

# MC90 Kommunikation

Beschreibung der seriellen Kommunikationsprotokolle zwischen  
einem PC und Systemen der MC90 Familie.

# MC90 Kommunikation

---

Beschreibung der seriellen Kommunikationsprotokolle zwischen einem PC und Systemen der MC90 Familie.

Jede Vervielfältigung dieses Dokumentes sowie der zugehörigen Software oder Firmware bedarf der vorherigen schriftlichen Zustimmung durch die Fa. MICRO DESIGN Industrieelektronik GmbH. Zuwiderhandlung wird strafrechtlich verfolgt. Alle Rechte an dieser Dokumentation sowie der zugeordneten Software, Hardware und/oder Firmware liegen bei MICRO DESIGN.

Im Text erwähnte Warenzeichen werden unter Berücksichtigung und Anerkennung der Inhaber der jeweiligen Warenzeichen verwendet. Ein getrennte Kennzeichnung verwendeter Warenzeichen erfolgt im Text ggf. nicht durchgängig. Die Nichterwähnung oder Nichtkennzeichnung eines Warenzeichens bedeutet nicht, daß das entsprechende Zeichen nicht anerkannt oder nicht eingetragen ist.

Insofern diesem Dokument eine System- und/oder Anwendungssoftware zugeordnet ist, sind Sie als rechtmäßiger Erwerber berechtigt, diese Software zusammen mit MICRO DESIGN Hardwarekomponenten an Ihre Endkunden lizenzfrei weiterzugeben, solange keine getrennte, hiervon abweichende Vereinbarung getroffen wurde. Beinhaltet die diesem Dokument zugeordnete Software Beispielprogramme und Beispielapplikationen, so dürfen Sie diese nicht unverändert an Ihren Endkunden weitergeben, sondern ausschließlich zum eigenen Gebrauch und zu Lernzwecken verwenden.

Einschränkung der Gewährleistung: Es wird keine Haftung für die Richtigkeit des Inhaltes dieses Dokumentes übernommen. Da sich Fehler, trotz aller Bemühungen und Kontrollen, nie vollständig vermeiden lassen, sind wir für Hinweise jederzeit dankbar.

Technische Änderungen an der diesem Dokument zugeordneten Software, Hardware und/oder Firmware behalten wir uns jederzeit – auch unangekündigt – vor.

Copyright © 1998, 1999 MICRO DESIGN Industrieelektronik GmbH.

Waldweg 55, 88690 Uhdingen, Deutschland

Telefon +49-7556-9218-0, Telefax +49-7556-9218-50

E-Mail: [technik@microdesign.de](mailto:technik@microdesign.de)

<http://www.microdesign.de>

**We like to move it!™**

# Inhaltsverzeichnis

|   |          |
|---|----------|
| <b>Kapitel 1 Grundlagen .....</b>                   | <b>3</b> |
| n Verfügbare Treiberbibliotheken .....              | 3        |
| 1.1 Systemvoraussetzungen .....                     | 3        |
| n Verkabelung .....                                 | 3        |
| n Schnittstelleneinstellung .....                   | 3        |
| n Vorbereitung der Schnittstelle (nur MC90/B) ..... | 3        |
| 1.2 Verbindungsaufbau .....                         | 3        |
| n Typische Erkennungssequenz .....                  | 3        |
| n Erkennungssequenz im Quellcode .....              | 3        |
| 1.3 Grundsätzlicher Protokollaufbau .....           | 3        |
| n Sendetelegramme .....                             | 3        |
| n Empfangstelegramme .....                          | 3        |
| n Synchronität und Timeouts .....                   | 3        |
| n Checksummenberechnung .....                       | 3        |
| <b>Kapitel 2 Protokollreferenz .....</b>            | <b>3</b> |
| 2.1 Opcode 00H: Lese Variable .....                 | 3        |
| n Sendetelegramm .....                              | 3        |
| n Antworttelegramm .....                            | 3        |
| 2.2 Opcode 01H: Schreibe Variable .....             | 3        |
| n Sendetelegramm .....                              | 3        |
| n Antworttelegramm .....                            | 3        |
| 2.3 Opcode 06H: Lese I/Os und Merker .....          | 3        |
| n Sendetelegramm .....                              | 3        |
| n Antworttelegramm .....                            | 3        |
| n 6-Bit Codierung der Merkerdaten .....             | 3        |
| n Stellung des Schlüsselschalters .....             | 3        |
| 2.4 Opcode 07H: Setze Merker .....                  | 3        |
| n Sendetelegramm .....                              | 3        |
| n Antworttelegramm .....                            | 3        |
| 2.5 Opcode 09H: Lese Merker 300-499 .....           | 3        |
| n Sendetelegramm .....                              | 3        |
| n Antworttelegramm .....                            | 3        |
| 2.6 Opcode 0AH: Lese Merker 500-750 .....           | 3        |
| n Sendetelegramm .....                              | 3        |
| n Antworttelegramm .....                            | 3        |
| 2.7 Opcode 0BH: Schreibe erweiterten Merker .....   | 3        |
| n Sendetelegramm .....                              | 3        |

|   |          |
|---|----------|
| n Antworttelegramm.....                           | 3        |
| 2.8 Opcode 0CH: Lese UDB.....                     | 3        |
| n Sendetelegramm.....                             | 3        |
| n Antworttelegramm.....                           | 3        |
| 2.9 Opcode 0DH: Lese Daten aus Speicher.....      | 3        |
| n Sendetelegramm.....                             | 3        |
| n Antworttelegramm.....                           | 3        |
| 2.10 Opcode 0EH: Schreibe Daten in Speicher.....  | 3        |
| n Sendetelegramm.....                             | 3        |
| n Antworttelegramm.....                           | 3        |
| 2.11 Opcode 0FH: Lese Daten via MMU.....          | 3        |
| n Sendetelegramm.....                             | 3        |
| n Antworttelegramm.....                           | 3        |
| 2.12 Opcode 10H: Schreibe Daten via MMU.....      | 3        |
| n Sendetelegramm.....                             | 3        |
| n Antworttelegramm.....                           | 3        |
| <b>Kapitel 3 Spezielle Variablen.....</b>         | <b>3</b> |
| 3.1 Adress-Variable 65101.....                    | 3        |
| 3.2 Baudrate-Variable 65102.....                  | 3        |
| 3.3 Befehlsvariable 65107.....                    | 3        |
| <b>Kapitel 4 MC90/B Speicherorganisation.....</b> | <b>3</b> |
| n Lesen des Universal Data Block.....             | 3        |
| 4.1 Aufbau des Universal Data Blocks.....         | 3        |
| 4.2 Lesen von Daten aus dem Speicher.....         | 3        |
| n Variablen lesen.....                            | 3        |
| n Digitale Eingänge lesen.....                    | 3        |
| n Digitale Ausgänge lesen.....                    | 3        |
| n Merker lesen.....                               | 3        |
| <b>Anhang A Tabellenverzeichnis.....</b>          | <b>3</b> |

# Kapitel 1 Grundlagen

Diese Dokumentation beschreibt den Aufbau und die Syntax der Kommunikation zwischen einem PC (bzw. einem anderen seriellen Master) und Steuerungen der MC90 Familie. Zur MC90 Familie gehören folgende Steuerungssysteme:

- n MC90: Ur-Version der MC90 Steuerung im Einbaugeschütz.
- n MC90/A: Wie MC90, jedoch mit erweiterter Betriebssystemfunktionalität und zusätzlichen Protokollen. Benötigt Betriebssysteme ab Version 90.0.
- n MC90/B: Neue Version der MC90 Steuerung in Hutschienen-Modulbauweise. Grundsätzlich kompatibel zu MC90 und MC90/A, jedoch mit stark erweiterter Funktionalität und direktem Speicherzugriff.

