

Dokumentation MOCS-1100 INTERBUS-S-Software

Dok-Rev. 1.5 vom 14.11.2007

Software-Rev. 0.09 vom 16.03.1999

Inhaltsverzeichnis

1	Urheberrecht und Haftung.....	4
1.1	Handhabung	4
1.2	Erklärung	4
2	Programmierung der INTERBUS-S-Anschaltung des MOCS-1100	5
2.1	Grundlagen	5
2.1.1	Funktionsprinzip des INTERBUS-S	5
2.1.2	Topologie eines INTERBUS-S-Netzes	6
2.1.3	Geräteerkennung	6
2.1.4	Identifikation der Netztopologie	7
2.1.5	Datenverkehr über den INTERBUS-S	7
2.1.6	INTERBUS-S-Steuerung	8
2.2	Aufbau des Dual-ported Memorys	8
2.2.1	Adreßbereiche	8
2.2.2	Abbildung der I/O-Module auf den DPM-Datenbereich	8
2.2.3	Diagnosebitregister	9
2.2.4	Diagnoseparameterregister	9
2.3	Synchronisation nach Power Up oder Reset	10
3	Treibersoftware	11
3.1	Start des InterBus-S von der Bedienkonsole	11
3.1.1	Task IBS	11
3.1.2	Shellmodul IBSIO	11
3.1.3	Hilfs-Tasks	12
3.2	Grundfunktionen der Treibersoftware	12
3.2.1	Start_IBS_Setup	13
3.2.2	Add_IBS_Module	14
3.2.3	Check_and_Start_IBS	16
3.2.4	Do_IBS_IO	17
3.3	Erweiterte Treiberfunktionen	18
4	Abkürzungen.....	19

Revisionsliste:

Rev.	Datum	Na.	Änderung
1.0	02.12.1996	Ko	Übernahme
1.1	25.05.1998	Ha	Manueller Bus-Start hinzugefügt
1.2	04.04.2000	Ko	nach Software.dot konvertiert
1.3	25.06.2003	Ko	Diagnose-Register ergänzt (Kap. 2.2.3)
1.4	06.11.2003	Ko	Power Up Synchronisation ergänzt (Kap. 2.3)
1.5	15.03.2004	Ko	Zeitbedingungen beim Zugriff auf DPM spezifiziert

1 Urheberrecht und Haftung

Alle Rechte an diesen Unterlagen liegen bei der IEP GmbH, Langenhagen.

Die Vervielfältigung, auch auszugsweise, ist nur mit unserer ausdrücklichen schriftlichen Genehmigung zulässig.

In Verbindung mit dem Kauf von Software erwirbt der Käufer einfaches, nicht übertragbares Nutzungsrecht. Dieses Recht zur Nutzung bezieht sich ausschließlich darauf, daß dieses Produkt auf oder in Zusammenhang mit jeweils **einem** Computer zu benutzen ist. Das Erstellen einer Kopie ist ausschließlich zu Archivierungszwecken unter Aufsicht des Käufers oder seines Beauftragten zulässig. Der Käufer haftet für Schäden, die sich aus der Verletzung seiner Sorgfaltspflicht ergeben, z.B. bei unautorisiertem Kopieren, unberechtigter Weitergabe der Software usw.. Der Käufer gibt mit dem Erwerb der Software seine Zustimmung zu den genannten Bedingungen. Bei unlizensiertem Kopieren muß vorbehaltlich einer endgültigen juristischen Klärung von Diebstahl ausgegangen werden. Dies gilt ebenso für Dokumentation und Software, die durch Modifikation aus Unterlagen und Programmen von IEP hervorgegangen ist, gleichgültig, ob die Änderungen als geringfügig oder erheblich anzusehen sind.

Eine Haftung seitens IEP für Schäden, die auf den Gebrauch von Software, Hardware oder Benutzung dieses Manuskriptes zurückzuführen sind, wird ausdrücklich ausgeschlossen, auch für den Fall fehlerhafter Software oder irrtümlicher Angaben.

