbindung verwendete Anschlußkabel müssen, ebenso wie der Hostrechner (PC), mit dem CE-Zertifizierungszeichen versehen sein.

Es wird darauf hingewiesen, daß der Anwender selbst dafür verantwortlich ist, daß eine veränderte, erweiterte, mit anderen als vom Hersteller gelieferten IC's bestückte oder mit Anschlußkabeln versehene Baugruppe den EMV-Vorschriften entspricht.