## n Verfügbare Treiberbibliotheken

Müssen Sie überhaupt eine eigene Kommunikation schreiben? Bitte prüfen Sie anhand der folgenden Tabelle, ob möglicherweise bereits ein Treiber für Ihre Steuerung und das gewünschte Betriebssystem zur Verfügung steht:

| Betriebssystem            | MC90 und MC90/A   | MC90/B  |
|---------------------------|---|---|
| MS-DOS ab 3.11            | Treiber verfügbar als DLL oder C-Quellcode                          | Kein Treiber verfügbar  |
| Windows 3.11              | Verwenden Sie den MS-DOS Treiber                                    | Kein Treiber verfügbar  |
| Windows 95/Windows 98     | Nativer Treiber verfügbar (VMC Application Layer DLL/Gerätetreiber) | Nativer Treiber verfügbar (VMC Application Layer DLL/Gerätetreiber) |
| Windows NT 4/Windows NT 5 | Nativer Treiber verfügbar (VMC Application Layer DLL/Gerätetreiber) | Nativer Treiber verfügbar (VMC Application Layer DLL/Gerätetreiber) |
| Sonstige Betriebssysteme  | Kein Treiber verfügbar  | Kein Treiber verfügbar  |

n Tabelle 1 – Übersicht Treiberbibliotheken

# 1.1 Systemvoraussetzungen

## n Verkabelung

### MC90 und MC90/A

Sie benötigen eine zweiadrige Kabel. RXD und TXD müssen getauscht werden. Der Anschluß erfolgt an der mit "RS232" beschrifteten Buchse auf der Frontseite des Einbaugeschäfts.

### MC90/B

Sie benötigen ein vollständig durchverbundenes, 1:1 verdrahtetes, 9-poliges Standardkabel. Der Anschluß erfolgt an der Sub-D Buchse des Netzteilmoduls.

## n Schnittstelleneinstellung

Die Steuerung kann an jede beliebige RS232-Schnittstelle, die vom PC unterstützt wird, angeschlossen werden. Folgende Anschlußparameter sind notwendig:

| Bezeichnung | MC90      | MC90/A     | MC90/B     |
|-------------|-----------|------------|------------|
| Baudrate    | 9600 Baud | 19200 Baud | 38400 Baud |
| Parity      | Keine     | Keine      | Keine      |
| Datenbits   | 8         | 8          | 8          |
| Stopbits    | 1         | 1          | 1          |

n Tabelle 2 - Schnittstellenparameter

## n Vorbereitung der Schnittstelle (nur MC90/B)

Die MC90/B Steuerung verfügt auf dem Netzteilmodul über eine galvanisch getrennte Schnittstelle, um elektrischen Schäden am PC zu vermeiden. Aus diesem Grund erfolgt die Spannungsversorgung der RS232-Treiber auf dem Netzteilmodul durch die Handshakeleitungen des PC, die entsprechend gesetzt werden müssen.

Bevor eine Kommunikation mit der Steuerung aufgebaut wird, stellen Sie sicher, daß

- n DTR auf Dauer-High und
- n RTS auf Dauer-Low liegt.

Wenn diese Voraussetzungen nicht erfüllt sind, kann keine Kommunikation erfolgen, da dann die RS232-Treiber der Steuerung nicht mit Spannung versorgt werden!

## 1.2 Verbindungsaufbau

Bei einem MC90 Steuerungssystem wird die Baudrate der Verbindung durch die Parameterierung der Steuerung vorgegeben. Da diese Einstellung per Software geändert werden kann, sollte der PC beim Verbindungsaufbau mit der Steuerung alle gültigen Baudraten ausprobieren. Es wird aus Gründen der Performance empfohlen, stets mit der höchsten gültigen Baudrate zu beginnen und, falls mit der jeweiligen Baudrate keine Steuerung erkannt werden konnte, mit der nächsten Baudrate fortzufahren.

### n Typische Erkennungssequenz

Hier einmal der Ablauf einer typischen Erkennungssequenz für MC90 Steuerungen:

- n Höchste Baudrate (38400 Baud) aktivieren
- n Versuch, die Variable 65106 (enthält Versionskennung) auszulesen
- n Wenn erfolgreich, dann Steuerungstyp aus dem Inhalt der Variable ermitteln
- n Wenn nicht erfolgreich, Baudrate auf nächste unterstützte Baudrate herabsetzen und erneut lesen
- n Wenn alle Baudraten ausprobiert, Fehler ausgeben

Die jeweils aktuelle Baudrate innerhalb der Steuerung ist in der Variable 65102 codiert. Eine Beschreibung finden Sie in Kapitel 3.2 - Baudrate-Variable 65102 (Seite 3).

### n Erkennungssequenz im Quellcode

Folgende Erkennungssequenz (stark vereinfacht) wird von der VMC Application Layer DLL verwendet:

```
PurgeComm(hComm, PURGE_TXCLEAR | PURGE_RXCLEAR);
*IoBuffer = (UCHAR) STX;
IoBuffer[1] = (UCHAR) VMC_MC90Detection_Current;
IoBuffer[2] = (UCHAR) MC90_READVAR;
IoBuffer[3] = (UCHAR) 0x52;
IoBuffer[4] = (UCHAR) 0xFE;
IoBuffer[5] = (UCHAR) ETX;
IoBuffer[6] = (UCHAR) (STX+VMC_MC90Detection_Current+MC90_READVAR+0x52+0xFE+ETX);
if (VMC_Detect_Write(7L))
    if (VMC_Detect_Read(1L))
        if (*IoBuffer == MC90_ACKNOWLEDGE)
            if (VMC_Detect_Read(5L))
                if ((*IoBuffer == STX) && (IoBuffer[3] == ETX))
                    if ((IoBuffer[1] == 0) && (IoBuffer[2] == 0))
                        // MC90 Urversion!
                    else
                        Version = (USHORT) IoBuffer[1] + (((USHORT) IoBuffer[2]) << 8);
                        if (Version & 0x8000)
                            // MC90/B
                        else
                            // MC90/A
```

## 1.3 Grundsätzlicher Protokollaufbau

Sämtliche Protokolle, die zur Kommunikation mit einer Steuerung der MC90 Familie verwendet werden, unterliegen dem gleichen Protokollaufbau.

### n Sendetelegramme

Die nachfolgende Tabelle bestimmt die Struktur der an die Steuerung gesendeten Daten:

| Byte    | Inhalt    | Bedeutung   |
|---------|-----------|---|
| 1       | 02H (STX) | Telegramm-Start   |
| 2       | ??        | Steuerungsadresse (normalerweise 01H, entspricht dem Inhalt der Variable 65101 der Steuerung) |
| 3       | ??        | Telegramm-Opcode  |
| 4 – N   | ??        | Nutzdaten in Abhängigkeit vom Telegramm-Opcode  |
| (N + 1) | 03H (ETX) | Telegramm-Ende  |
| (N + 2) | ??        | Additive 8-Bit Checksumme der Telegrammdaten von Byte 1 bis Byte (N + 1)                      |

n Tabelle 3 – Grundsätzlicher Aufbau eines Sendetelegramms

### n Empfangstelegramme

Die Antwort der Steuerung auf das Sendetelegramm ist abhängig vom gewählten Opcode und vom Erfolg der Übertragung. Folgende Fälle sind möglich:

- n Die Steuerung hat ein korrektes Telegramm empfangen, aber die angegebene Steuerungsadresse stimmt nicht mit dem Inhalt der Variable 65101 überein. In diesem Fall wird das Telegramm ignoriert, die Steuerung antwortet nicht.
- n Die Steuerung hat ein Telegramm empfangen, jedoch ist die Syntax nicht korrekt oder die Checksumme stimmt nicht mit der von der Steuerung errechneten Checksumme überein. In diesem Fall sendet die Steuerung ein BEL-Character (07H) als "Not Acknowledge" Signal zurück.
- n Die Steuerung hat ein Telegramm empfangen und korrekt decodiert; es handelt sich jedoch um eine Anweisung, die keine zusätzlichen Informationen bei der Beantwortung benötigt (abhängig vom Telegramm-Opcode). In diesem Fall sendet die Steuerung ein ACK-Character (06H) als "Acknowledge" Signal zurück.
- n Die Steuerung hat ein Telegramm empfangen und korrekt decodiert; aufgrund des Telegramm-Opcodes steht fest, daß der Sender Nutzdaten zurück erwartet. Zunächst sendet die Steuerung ein ACK-Character (06H) als "Acknowledge"-Signal zurück, anschließend die entsprechenden Nutzdaten. Diesen Nutzdaten entsprechend dem Aufbau eines Sendetelegramms wie in Tabelle 3 – Grundsätzlicher Aufbau eines Sendetelegramms aufgezeichnet.