Das Einverständnis des Käufers oder Nutzers für den Haftungsausschluß gilt mit dem Kauf und der Nutzung der Software und dieser Unterlagen als erteilt.

1.1 Handhabung

Lesen Sie bitte zuerst sorgfältig diese Dokumentation bevor Sie anfangen zu programmieren. Sie sparen Zeit und vermeiden Probleme.

1.2 Erklärung

Wir behalten uns das Recht vor, Änderungen, die einer Verbesserung der Schaltung oder des Produktes dienen, ohne besondere Hinweise vorzunehmen. Trotz sorgfältiger Kontrolle kann für die Richtigkeit der hier gegebenen Daten, Schaltpläne, Programme und Beschreibungen keine Haftung übernommen werden. Die Eignung des Produktes für einen bestimmten Einsatzzweck wird nicht zugesichert.

2 Programmierung der INTERBUS-S-Anschaltung des MOCS-1100

Diese Dokumentation beschreibt die Nutzung der INTERBUS-S-Anschaltung des MOCS-1100-Systems für PEARL-Programmierer.

Die Dokumentation ist gültig für Treibersoftware der Version 0.09.

2.1 Grundlagen

Die Anschaltung der INTERBUS-S-Module an das MOCS-1100 Grundgerät erfolgt über eine intelligente Baugruppe, die als Master auf dem INTERBUS-S fungiert. Diese Baugruppe wird vom MOCS-1100 über ein Dual-ported Memory (DPM) angesprochen.

2.1.1 Funktionsprinzip des INTERBUS-S

Aus Anwendersicht kann der INTERBUS-S als ein großes Schieberegister betrachtet werden. Bei aktivem Bus werden in den sogenannten Buszyklen Daten durch alle angeschlossenen Teilnehmer geschoben, jeder Teilnehmer entnimmt dem Datenstrom die ihm durch seine Einordnung in die Netztopologie zugeordneten Ausgabedaten und fügt seine Eingabedaten entsprechend in den Datenstrom ein.

Für die physikalische Ausprägung des Netzes können zwei Transportschichten realisiert werden:

- Fernbus

Mit einer RS422-Übertragungsstrecke wird das Rückgrat des Netzes gebildet. Der Fernbus ist zur Überbrückung größerer Entfernungen geeignet.

- Localbus

Mit Hilfe sogenannter Busklemmen (BK's) können Abzweige aus dem Fernbus-Rückgrat realisiert werden. An eine Busklemme können I/O-Module als Stich angeschlossenen werden. Der Stich wird hierbei in das angenommene, große Schieberegister eingeblendet.

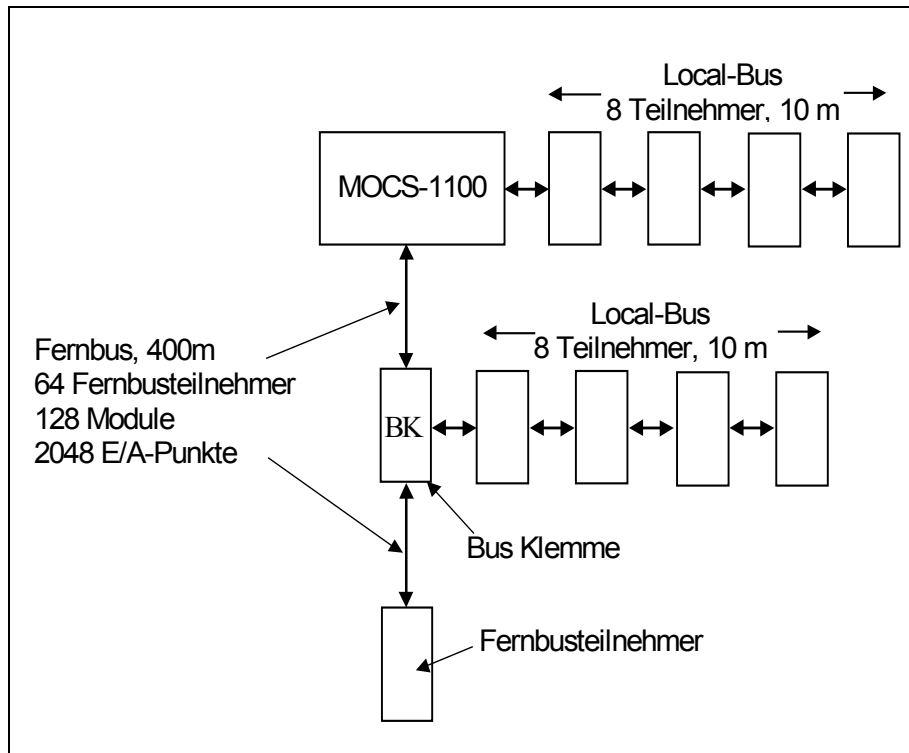
Der Localbus wird mit einer mehrpoligen, nichtdifferentiellen Übertragungsstrecke realisiert.