### n Synchronität und Timeouts

Nach jedem gesendeten Telegramm muß der Sender eine Antwort der Steuerung abwarten, bevor ein neues Telegramm gesendet werden darf. Ausnahme: die Steuerung antwortet nicht binnen eines realistischen Timeout-Wertes (bei 38400 Baud dürfte ein Timeout von max. 100ms realistisch sein). In diesem Fall kann das vorherige Telegramm wiederholt wiederholt werden.

Bitte beachten Sie jedoch, daß es stets einen triftigen Grund haben muss, wenn ein Telegramm von der Steuerung nicht beantwortet wird. Spätestens nach der dritten Wiederholung eines Telegramms ohne Antwort seitens der Steuerung können Sie davon ausgehen, daß die Steuerung nicht eingeschaltet, nicht betriebsbereit, nicht korrekt angeschlossen oder aber falsch parametrier ist.

## n Checksummenberechnung

Die verwendete Checksumme ist eine einfache additive Prüfsumme aller Datenbyte eines Telegramms mit Ausnahme der Checksumme selbst. Nehmen wir einmal an, die Variable 65106 soll aus der Steuerung gelesen werden. Dies würde folgendes Telegramm ergeben:

| Byte | Inhalt    | Bedeutung                                       |
|------|-----------|---|
| 1    | 02H (STX) | Telegramm-Start                                 |
| 2    | 01H       | Steuerungsadresse 1 (entspricht Variable 65101) |
| 3    | 00H       | Opcode "Lese Variable"                          |
| 4    | 52H       | Low-Byte der Variablennummer (FE52H = 65102)    |
| 5    | FEH       | High-Byte der Variablennummer (FE52H = 65102)   |
| 6    | 03H (ETX) | Telegramm-Ende                                  |
| 7    | ??        | Additive 8-Bit Checksumme der Telegrammdaten    |

n Tabelle 4 – Beispiel zur Checksummenberechnung

Die Checksumme ergibt sich in diesem Fall durch die 8-Bit Addition aller Daten der Byte 1-6, also:

$$02H + 01H + 00H + 52H + FEH + 03H = 156H \text{ AND } FFH = 56H$$

Das bedeutet: die Checksumme des oben angeführten Telegramms wäre 56H.

## n Raum für Ihre Notizen

## Kapitel 2 Protokollreferenz

In diesem Kapitel finden Sie eine Übersicht sämtlicher unterstützten Telegramm-Opcodes der MC90 Steuerungsfamilie.

Einige der hier aufgeführten Telegramm werden nur von bestimmten Steuerungstypen (MC90/A und/oder MC90/B) unterstützt. Die entsprechenden Telegramm sind deutlich markiert.

| Opcode | Bedeutung                                    | Parameter   | Rückgabe               | MC90 | MC90/A | MC90/B |
|--------|--|---|------------------------|------|--------|--------|
| 00H    | Lese die angegebene Variable                 | Nummer der Variable (2 Byte)                                  | Variablenwert (16 Bit) | Ja   | Ja     | Ja     |
| 01H    | Schreibe die angegebene Variable             | Nummer der Variable (2 Byte), neuer Wert (16 Bit)             | Keine                  | Ja   | Ja     | Ja     |
| 06H    | Lese alle I/Os und Merker                    | Keine   | Datenblock             | Ja   | Ja     | Ja     |
| 07H    | Schreibe den angegebenen Merker oder Ausgang | Merkernummer (8 Bit), neuer Wert (8 Bit)                      | Keine                  | Ja   | Ja     | Ja     |
| 09H    | Lese erweiterte Merker 300-499               | Merkernummer als Offset (8 Bit)                               | Datenblock             | Nein | Nein   | Ja     |
| 0AH    | Lese erweiterte Merker 500-750               | Merkernummer als Offset (8 Bit)                               | Datenblock             | Nein | Nein   | Ja     |
| 0BH    | Setze erweiterte Merker von 300-750          | Merkernummer (15 Bit), neuer Wert (1 Bit)                     | Keine                  | Nein | Nein   | Ja     |
| 0CH    | Lese Universal Data Block (UDB)              | Keine   | Datenblock             | Nein | Ja     | Ja     |
| 0DH    | Lese Daten aus Speicher                      | Speicheradresse (16 Bit), Länge der Daten (8 Bit)             | Datenblock             | Nein | Ja     | Ja     |
| 0EH    | Schreibe Daten in Speicher                   | Speicheradresse (16 Bit), Länge der Daten (8 Bit), Datenblock | Keine                  | Nein | Ja     | Ja     |
| 0FH    | Lese Daten aus Speicher via MMU              | Speicheradresse (24 Bit), Länge der Daten (8 Bit)             | Datenblock             | Nein | Ja     | Ja     |
| 10H    | Schreibe Daten in Speicher via MMU           | Speicheradresse (24 Bit), Länge der Daten (8 Bit), Datenblock | Keine                  | Nein | Ja     | Ja     |

n Tabelle 5 – Übersicht Protokoll-Opcodes

Detaillierte Beschreibungen zu den einzelnen Befehlen finden Sie auf den nächsten Seiten.

## 2.1 Opcode 00H: Lese Variable

Mit dem Befehl 00H kann eine beliebige Variable aus dem Speicher der Steuerung gelesen werden. Bitte beachten Sie, daß die erste gültige Variable die Nummer 65100 trägt, sprich: jede Variable einen Offset von 65100 beinhaltet.

### n Sendetelegramm

| Byte | Inhalt    | Bedeutung                                       |
|------|-----------|---|
| 1    | 02H (STX) | Telegramm-Start                                 |
| 2    | 01H       | Steuerungsadresse 1 (entspricht Variable 65101) |
| 3    | 00H       | Opcode "Lese Variable"                          |
| 4    | ??        | Low-Byte der Variablennummer                    |
| 5    | ??        | High-Byte der Variablennummer                   |
| 6    | 03H (ETX) | Telegramm-Ende                                  |
| 7    | ??        | Additive 8-Bit Checksumme der Telegrammdaten    |

n Tabelle 6 – Sendetelegramm Opcode 00H

### n Antworttelegramm

| Byte | Inhalt    | Bedeutung                                       |
|------|-----------|---|
| 1    | 06H (ACK) | Bestätigung: Sendetelegramm korrekt empfangen   |
| 2    | 02H (STX) | Telegramm-Start                                 |
| 3    | 01H       | Steuerungsadresse 1 (entspricht Variable 65101) |
| 4    | ??        | Low-Byte des Variablenwertes                    |
| 5    | ??        | High-Byte des Variablenwertes                   |
| 6    | 03H (ETX) | Telegramm-Ende                                  |
| 7    | ??        | Additive 8-Bit Checksumme der Telegrammdaten    |

n Tabelle 7 – Antworttelegramm Opcode 00H

## 2.2 Opcode 01H: Schreibe Variable

Mit dem Befehl 00H kann eine beliebige Variable im Speicher der Steuerung beschrieben werden. Bitte beachten Sie, daß die erste gültige Variable die Nummer 65100 trägt, sprich: jede Variable einen Offset von 65100 beinhaltet.

### n Sendetelegramm

| Byte | Inhalt    | Bedeutung                                       |
|------|-----------|---|
| 1    | 02H (STX) | Telegramm-Start                                 |
| 2    | 01H       | Steuerungsadresse 1 (entspricht Variable 65101) |
| 3    | 01H       | Opcode "Schreibe Variable"                      |
| 4    | ??        | Low-Byte der Variablennummer                    |
| 5    | ??        | High-Byte der Variablennummer                   |
| 6    | ??        | Low-Byte des neuen Variablenwertes              |
| 7    | ??        | High-Byte des neuen Variablenwertes             |
| 8    | 03H (ETX) | Telegramm-Ende                                  |
| 9    | ??        | Additive 8-Bit Checksumme der Telegramm Daten   |

n Tabelle 8 – Sendetelegramm Opcode 01H

### n Antworttelegramm

| Byte | Inhalt    | Bedeutung                                     |
|------|-----------|---|
| 1    | 06H (ACK) | Bestätigung: Sendetelegramm korrekt empfangen |

n Tabelle 9 – Antworttelegramm Opcode 01H

## 2.3 Opcode 06H: Lese I/Os und Merker

Der Befehl 06H liest alle Standard-Merker und I/O-Kanäle aus dem Speicher der Steuerung. Bitte beachten Sie: dieser Befehl ist vollständig kompatibel zur MC90-Urversion, d.h. es werden nur die unteren Merkerbereiche gelesen. Merker ab der Nummer 300 sind hier nicht berücksichtigt.

### n Sendetelegramm

| Byte | Inhalt    | Bedeutung                                       |
|------|-----------|---|
| 1    | 02H (STX) | Telegramm-Start                                 |
| 2    | 01H       | Steuerungsadresse 1 (entspricht Variable 65101) |
| 3    | 06H       | Opcode "Lese I/Os und Merker"                   |
| 4    | 03H (ETX) | Telegramm-Ende                                  |
| 5    | ??        | Additive 8-Bit Checksumme der Telegrammdaten    |

n Tabelle 10 – Sendetelegramm Opcode 06H

### n Antworttelegramm

| Byte    | Inhalt    | Bedeutung  |
|---------|-----------|--|
| 1       | 06H (ACK) | Bestätigung: Sendetelegramm korrekt empfangen                                    |
| 2       | 02H (STX) | Telegramm-Start  |
| 3       | 01H       | Steuerungsadresse 1 (entspricht Variable 65101)                                  |
| 4 – 9   | ??        | Status der ersten 48 digitalen Eingänge  |
| 10 – 26 | ??        | Status der ersten 88 Merker in 6-Bit Codierung                                   |
| 27      | ??        | Status des Schlüsselschalters (MC90 und MC90/A) bzw. Menü-Modus Auswahl (MC90/B) |
| 27 – 35 | ??        | Status der ersten 72 digitalen Ausgänge  |
| 36      | 03H (ETX) | Telegramm-Ende   |
| 37      | ??        | Additive 8-Bit Checksumme der Telegrammdaten                                     |

n Tabelle 11 – Antworttelegramm Opcode 06H

## n 6-Bit Codierung der Merkerdaten

Aus historischen Gründen sind die Merkerdaten in der Steuerung im 6-Bit Format abgelegt – dies beruht auf einer alten Limitierung einer relaisbasierten Vorgängersteuerung und wurde aus Gründen der Abwärtskompatibilität beibehalten. Um dieses etwas unübliche Format in eine normale 8-Bit Adressierung umzuwandeln, wird innerhalb der VMC Application Layer eine Konvertierungsroutine verwendet, die Sie als Quelltext-Auszug auf der nächsten Seite finden:

```

PUCHAR PtrMarkerSource = (Adresse Antworttelegramm + 9)
PUCHAR PtrMarkerDest = (Adresse zur internen Speicherung der Merker)
ULONG i;

for (i = 0; i < 90; i++)
{
    if (i)
    {
        if (!(i % 6))
            PtrMarkerSource++;
        else
            *PtrMarkerSource >>= 1;

        if (!(i % 8))
            PtrMarkerDest++;
    }
    *PtrMarkerDest |= ((*PtrMarkerSource & 0x01) << (i % 8));
}

```

## n Stellung des Schlüsselschalters

Steuerungen der Typen MC90 und MC90/A verfügen über einen Schlüsselschalter zur Auswahl des Betriebsmodus auf der Frontseite des Gehäuses, bei MC90/B Steuerungen kann dieser Modus über Tastenkombinationen am Handbedienteil ausgewählt werden.

Der jeweilige Status wird in dem Byte "Stellung des Schlüsselschalters" zurückgegeben. Folgende Werte sind gültig:

| Wert | Bedeutung                     |
|------|-------------------------------|
| 00H  | Betriebsmodus "Automatik"     |
| 04H  | Betriebsmodus "Manuell"       |
| 08H  | Betriebsmodus "Diagnose"      |
| 10H  | Betriebsmodus "Programmieren" |

n Tabelle 12 – Stellung des Schlüsselschalters

## 2.4 Opcode 07H: Setze Merker

Mit dem Befehl 07H können Merker im Bereich von 1 bis 255 gesetzt werden. Bitte beachten Sie hierbei:

- n Merker bis 109 kennzeichnen den Status eines digitalen Ausgangs
- n Merker ab 110 kennzeichnen den Status einen Bit-Merker
- n Merker größer als 255 können nicht mit diesem Befehl geschrieben werden. Verwenden Sie in diesem Fall den Befehl "Setze erweiterten Merker" (Opcode 0BH).

### n Sendetelegramm

| Byte | Inhalt    | Bedeutung  |
|------|-----------|--|
| 1    | 02H (STX) | Telegramm-Start                                  |
| 2    | 01H       | Steuerungsadresse 1 (entspricht Variable 65101)  |
| 3    | 06H       | Opcode "Setze Merker"                            |
| 4    | ??        | Nummer des zu setzenden Merkers/Ausgangs         |
| 5    | ??        | Neuer Status des Merkers/Ausgangs (0=aus, 1=ein) |
| 6    | 03H (ETX) | Telegramm-Ende                                   |
| 7    | ??        | Additive 8-Bit Checksumme der Telegrammdaten     |

n Tabelle 13 – Sendetelegramm Opcode 07H

### n Antworttelegramm

| Byte | Inhalt    | Bedeutung                                     |
|------|-----------|---|
| 1    | 06H (ACK) | Bestätigung: Sendetelegramm korrekt empfangen |

n Tabelle 14 – Antworttelegramm Opcode 07H

#### Hinweis

Dieser Befehl wird aus Kompatibilitätsgründen bei allen Steuerungen der MC90 Familie unterstützt. Ist sichergestellt, daß Sie mit einer Steuerung des Typs MC90/B arbeiten, verwenden Sie besser den Befehl "Setze erweiterten Merker" (Opcode 0BH), mit dem alle Ausgänge und Merker beschrieben werden können.

## 2.5 Opcode 09H: Lese Merker 300-499

Mit dem Befehl 09H kann ein Merker des erweiterten Merkerbereichs von 300-499 gelesen werden. Die Merker Nummer wird als ein Offset ab dem Merker 300 angegeben.