Daneben existierende Sonderformen (Installationsfernbus, Fernbus-Stichleitungen) sollen hier nicht betrachtet werden.

Jeder einzelne Busteilnehmer besitzt einen Busein- sowie einen Busausgang. Durch eine Brücke im Buskabel wird jedem Teilnehmer an seinem Ausgang signalisiert, ob ihm noch weitere Teilnehmer in seinem Bussegment folgen.

2.1.2 Topologie eines INTERBUS-S-Netzes

Die INTERBUS-S-Baugruppe des MOCS-1100 stellt sowohl einen Fern- als auch einen Localbus zur Verfügung. Folgende Netztopologie ist somit realisierbar:



Intern ist der INTERBUS-S-Anschluß in Form einer Masterbaugruppe (MAB/T) mit Original-Phoenix-Software und integrierter Busklemme (BK) realisiert. Die Busklemme ist hierbei als erster Fernbus-Teilnehmer an die Masterbaugruppe angeschlossen.

2.1.3 Geräteerkennung

Jedes INTERBUS-S-Modul ist über eine Identifikationsnummer (ID) gekennzeichnet, die eine funktionelle Beschreibung des Moduls darstellt. Die ID wird vom Hersteller festgelegt und ist üblicherweise sowohl in der technischen Dokumentation als auch als Aufdruck auf jedem Modul zu finden. Die ID kennzeichnet insbesondere das Verhalten des Moduls auf dem INTERBUS-S (Datenbreite etc.).

Die ID's stellen nur eine sehr grobe, funktionelle Einordnung dar. Trotz gleicher ID können verschiedene Module auch unterschiedliche Funktionalitäten bieten.

Ohne Kenntnis der ID's aller eingesetzten Module ist die Programmierung eines INTERBUS-S-Systems nicht möglich.

Die ID's einiger beispielhafter Module lauten:

Modul	Funktion	ID (hexadezimal)
BK	Busklemme	0x0034
DIO8	Digital-I/O, 8 Bit	0x0183
AI3	Analog-In	0x01CF

2.1.4 Identifikation der Netztopologie

Der INTERBUS-S-Master erfaßt die aktuelle Netztopologie während einer Konfigurationsphase. Er erstellt hierbei eine Liste aller in einem INTERBUS-S-Netz enthaltenen Modul-ID's.

Die Konfigurationsphase wird bei angehaltenem INTERBUS-S auf Programmanforderung hin durchgeführt. Der Master führt spezielle Konfigurationszyklen durch, in denen er die ID's aller angeschlossenen erhält und als Liste im DPM ablegt.

2.1.5 Datenverkehr über den INTERBUS-S

Nach Ablauf der Konfigurationsphase ist dem INTERBUS-S-Master die aktuelle Buskonfiguration bekannt. Der Master bildet die Datenbereiche aller angeschlossenen Module in Speicherbereiche des DPM ab.