### n Sendetelegramm

| Byte | Inhalt    | Bedeutung                                       |
|------|-----------|---|
| 1    | 02H (STX) | Telegramm-Start                                 |
| 2    | 01H       | Steuerungsadresse 1 (entspricht Variable 65101) |
| 3    | 09H       | Opcode "Lese Merker 300-499"                    |
| 4    | ??        | Nummer des Merkers – 300                        |
| 5    | 03H (ETX) | Telegramm-Ende                                  |
| 6    | ??        | Additive 8-Bit Checksumme der Telegramm Daten   |

n Tabelle 15 – Sendetelegramm Opcode 09H

### n Antworttelegramm

| Byte | Inhalt    | Bedeutung                                       |
|------|-----------|---|
| 1    | 06H (ACK) | Bestätigung: Sendetelegramm korrekt empfangen   |
| 2    | 02H (STX) | Telegramm-Start                                 |
| 3    | 01H       | Steuerungsadresse 1 (entspricht Variable 65101) |
| 4    | ??        | Status des Merkers: 0=aus, ansonsten ein        |
| 5    | 03H (ETX) | Telegramm-Ende                                  |
| 6    | ??        | Additive 8-Bit Checksumme der Telegramm Daten   |

n Tabelle 16 – Antworttelegramm Opcode 09H

## 2.6 Opcode 0AH: Lese Merker 500-750

Mit dem Befehl 08H kann ein Merker des erweiterten Merkerbereichs von 500-750 gelesen werden. Die Merker Nummer wird als ein Offset ab dem Merker 500 angegeben.

### n Sendetelegramm

| Byte | Inhalt    | Bedeutung                                       |
|------|-----------|---|
| 1    | 02H (STX) | Telegramm-Start                                 |
| 2    | 01H       | Steuerungsadresse 1 (entspricht Variable 65101) |
| 3    | 0AH       | Opcode "Lese Merker 500-750"                    |
| 4    | ??        | Nummer des Merkers – 500                        |
| 5    | 03H (ETX) | Telegramm-Ende                                  |
| 6    | ??        | Additive 8-Bit Checksumme der Telegramm Daten   |

n Tabelle 17 – Sendetelegramm Opcode 0AH

### n Antworttelegramm

| Byte | Inhalt    | Bedeutung                                       |
|------|-----------|---|
| 1    | 06H (ACK) | Bestätigung: Sendetelegramm korrekt empfangen   |
| 2    | 02H (STX) | Telegramm-Start                                 |
| 3    | 01H       | Steuerungsadresse 1 (entspricht Variable 65101) |
| 4    | ??        | Status des Merkers: 0=aus, ansonsten ein        |
| 5    | 03H (ETX) | Telegramm-Ende                                  |
| 6    | ??        | Additive 8-Bit Checksumme der Telegramm Daten   |

n Tabelle 18 – Antworttelegramm Opcode 0AH

## 2.7 Opcode 0BH: Schreibe erweiterten Merker

Mit dem Befehl 0AH können alle Merker gesetzt werden. Bitte beachten Sie hierbei:

- n Merker bis 109 kennzeichnen den Status eines digitalen Ausgangs
- n Merker ab 110 kennzeichnen den Status einen Bit-Merker

### n Sendetelegramm

| Byte | Inhalt    | Bedeutung  |
|------|-----------|--|
| 1    | 02H (STX) | Telegramm-Start  |
| 2    | 01H       | Steuerungsadresse 1 (entspricht Variable 65101)  |
| 3    | 0BH       | Opcode "Schreibe erweiterten Merker"   |
| 4    | ??        | Low-Byte der Merker Nummer   |
| 5    | ??        | Bits 0-14: High-Byte der Merker Nummer<br>Bit 15: Neuer Zustand des Merkers (0=aus, 1=ein) |
| 6    | 03H (ETX) | Telegramm-Ende   |
| 7    | ??        | Additive 8-Bit Checksumme der Telegramm Daten  |

n Tabelle 19 – Sendetelegramm Opcode 0BH

### n Antworttelegramm

| Byte | Inhalt    | Bedeutung                                     |
|------|-----------|---|
| 1    | 06H (ACK) | Bestätigung: Sendetelegramm korrekt empfangen |

n Tabelle 20 – Antworttelegramm Opcode 0BH

## 2.8 Opcode 0CH: Lese UDB

Mit dem Befehl 0CH wird der UDB – Universal Data Block – aus der Steuerung gelesen. Dieser UDB enthält alle Informationen, die für den schnellen und effektiven direkten Zugriff auf den Speicher einer MC90/B Steuerung notwendig sind. Der Aufbau und Inhalt des UDB ist in Kapitel 4.1 - Aufbau des Universal Data Blocks (Seite 3) beschrieben.

### n Sendetelegramm

| Byte | Inhalt    | Bedeutung                                       |
|------|-----------|---|
| 1    | 02H (STX) | Telegramm-Start                                 |
| 2    | 01H       | Steuerungsadresse 1 (entspricht Variable 65101) |
| 3    | 0CH       | Opcode "Lese UDB"                               |
| 4    | 03H (ETX) | Telegramm-Ende                                  |
| 5    | ??        | Additive 8-Bit Checksumme der Telegramm Daten   |

n Tabelle 21 – Sendetelegramm Opcode 0CH

### n Antworttelegramm

| Byte   | Inhalt    | Bedeutung   |
|--------|-----------|---|
| 1      | 06H (ACK) | Bestätigung: Sendetelegramm korrekt empfangen                                       |
| 2      | 02H (STX) | Telegramm-Start   |
| 3      | 01H       | Steuerungsadresse 1 (entspricht Variable 65101)                                     |
| 4 – 67 | ??        | 64 Byte UDB Daten (siehe Kapitel 4.1 - Aufbau des Universal Data Blocks ab Seite 3) |
| 68     | 03H (ETX) | Telegramm-Ende  |
| 69     | ??        | Additive 8-Bit Checksumme der Telegramm Daten                                       |

n Tabelle 22 – Antworttelegramm Opcode 0CH

## 2.9 Opcode 0DH: Lese Daten aus Speicher

Nachdem Sie die interne Speicherstruktur mit dem Befehl "Lese UDB" (Opcode 0CH) ermittelt haben, können Sie auf die Informationen der Steuerung direkt über den Speicher zugreifen. Hierzu dient der Befehl 0DH, der direkt – ohne zusätzliche Umwandlung oder Validierung – die Daten aus dem Steuerungsspeicher übermittelt.

### n Sendetelegramm

| Byte | Inhalt    | Bedeutung  |
|------|-----------|--|
| 1    | 02H (STX) | Telegramm-Start                                  |
| 2    | 01H       | Steuerungsadresse 1 (entspricht Variable 65101)  |
| 3    | 0DH       | Opcode "Lese Daten aus Speicher"                 |
| 4    | ??        | Low-Byte der Startadresse im Speicher            |
| 5    | ??        | High-Byte der Startadresse im Speicher           |
| 6    | ??        | Länge des zu lesenden Blocks (maximal 120 Byte!) |
| 7    | 03H (ETX) | Telegramm-Ende                                   |
| 8    | ??        | Additive 8-Bit Checksumme der Telegramm Daten    |

n Tabelle 23 – Sendetelegramm Opcode 0DH

### n Antworttelegramm

| Byte  | Inhalt    | Bedeutung                                       |
|-------|-----------|---|
| 1     | 06H (ACK) | Bestätigung: Sendetelegramm korrekt empfangen   |
| 2     | 02H (STX) | Telegramm-Start                                 |
| 3     | 01H       | Steuerungsadresse 1 (entspricht Variable 65101) |
| 4 – N | ??        | Angeforderte Daten                              |
| N + 1 | 03H (ETX) | Telegramm-Ende                                  |
| N + 2 | ??        | Additive 8-Bit Checksumme der Telegramm Daten   |

n Tabelle 24 – Antworttelegramm Opcode 0DH

## 2.10 Opcode 0EH: Schreibe Daten in Speicher

Nachdem Sie die interne Speicherstruktur mit dem Befehl "Lese UDB" (Opcode 0CH) ermittelt haben, können Sie nun mit dem Befehl 0EH direkt in den Speicher der Steuerung schreiben. Bitte verwenden Sie diese Methode auf keinen Fall für bitcodierte Werte (Merker und Ausgänge), da ansonsten interne Kontrollen für veränderte Werte nicht mehr greifen können.