Programme können Ihre Eingabedaten dem DPM entnehmen sowie dort ihre Ausgabedaten ablegen. Der Datenverkehr zwischen DPM und INTERBUS-S-Modulen läuft hierbei zweistufig ab:

1. INTERBUS-S-Zyklen

Nach einmaligem Starten der INTERBUS-S-Datenzyklen tauscht der Master zyklisch Daten aus einem lokalen Speicherbereich mit den INTERBUS-S-Modulen aus. Die Zyklusdauer bzw. -frequenz hängen vom Busausbau ab und liegen in der Größenordnung von [ms] bzw. [kHz].

Nach dem programmgesteuerten Start der Datenzyklen bleibt diese Datenübertragung bis zu einem programmgesteuerten Stop oder dem Auftreten eines Busfehlers aktiv.

2. DPM-Interface

Zur Sicherstellung der Übertragung nur konsistenter Datensätze tauscht der Master Daten zwischen DPM und lokalem Speicher nur auf Anforderung durch das Anwenderprogramm aus. Dem Anwenderprogramm wird hierbei auch mitgeteilt, wann wieder ein konsistenter Satz von Eingangsdaten im DPM vorhanden ist.

Bitte beachten Sie die folgenden zeitlichen Zusammenhänge:

Nach dem Anstoßen des DPM-Zyklus (Schreiben des Sync-Registers mit \$AA55) werden die Ausgangsdaten vom DPM in den internen Speicher übernommen und im nächsten INTERBUS-S-Zyklus übertragen. Die Eingangsdaten dieses Zyklus werden dann in das DPM übertragen und der Zyklus ist beendet (Sync-Register auf \$55AA). Die aktuellen Eingangsdaten können vom Anwendungsprogramm ausgewertet werden. **Nun darf nicht sofort ein neuer DPM-Zyklus gestartet werden, da der INTERBUS-S-Master durcheinander kommen kann! Es müssen bis zum Start des nächsten Zyklus min. 8 ms Pause gemacht werden!**



Mit Hilfe dieser zweistufigen Realisierung kann in den Fällen, in denen sichergestellt werden muß, daß zusammenhängende Daten auch zur gleichen Zeit auf dem INTERBUS-S ausgegeben werden, eine einfache Synchronisation zwischen den INTERBUS-S-Zyklen und dem Datenaustausch des Anwenderprogramms mit dem DPM erfolgen.

2.1.6 INTERBUS-S-Steuerung

Die Kommunikation zwischen Anwenderprogramm bzw. Treiberrouinen und INTERBUS-S-Master erfolgt ebenfalls über das DPM.

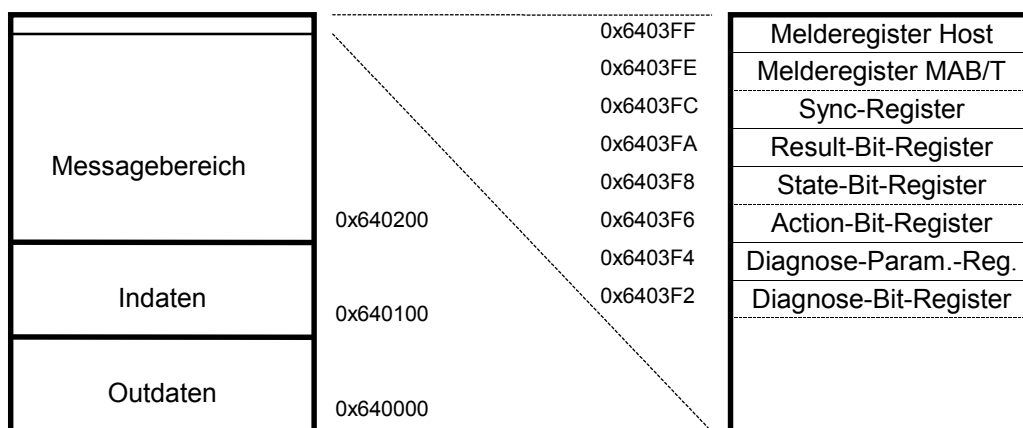
Auf die Details dieser Kommunikation soll hier nicht eingegangen werden, da sie größtenteils von den Treiberrouinen übernommen wird.