### n Sendetelegramm

| Byte  | Inhalt    | Bedeutung  |
|-------|-----------|--|
| 1     | 02H (STX) | Telegramm-Start                                      |
| 2     | 01H       | Steuerungsadresse 1 (entspricht Variable 65101)      |
| 3     | 0EH       | Opcode "Schreibe Daten in Speicher"                  |
| 4     | ??        | Low-Byte der Startadresse im Speicher                |
| 5     | ??        | High-Byte der Startadresse im Speicher               |
| 6     | ??        | Länge des zu schreibenden Blocks (maximal 120 Byte!) |
| 7 – N | ??        | Zu schreibender Datenblock                           |
| N + 1 | 03H (ETX) | Telegramm-Ende                                       |
| N + 2 | ??        | Additive 8-Bit Checksumme der Telegramm Daten        |

n Tabelle 25 – Sendetelegramm Opcode 0EH

### n Antworttelegramm

| Byte | Inhalt    | Bedeutung                                     |
|------|-----------|---|
| 1    | 06H (ACK) | Bestätigung: Sendetelegramm korrekt empfangen |

n Tabelle 26 – Antworttelegramm Opcode 0EH

## 2.11 Opcode 0FH: Lese Daten via MMU

Im Gegensatz zu dem Befehl "Lese Daten aus Speicher" (Opcode 0DH) ermöglicht der Befehl 0FH den Zugriff auf die vollen 256 kB der Steuerung durch direkte Programmierung der Memory Management Unit (MMU). Dieser Befehl ist hier nur der Vollständigkeit halber aufgeführt – von einer Nutzung durch eine User-Implementation wird abgeraten.

### n Sendetelegramm

| Byte | Inhalt    | Bedeutung  |
|------|-----------|--|
| 1    | 02H (STX) | Telegramm-Start                                  |
| 2    | 01H       | Steuerungsadresse 1 (entspricht Variable 65101)  |
| 3    | 0FH       | Opcode "Lese Daten via MMU"                      |
| 4    | ??        | Low-Byte der Startadresse im Speicher            |
| 5    | ??        | High-Byte der Startadresse im Speicher           |
| 6    | ??        | MMU-Page Selektor                                |
| 7    | ??        | Länge des zu lesenden Blocks (maximal 120 Byte!) |
| 8    | 03H (ETX) | Telegramm-Ende                                   |
| 9    | ??        | Additive 8-Bit Checksumme der Telegramm Daten    |

n Tabelle 27 – Sendetelegramm Opcode 0FH

### n Antworttelegramm

| Byte  | Inhalt    | Bedeutung                                       |
|-------|-----------|---|
| 1     | 06H (ACK) | Bestätigung: Sendetelegramm korrekt empfangen   |
| 2     | 02H (STX) | Telegramm-Start                                 |
| 3     | 01H       | Steuerungsadresse 1 (entspricht Variable 65101) |
| 4 – N | ??        | Angeforderte Daten                              |
| N + 1 | 03H (ETX) | Telegramm-Ende                                  |
| N + 2 | ??        | Additive 8-Bit Checksumme der Telegramm Daten   |

n Tabelle 28 – Antworttelegramm Opcode 0FH

## 2.12 Opcode 10H: Schreibe Daten via MMU

Im Gegensatz zu dem Befehl "Schreibe Daten in Speicher" (Opcode 0EH) ermöglicht der Befehl 10H den Zugriff auf die vollen 256 kB der Steuerung durch direkte Programmierung der Memory Management Unit (MMU). Dieser Befehl ist hier nur der Vollständigkeit halber aufgeführt – von einer Nutzung durch eine User-Implementation wird abgeraten.

### n Sendetelegramm

| Byte  | Inhalt    | Bedeutung  |
|-------|-----------|--|
| 1     | 02H (STX) | Telegramm-Start                                      |
| 2     | 01H       | Steuerungsadresse 1 (entspricht Variable 65101)      |
| 3     | 0EH       | Opcode "Schreibe Daten in Speicher"                  |
| 4     | ??        | Low-Byte der Startadresse im Speicher                |
| 5     | ??        | High-Byte der Startadresse im Speicher               |
| 6     | ??        | MMU-Page Selektor                                    |
| 7     | ??        | Länge des zu schreibenden Blocks (maximal 120 Byte!) |
| 8 – N | ??        | Zu schreibender Datenblock                           |
| N + 1 | 03H (ETX) | Telegramm-Ende                                       |
| N + 2 | ??        | Additive 8-Bit Checksumme der Telegramm Daten        |

n Tabelle 29 – Sendetelegramm Opcode 10H

### n Antworttelegramm

| Byte | Inhalt    | Bedeutung                                     |
|------|-----------|---|
| 1    | 06H (ACK) | Bestätigung: Sendetelegramm korrekt empfangen |

n Tabelle 30 – Antworttelegramm Opcode 10H

## Kapitel 3 Spezielle Variablen

Einige Variablen haben innerhalb der MC90 Steuerungsfamilie eine besondere Bedeutung und verändern das Verhalten der Steuerung unter Umständen in gravierender Weise. Bitte nutzen Sie die entsprechenden Funktionen mit äußerster Sorgfalt.

| Variable | Standardwert                | Beschreibbar | Bedeutung  |
|----------|-----------------------------|--------------|--|
| 65101    | 1                           | Ja           | Steuerungsadresse, siehe Kapitel 3.1 - Adress-Variable 65101 (Seite 3)             |
| 65102    | 7                           | Ja           | Baudrate, siehe Kapitel 3.2 - Baudrate-Variable 65102 (Seite 3)                    |
| 65105    | 0                           | Ja           | Fernwortungspasswort, 0=deaktiviert  |
| 65106    | Abhängig vom Betriebssystem | Nein         | Versionsnummer des in der Steuerung gespeicherten Betriebssystems in BCD-Codierung |
| 65107    | 0                           | Ja           | Befehlsvariable, siehe Kapitel 3.3 - Befehlsvariable 65107 (Seite 3)               |
| 65108    | 0                           | Ja           | Auswahl einer Task für Debugging   |
| 65109    | 1                           | Nein         | Aktuell ausgeführte Programmzeile  |
| 65490    | 1                           | Nein         | Aktuell gewählte Display-Textseite   |
| 65499    | 100                         | Nein         | Aktueller Override-Wert für die Achsen   |

n Tabelle 31 – Spezielle Variablen

Die konkrete Funktionalität einiger der in obiger Tabelle beschriebenen Variablen wird auf den nächsten Seiten näher erläutert.

## 3.1 Adress-Variable 65101

Es ist möglich, innerhalb eines Stromschleifen-Netzwerks bis zu 255 Steuerungen der MC90 Familie zusammenzuschalten. Damit in diesem Zusammenhang eine eindeutige Identifizierung der jeweils gewünschten Steuerung vorgenommen werden kann, verfügt jede einzelne Steuerung über eine Adresse. Diese Steuerungsadresse wird bei jedem Telegramm im Telegrammkopf mit übermittelt.