2.2 Aufbau des Dual-ported Memorys

2.2.1 Adreßbereiche

Das DPM als Kernstück der Kommunikation zwischen Anwenderprogramm sowie Treiberrouinen und dem INTERBUS-S-Master hat für bestimmte Zwecke 4 fest definierte Adreßbereiche:

1. INTERBUS-S-Outdaten (0x640000 bis 0x6400FF)
2. INTERBUS-S-Indaten (0x640100 bis 0x6401FF)
3. Messagebereich (0x640200 bis 0x6402FF)
4. Registerbereich



Der Datenaustausch zwischen Anwenderprogramm und INTERBUS-S erfolgt über die Datenbereiche. Der Messagebereich ist hier nur zur Information wiedergegeben.

2.2.2 Abbildung der I/O-Module auf den DPM-Datenbereich

Die Zuordnung des Datenbereiches zu den einzelnen I/O-Modulen ist von der Netztopologie abhängig und wird von der Treibersoftware bzw. der MAB/T-Firmware vorgegeben.

Zur manuellen Bestimmung des Datenabbilds im DPM ist für den In- und den Outdatenbereich jeweils eine Liste aller angeschlossenen INTERBUS-S-Module gemäß folgendem Vorgehen zu erstellen:

1. Beginne beim Master
 2. Verfolge den Fernbusstrang, das „Bus-Rückgrat“
 3. Bei einer Busklemme verfolge zunächst den Localbus, dann den weiterführenden Fernbus.
-

Während des Verfolgens der Netztopologie sind alle überlaufenen Module mit ihrer Datenbreite im In- bzw. Outdatenbereich zu notieren. Bei einer Konfiguration mit dem MOCS-1100 muß diese Liste also stets mit einer Busklemme, gefolgt von einer Liste aller an den Localbus des MOCS-1100 angeschlossenen Module, beginnen.

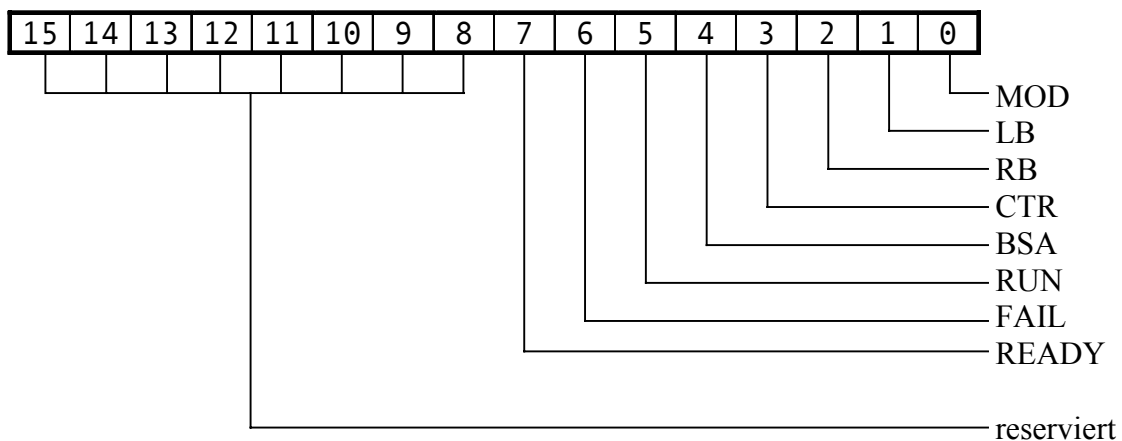
Die Abbildung der so erhaltenen Offsets auf DPM-Adressen erfolgt linear, d.h. der Offset der Moduldaten eines Moduls im In- bzw. Outdatenbereich des DPM ergibt sich aus der Summe aller von vorhergehenden Modulen belegten Daten. Die Aufsummation muß in der Reihenfolge des Durchlaufens der Netztopologie gemäß obiger Vorschrift erfolgen.

Die so erhaltenen Adressen können im Anwenderprogramm in Form von BU-Datenstationen oder über Pointerzugriffe verwendet werden.

Dieses Vorgehen wird von der Treibersoftware automatisiert.

2.2.3 Diagnosebitregister

Das Diagnosebitregister (0x6403F2) enthält Informationen über den augenblicklichen Zustand des Interbus-S. Es hat folgenden Aufbau:



Die Bits haben folgende Bedeutung:

- | | |
|-------|-----------------------------|
| MOD | - Modulfehler |
| LB | - Lokalbusfehler |
| RB | - Fernbusfehler |
| CTR | - Fehler des Masters |
| BSA | - Bussegment abgeschaltet |
| RUN | - Interbus-S Zyklen laufen |
| FAIL | - Fehler des Hostsystems |
| READY | - Master ist betriebsbereit |

2.2.4 Diagnoseparameterregister

In den oberen 8 Bit steht die Anzahl der am Interbus-S angeschlossenen Module. Im unteren Byte steht im Fehlerfall (RB-, LB-, MOD- oder CTR-Bit im Diagnosebitregister = 1) ein den Fehler näher spezifizierender Fehlercode.

Über eine Änderung im Diagnosebit- oder Diagnoseparameterregister informiert der Master durch ein gesetztes Bit 3 im Melderegister MAB/T (0x6403FE).

2.3 Synchronisation nach Power Up oder Reset

Der InterBus-S-Master führt zuerst einen Selbsttest durch. Das Melderegister wird nicht initialisiert, da damit sonst ein IR ausgelöst werden könnte. Durch Setzen des Bits READY im Diagnosebitregister sowie durch Setzen von Bit 3 im Melderegister MAB/T meldet sich der Master ready. Ab diesem Zeitpunkt ist er bereit Kommandos entgegen zu nehmen.

Um sicher zu erkennen, dass der Master betriebsbereit ist, sollte folgendes Vorgehen gewählt werden, wenn keine Interrupts verwendet werden:

1. Lösche das Diagnosebitregister
2. Warte bis READY im Diagnosebitregister gesetzt und Bit 3 im Melderegister MAB/T gesetzt
3. Beginne mit dem 1. Kommando...

Das Diagnosebitregister wird vom Master immer wieder neu geschrieben, auch wenn der Master seinen Selbsttest schon beendet hat, kann das Register gelöscht werden.

Wird der Master mit Interrupts betrieben, ist er nach dem ersten IR betriebsbereit, wenn das READY-Bit im Diagnosebitregister gesetzt ist.

3 Treibersoftware

Die Treibersoftware besteht aus dem Modul **IBS**, das sowohl Grundfunktionen als auch erweiterte Funktionen zur Verfügung stellt, sowie einer Datei **IBS.H**, die bei Einsatz eines Preprozessors als Include-Datei verwendet werden kann und die *GLOBAL*-Spezifikationen für das Modul sowie einige Definitionen für Konstanten enthält. Ein Minimalprogramm **IBSTST.P** ist als Anwendungsbeispiel beigelegt.

Das Modul **IBS** muß vor Benutzung geladen werden. Alle in diesem Modul enthaltenen Extern-Definitionen sind *GLOBAL* vereinbart und mit der */*+G*/*-Option des PEARL-Compilers zur Verfügung gestellt. Auf Grund dieses Verfahrens sind beim Einsatz des Treibers folgende Punkte zu beachten:

1. **IBS** muß vor allen Modulen, die auf die Treiberrouinen zugreifen wollen, geladen werden.
2. Soll **IBS** ggf. entladen werden, so müssen alle Module, die auf Routinen von **IBS** zugreifen, ebenfalls entladen werden, da sie Referenzen auf nicht mehr vorhandene Symbole besitzen.
3. Soll **IBS** entladen und neu geladen werden, so müssen weiterhin alle anderen Programme, die auf dem gleichen Rechner eingesetzt werden, mit der *SC*-Option des PEARL-Compilers übersetzt werden. Andernfalls besteht die Gefahr, daß der Lader des Betriebssystems uninitialisierte Daten als Kennung für globale Symbole betrachtet und fehlerhafte Bindungen vornimmt. Dies kann zu sehr diffusen Fehlerbildern bis hin zum Systemabsturz führen.