Durch Beschreiben der Variable 65101 kann die Steuerungsadresse verändert werden. Die Änderung tritt nach dem nächsten Reset der Steuerung in Kraft.

**Achtung!** Es wird innerhalb der Steuerung keine Kontrolle auf ungültige Werte vorgenommen.

| Wert | Baudrate   |
|------|------------|
| 7    | 38400 Baud |
| 6    | 19200 Baud |
| 4    | 9600 Baud  |

n Tabelle 32 – Baudraten in Variable 65102

Bitte beachten Sie auch die Kapitel 1.2 - Verbindungsaufbau (Seite 3) sowie 1.3 - Grundsätzlicher Protokollaufbau (Seite 3).

## 3.2 Baudrate-Variable 65102

Durch Schreiben eines Werts in die Variable 65102 wird die Baudrate für die Kommunikation der Steuerung mit dem PC ausgewählt. Jegliche Änderung in dieser Variable wird erst nach dem nächsten Reset der Steuerung aktiv.

Achtung! Es wird innerhalb der Steuerung keine Kontrolle auf ungültige Werte vorgenommen.

| Wert | Baudrate   |
|------|------------|
| 7    | 38400 Baud |
| 6    | 19200 Baud |
| 4    | 9600 Baud  |

n Tabelle 33 – Baudraten in Variable 65102

Bitte beachten Sie auch das Kapitel 1.2 - Verbindungsaufbau (Seite 3).

### 3.3 Befehlsvariable 65107

Um spezielle Aktionen innerhalb der Steuerung auszulösen, steht Ihnen eine Befehlsvariable innerhalb der Steuerung zur Verfügung: die Variable 65107. Durch Schreiben eines Wertes in diese Variable lösen Sie eine Aktion aus. Die Steuerung quittiert das Schreiben der Variable mit einem Acknowledge (siehe 2.2 - Opcode 01H: Schreibe Variable auf Seite 3) und führt die Aktion aus. Nach erfolgreichem Ausführen der Aktion wird der Wert der Variablen 65107 wieder auf 0 zurückgesetzt.

Falls die ausgelöste Aktionen weitere Daten vom PC erwartet, muss der PC abwarten, bis die Variable wieder auf 0 gesetzt wurde, bevor weitere Daten gesendet werden.

| Wert | Bedeutung   | MC90 | MC90/A | MC90/B |
|------|---|------|--------|--------|
| 001H | Starte Übertragung eines SPS-Programms im ASCII-Modus                             | Ja   | Ja     | Ja     |
| 002H | Starte Empfang eines SPS-Programms im ASCII-Modus                                 | Ja   | Ja     | Ja     |
| 003H | Starte Übertragung eines SPS-Programms im Binär-Modus                             | Ja   | Ja     | Ja     |
| 065H | Starte Übertragung eines SPS-Programms im Binär-Modus mit Checksummen-Absicherung | Nein | Nein   | Ja     |
| 066H | Beende Übertragung eines SPS-Programms im Binär-Modus mit Checksummen-Absicherung | Nein | Nein   | Ja     |
| 00FH | Führe einen Reset mit Urlöschung durch (Hardware-Reset)                           | Ja   | Ja     | Ja     |
| 010H | Führe einen Reset durch (Software-Reset)  | Ja   | Ja     | Ja     |
| 011H | Lösche das gespeicherte SPS-Programm  | Ja   | Ja     | Ja     |
| 005H | Übertrage Betriebssystem zum CPU-Modul  | Nein | Nein   | Ja     |
| 00BH | Übertrage Betriebssystem zur allen Handbedienteilen                               | Nein | Nein   | Ja     |
| 907H | Übertrage Betriebssystem zu allen Achscontrollern                                 | Nein | Nein   | Ja     |

■ Tabelle 34 – Befehlscodes in Variable 65107

Bitte beachten Sie: es wird ausdrücklich davon abgeraten, diese Funktionen aufgrund ihrer Komplexität mit in eine eigene Protokollimplementierung zu übernehmen. Sämtliche hier beschriebenen Funktionalitäten können über die Entwicklungsumgebung VMC Workbench oder über die erhältlichen Systemtreiber für alle Windows-Versionen ausgelöst werden.

# Kapitel 4 MC90/B Speicherorganisation

Bei Steuerungen des Typs MC90/B ist es möglich, Daten direkt aus dem Speicher der Steuerung zu lesen. Dies ermöglicht nicht nur eine einfachere Programmierung der Protokolle, sondern auch einen effektiveren Datentransfer: es können jeweils die Datenbereiche aus der Steuerung in einem Block gelesen werden, die für das Anwendungsprogramm relevant sind.

## n Lesen des Universal Data Block

Wesentlicher Bestandteil dieses Konzepts ist der Universal Data Block, der mit dem Befehl 0CH gelesen wird (Kapitel 2.8 - Opcode 0CH: Lese UDB auf Seite 3). In diesem Universal Data Block sind dann alle wichtigen Speicheradressen enthalten. Nicht im Universal Data Block enthaltene Speicheradressen können leicht rechnerisch ermittelt werden.

Nachdem die Speicheradresse ermittelt ist, können die entsprechenden Daten mit dem Befehl 0AH gelesen bzw. mit dem Befehl 0BH geschrieben werden (siehe auch Kapitel 2.9 - Opcode 0DH: Lese Daten aus Speicher auf Seite 3 sowie Kapitel 2.10 - Opcode 0EH: Schreibe Daten in Speicher auf Seite 3).

### Wiederholtes Lesen

Es ist nicht notwendig, den Universal Data Block mehr als einmal pro Sitzung einzulesen. Die Steuerung trägt beim Systemstart alle wichtige Informationen in diesen Datenbereich ein und verändert sie danach nicht mehr.

## 4.1 Aufbau des Universal Data Blocks

Wenn mit dem Befehl 0CH die Daten des Universal Data Blocks angefordert werden (siehe auch Kapitel 2.8 - Opcode 0CH: Lese UDB auf Seite 3), wird stets ein Datenblock von 64 Byte Größe zurückgesendet. Jede Information ist als 16-Bit Wort enthalten.

| Offset | Bedeutung                             |
|--------|---------------------------------------|
| 0000H  | Anzahl gültiger Worte im UDB          |
| 0002H  | Version des MC90/B CPU-Moduls         |
| 0004H  | Version der angeschlossenen Achsen    |
| 0006H  | Adresse der Variablendaten            |
| 0008H  | Adresse der Achsdaten                 |
| 000AH  | Adresse der Displaydaten              |
| 000CH  | Adresse der I/O-Daten und Merker      |
| 000EH  | Adresse des UDB im Speicher           |
| 0010H  | Adresse interner Flag-Daten           |
| 0012H  | Adresse der MCSave(tm) Aufzeichnungen |

n Tabelle 35 – Aufbau des Universal Data Block

Zwar zeigen diese Adressen lediglich auf interne, komplex aufgearbeitete Datenbereiche. Aufgrund dieser Informationen können Sie aber nun die Adressen aller wichtigen Daten errechnen.

| Bedeutung  | Adresse        |
|--|----------------|
| Adresse der SPS-Variablen                                    | UDB[06H]       |
| Status der digitalen Ausgänge (linear adressiert, 8 Bit)     | UDB[0CH] + 16H |
| Status der digitalen Eingänge (linear adressiert, 8 Bit)     | UDB[0CH] + 32H |
| Status der Merker von 110 bis 199 (linear adressiert, 6 Bit) | UDB[0CH] + 20H |
| Status der Merker von 300 bis 499 (linear adressiert, 6 Bit) | UDB[0CH] + 40H |
| Status der Merker von 500 bis 749 (linear adressiert, 8 Bit) | UDB[0CH] + 62H |
| Stellung des Schlüsselschalter                               | UDB[0CH] + 31H |
| Aktueller Achsen-Override                                    | UDB[0CH] + 76H |
| Diverse Systemdaten und Flags zum SPS-Programm               | UDB[10H]       |
| SPS-Timer  | UDB[10H] + 08H |
| SPS-Debugfunktionen und Zeilenanzeige (SPS-Merker 86 und 87) | UDB[10H] + 08H |
| Flag für die 12MHz-Version der CPU                           | UDB[10H] + 11H |

n Tabelle 36 – Ermittlung wichtiger Speicheradressen

## 4.2 Lesen von Daten aus dem Speicher

Wir wissen nun, wo die Informationen im Speicher der MC90/B Steuerung abgelegt sind. Um jedoch genau die Informationen zu lesen, die vom Programm benötigt werden, sind entsprechende Berechnungsroutinen erforderlich. Dies wird im Folgenden für die wichtigsten Daten exemplarisch vorgeführt.