3.1 Start des InterBus-S von der Bedienkonsole

Die Treibersoftware stellt Tasks und Shell-Module zur Verfügung, die einen manuellen Start des InterBus-S sowie eine eingeschränkte Busdiagnose ermöglichen.

Im Gegensatz zum programmgesteuerten Bus-Start wird der Bus hierbei mit seiner aktuellen Ist-Konfiguration gestartet. Die aktuelle Konfiguration wird auf der Konsole ausgegeben.

3.1.1 Task IBS

Die Task **IBS** führt einen Neustart des InterBus-S durch. Ein ggf. laufender Bus wird angehalten, ggf. aufgelaufene Fehlermeldungen im Bus-Status werden gelöscht und ein Neustart des Busses mit der aktuellen Konfiguration durchgeführt.

Vor dem Start des Busses wird die Buskonfiguration in Tabellenform mit Belegung des In- und Out-Bereiches des DPM auf der Konsole ausgegeben.

Die Übertragung der Daten zwischen InterBus-S und DPM wird nicht beeinflusst.

3.1.2 Shellmodul IBSIO

Das Shell-Modul **IBSIO** veranlaßt die zyklische Übertragung der Daten zwischen InterBus-S und DPM.

Parameter: *zyklusdauer* in ms.

Die zulässige *zyklusdauer* liegt im Bereich von 0...1000. Der Wert 0 kennzeichnet eine einmalige Übertragung.

Nach dem Aufruf

IBSIO 10

erfolgt der Datentransfer zwischen DPM und InterBus-S zyklisch alle 10 Millisekunden.

Mit dem Aufruf

IBSIO 0

wird die `zyklusdauer` auf 0 gesetzt. Es findet noch genau ein Datentransfer zwischen DPM und InterBus-S statt. Eine ggf. vorher gestartete zyklische Übertragung wird abgebrochen.

3.1.3 Hilfs-Tasks

IBSVERSION:	Gibt die Version der Treibersoftware aus
IBSDOIO	Gibt die ersten 16 Byte des In- und Out-Datenbereiches im DPM aus.
IBSMEM	Gibt die Konfigurationsliste aus
IBSPLAIN	Gibt die aktuelle InterBus-S-Konfiguration aus
IBSREG	Gibt Diagnoseregister des InterBus-S aus

3.2 Grundfunktionen der Treibersoftware

Die Treibersoftware ermöglicht den Start des InterBus-S von einem Programm her. Hierbei werden zwei wesentliche Funktionalitäten zur Verfügung gestellt:

- Überprüfung der aktuellen InterBus-S-Konfiguration

Da der InterBus-S in zahlreichen verschiedenen Konfigurationen betrieben werden kann, ist eine programmgesteuerte Überprüfung der aktuellen Konfiguration mit der vom Programm erwarteten Konfiguration sinnvoll.

Die Treibersoftware kann zu diesem Zweck eine Konfigurationsliste verwalten, die mit der tatsächlichen Buskonfiguration verglichen werden kann. Das Vorgehen hierzu ist wie folgt:

1. Initialisierung der Konfigurationsliste mit **Start_IBS_Setup**
2. Modulweises Hinzufügen der erwarteten Busteilnehmer mit **Add_IBS_Module**
3. Vergleich von Ist- und Sollkonfiguration mit **Check_and_Start_IBS**

Stimmen beide Konfigurationen überein, so wird hiermit gleichzeitig der Bus gestartet. *Damit ist die Übertragung von und zum DPM noch nicht gestartet !* Dies muß mit `Do_IBS_IO` erfolgen.