### n Variablen lesen

Zum Lesen der Variable mit der Nummer (VarNum) gehen Sie wie folgt vor:

- n  $\text{BaseAddress} = \text{UDB}[08\text{H}]$
- n  $\text{VarNum} = \text{VarNum} - 65100$
- n  $\text{VarAddress} = \text{BaseAddress} + \text{VarNum} \ll 1$
- n Lesen eines 16-Bit Integerwertes aus dem Speicher der Steuerung ab (VarAddress) ergibt den aktuellen Wert der Variablen

### n Digitale Eingänge lesen

Zum Lesen eines digitalen Eingangs mit der Nummer (EinNum) gehen Sie wie folgt vor:

- n  $\text{BaseAddress} = \text{UDB}[0\text{CH}] + 32\text{H}$
- n  $\text{EinNum} = \text{EinNum} - 1$
- n  $\text{EinAddress} = \text{BaseAddress} + \text{EinNum} \gg 3$
- n Lesen eines Bytes aus dem Speicher der Steuerung ab (EinAddress) ergibt den Status von 8 Eingängen, in welchen auch der gewünschte Eingang enthalten ist.
- n Isolieren des Eingangs mit einer Und-Maske:  $\text{Wert} = \text{Wert} \& (1 \ll (\text{EinNum} \& 07\text{H}))$

### n Digitale Ausgänge lesen

Zum Lesen eines digitalen Ausgangs mit der Nummer (AusNum) gehen Sie wie folgt vor:

- n  $\text{BaseAddress} = \text{UDB}[0\text{CH}] + 16\text{H}$
- n  $\text{AusNum} = \text{AusNum} - 1$
- n  $\text{AusAddress} = \text{BaseAddress} + \text{AusNum} \gg 3$
- n Lesen eines Bytes aus dem Speicher der Steuerung ab (AusAddress) ergibt den Status von 8 Ausgängen, in welchen auch der gewünschte Ausgang enthalten ist.
- n Isolieren des Ausgangs mit einer Und-Maske:  $\text{Wert} = \text{Wert} \& (1 \ll (\text{AusNum} \& 07\text{H}))$

## n Merker lesen

Aufgrund der als Ganzes nicht linear angeordneten Merkerstruktur müssen beim Lesen eines Merkers verschiedene Fälle unterschieden werden:

n Merker < 200: 6 Bit Adressierung ab UDB[0CH] + 20H

n Merker >= 300 und Merker < 500: 6 Bit Adressierung ab UDB[0CH] + 40H

n Merker >= 500 und Merker < 750: 8 Bit Adressierung ab UDB[0CH] + 62H

Ist der Merker kleiner 200, gehen Sie zum Lesen eines Merkers mit der Nummer (MerNum) wie folgt vor:

n BaseAddress = UDB[0CH] + 20H

n MerNum = MerNum - 110

n MerAddress = BaseAddress + MerNum / 6

n Lesen eines Bytes aus dem Speicher der Steuerung ab (MerAdress) ergibt den Status von 6 Merkern, in welchen auch der gewünschte Merker enthalten ist.

n Isolieren des Merkers mit einer Und-Maske: Wert = Wert & (1 << (MerNum % 6))

Für die anderen Merkerbereiche gilt entsprechendes.

# Anhang A Tabellenverzeichnis

---

|   |  |   |
|---|--|---|
| n | Tabelle 1 – Übersicht Treiberbibliotheken .....                | 3 |
| n | Tabelle 2 - Schnittstellenparameter .....                      | 3 |
| n | Tabelle 3 – Grundsätzlicher Aufbau eines Sendetelegramms ..... | 3 |
| n | Tabelle 4 – Beispiel zur Checksummenberechnung .....           | 3 |
| n | Tabelle 5 – Übersicht Protokoll-Opcodes .....                  | 3 |
| n | Tabelle 6 – Sendetelegramm Opcode 00H .....                    | 3 |
| n | Tabelle 7 – Antworttelegramm Opcode 00H .....                  | 3 |
| n | Tabelle 8 – Sendetelegramm Opcode 01H .....                    | 3 |
| n | Tabelle 9 – Antworttelegramm Opcode 01H .....                  | 3 |
| n | Tabelle 10 – Sendetelegramm Opcode 06H .....                   | 3 |
| n | Tabelle 11 – Antworttelegramm Opcode 06H .....                 | 3 |
| n | Tabelle 12 – Stellung des Schlüsselschalters .....             | 3 |
| n | Tabelle 13 – Sendetelegramm Opcode 07H .....                   | 3 |
| n | Tabelle 14 – Antworttelegramm Opcode 07H .....                 | 3 |
| n | Tabelle 15 – Sendetelegramm Opcode 09H .....                   | 3 |
| n | Tabelle 16 – Antworttelegramm Opcode 09H .....                 | 3 |
| n | Tabelle 17 – Sendetelegramm Opcode 0AH .....                   | 3 |
| n | Tabelle 18 – Antworttelegramm Opcode 0AH .....                 | 3 |
| n | Tabelle 19 – Sendetelegramm Opcode 0BH .....                   | 3 |
| n | Tabelle 20 – Antworttelegramm Opcode 0BH .....                 | 3 |
| n | Tabelle 21 – Sendetelegramm Opcode 0CH .....                   | 3 |
| n | Tabelle 22 – Antworttelegramm Opcode 0CH .....                 | 3 |
| n | Tabelle 23 – Sendetelegramm Opcode 0DH .....                   | 3 |
| n | Tabelle 24 – Antworttelegramm Opcode 0DH .....                 | 3 |
| n | Tabelle 25 – Sendetelegramm Opcode 0EH .....                   | 3 |
| n | Tabelle 26 – Antworttelegramm Opcode 0EH .....                 | 3 |
| n | Tabelle 27 – Sendetelegramm Opcode 0FH .....                   | 3 |
| n | Tabelle 28 – Antworttelegramm Opcode 0FH .....                 | 3 |
| n | Tabelle 29 – Sendetelegramm Opcode 10H .....                   | 3 |
| n | Tabelle 30 – Antworttelegramm Opcode 10H .....                 | 3 |
| n | Tabelle 31 – Spezielle Variablen .....                         | 3 |
| n | Tabelle 32 – Baudraten in Variable 65102 .....                 | 3 |
| n | Tabelle 33 – Baudraten in Variable 65102 .....                 | 3 |
| n | Tabelle 34 – Befehlscodes in Variable 65107 .....              | 3 |
| n | Tabelle 35 – Aufbau des Universal Data Block .....             | 3 |
| n | Tabelle 36 – Ermittlung wichtiger Speicheradressen .....       | 3 |

## n Raum für Ihre Notizen