- Zuordnung des I/O-Adressbereiches (DPM) zu den angeschlossenen Modulen

Die Abbildung des Datenbereiches der einzelnen Module auf I/O-Adressen im DPM hängt von der Buskonfiguration ab. Diese Abbildung ist für jede Buskonfiguration konstant. Grundsätzlich können die einzelnen I/O's daher bei einer festliegenden Buskonfiguration über feste Adressen angesprochen werden.

Die Prozedur **Add_IBS_Module** gibt für jedes in die Konfigurationsliste eingetragene Modul sowohl die In- wie auch die Out-Adresse im DPM zurück. Mit diesen Information kann auf die eigene Berechnung der I/O-Adressen verzichtet werden. Das Programm kann sich automatisch der Buskonfiguration anpassen.

IBS stellt folgende Prozeduren als Grundfunktionalität zur Verfügung. Die Beschreibung der Funktionen erfolgt hier in der Reihenfolge, in der sie im Anwenderprogramm einzusetzen sind.

3.2.1 Start_IBS_Setup

Spezifikation: **SPC Start_IBS_Setup ENTRY GLOBAL;**

Parameter: keine

Funktion: **Start_IBS_Setup** initialisiert eine vom Treiber verwendete Konfigurationliste und bereitet die Definition einer INTERBUS-S- Konfiguration vor.
Start_IBS_Setup muß einmalig vor weiteren Aufrufen von Prozeduren des INTERBUS-S-Treibers aufgerufen werden.

Start_IBS_Setup dient nur zur Initialisierung einer vom Treiber gehaltenen Konfigurationsliste.

Mit dem Aufruf von **Start_IBS_Setup** wird eine eventuell vorher gehaltene Konfigurationsliste ungültig.

Start_IBS_Setup beeinflusst die eventuell gerade laufenden Aktivitäten auf dem INTERBUS-S nicht, d.h. eventuell laufende Datenzyklen werden nicht abgebrochen und der INTERBUS-S wird nicht angehalten.

Der mehrmalige Aufruf von **Start_IBS_Setup** ist möglich.

Die von **Add_IBS_Module** zurückgegebenen Adressen dürfen nur dann als Datenbereichsadressen verwendet werden, wenn das betroffene Modul tatsächlich einen entsprechenden Datenbereich besitzt. In allen anderen Fällen (Modul hat keinen, oder nur einen In- oder nur einen Outdatenbereich) muß **Add_IBS_Module** zwar aufgerufen werden, die Rückgabewerte für die nicht genutzten Datenbereiche können aber ignoriert werden.

Hierbei ist zu beachten, das es Module gibt, die nur In- oder nur Outdaten besitzen und daher auch nur in einem der Datenbereiche berücksichtigt werden.

Die vom Treiber gehaltene Konfigurationsliste muß vor den Aufrufen von **Add_IBS_Module** mit **Start_IBS_Setup** initialisiert werden. Ein Aufruf von **Start_IBS_Setup** zwischen **Add_IBS_Module** löscht schon vorgenommene Einträge in die Konfigurationsliste und sollte von daher unterbleiben.

Add_IBS_Module führt keinen Vergleich der übergebenen Konfiguration mit der tatsächlich vorliegenden Konfiguration durch.

Fehlersituationen:

noch nicht realisiert.

3.3 Erweiterte Treiberfunktionen

IBS stellt neben den genannten Prozeduren und Tasks weitere Dienste, insbesondere weitere Tasks zur genaueren Steuerung und Beobachtung des INTERBUS-S zur Verfügung.

Diese werden erst in zukünftigen Dokumentationen beschrieben.

4 Abkürzungen

DPM	Dual-Ported Memory
BK	Busklemme (Anschluß eines Localbus-Segmentes an den Fernbus)
ID	Geräteerkennung eines INTERBUS-S-Moduls
MAB/T	INTERBUS-S-Master-Baugruppe