

Mehrprozessorkommunikation über die serielle Schnittstelle

Variante 1

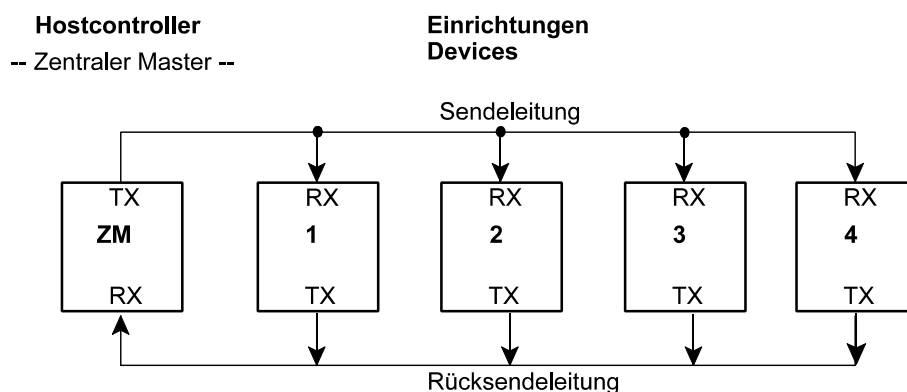
Zweidrahtbetrieb mit Zeichenübertragung (8 Bits)

Kurzbeschreibung

Stand: 1.1 vom 6. 8. 2012

1. Systemkonfiguration

Ein Mehrprozessorsystem besteht aus einem Hostcontroller (zentraler Master, Kommandogerät) und Einrichtungen (Devices). Alle Funktionseinheiten sind an zwei Leitungen angeschlossen, an eine Sendeleitung des Hostcontrollers und an eine Rücksendeleitung. Hierüber wird eine Multimasterkommunikation zwischen den Einrichtungen organisiert. Jede Einrichtung kann jede andere ansprechen und Schreib- oder Lesezugriffe ausführen. Die Zugriffsvermittlung ist Aufgabe des Hostcontrollers. Diese Funktionen werden programmseitig implementiert. Die Programmorganisation beruht auf einer Hauptsteuerschleife (Main Event Loop) und einem Unterbrechungsbehandler. Die Hauptsteuerschleife läuft im sogenannten Grundzustand. Die Einrichtungen senden im Grundzustand und empfangen im Unterbrechungszustand. Die eigentlichen Anwendungsprogramme müssen sich in die Programmorganisation des Grundzustands einfügen¹. Der Hostcontroller steuert die Vermittlung der Signalwege. Wenn eine Einrichtung senden will, muß sie die Busherrschaft anfordern und zum aktuellen Master werden. Der Hostcontroller fragt zyklisch alle Einrichtungen ab, ob sie die Busherrschaft anfordern oder nicht². Er empfängt die Zeichen, die die jeweilige Mastereinrichtung über die Rücksendeleitung sendet und reicht sie über die Sendeleitung weiter.



1: Prinzip wie Windows, nur einfacher. Keine Lizenzierung, keine Viren, keine Trojaner...

2: Ähnlich USB, nur einfacher...

Der Hostcontroller kann immer senden, was er will und alle Einrichtungen als Slaves ansprechen. Er kann nur dann Daten erhalten, wenn eine Einrichtung die Busherrschaft hat.

Die Einrichtungen sehen nur das, was der Hostcontroller sendet. Alle Einrichtungen sehen diese Zeichen gleichzeitig.

Der Hostcontroller kann anwendungsspezifische Aufgaben erledigen, vorausgesetzt, er ist in der Lage, die Einrichtungen in hinreichend kurzen Abständen abzufragen und das Weiterreichen der übertragenen Zeichen zu unterstützen.

Senden und Empfangen sind unabhängige Vorgänge, die programmseitig zu implementieren sind. Jede Einrichtung kann von sich aus nur senden. Ein Empfangen mit sofortigem Zurückliefern – also ein Lesen im eigentlichen Sinne (wie beispielsweise aus einem RAM) – ist nicht möglich. Das Lesen muß in zwei Schreibvorgänge aufgeteilt werden (Split Transactions):

1. Schreiben der Lesanforderung.
2. Zurücksenden der Lesedaten.

Auch eine Handshaking ist nur auf diese Weise möglich:

1. Schreiben des Auftrages (Kommando, Daten usw.).
2. Zurücksenden der Signalisierung (o.k. oder Fehler).

Hierfür ist ein Rollentausch zwischen Master und Slave vorgesehen. Der Master wird zum Slave, der Slave zum Master (turnaround).

Eine grundsätzliche Forderung:

Es muß schnell gehen. Die Datenübertragung ist der langsamste Vorgang. Bei 16 MHz dauert ein Befehl 62,5 ns. Bei 115 kBits/s dauert die Übertragung eines Bytes 87 µs. In dieser Zeit führt der Prozessor maximal 1392 Befehle aus. Deshalb mit so wenig Steuerzeichen wie möglich auskommen.

2. Datenübertragung

Die Datenübertragung beruht auf dem ASCII-Code. Binäre Daten können beispielsweise hexadezimal übertragen werden. Einige ASCII-Steuerzeichen werden zur Übertragungssteuerung verwendet (reservierte Steuerzeichen). Sie dürfen nicht im Datenstrom vorkommen. Alle anderen ASCII-Steuerzeichen sind im Datenstrom zugelassen. Es wurde Wert darauf gelegt, die typischen Umschaltzeichen (SOH, STX, ESC usw.) nicht zu Steuerzwecken auszunutzen.

Nur diese Zeichen werden ausgewertet (reservierte Steuerzeichen):

Zeichen	Code	Wirkung in der Einrichtung
CR	0DH	Allgemeines Abschlußzeichen. Pufferinhalt auswerten, danach Puffer löschen
CAN	18H	Puffer löschen. Sonst nichts tun.
EOT	04H	Übertragung beenden. Master aus. Slave aus. Slave wertet ggf. Pufferinhalt aus. Danach Puffer löschen.
ENQ	05H	Master und Slave tauschen ihre Rollen (Turnaround.) Der ursprüngliche Slave wertet ggf. den Pufferinhalt aus. Danach Puffer löschen.
ETB	17H	Trennen vom aktuellen Slave. Master bleibt.
EM	19H	Eine anhängige Verbindung wird zwangsweise getrennt (Interfacetrennen). Die folgenden Bytes laufen in einen Diagnosepuffer ein. Alle Einrichtungen reagieren auf EM.
BEL	07H	Startzeichen für Slavezugriff auf alle Einrichtungen (Broadcast).
ACK	06H	Positive Antwort auf Vermittlungsanfrage (Masteranforderung anhängig) sowie Trennbestätigung des Slaves
NAK	15H	Negative Antwort auf Vermittlungsanfrage (keine Masteranforderung anhängig)

Zeichen	Code	Auswertung/Wirkung im Hostcontroller
CR	0DH	Wird nur ausgewertet, wenn der Hostcontroller selbst Slave ist. Sonst weitergeben.
CAN	18H	Puffer löschen. Empfangenes CAN immer weitergeben.
EOT	04H	Immer weitergeben. Alle Einrichtungen müssen EOT zu sehen bekommen, um zu erkennen, daß der Übertragungsablauf zu Ende ist und daß sie die ankommenden Zeichen auswerten müssen. Wird intern nur ausgewertet, wenn der Hostcontroller selbst Slave ist.
ENQ	05H	Wird nur ausgewertet, wenn der Hostcontroller selbst Slave ist. Schaltet dann den Slavezustand aus und wirkt der anderen Einrichtung gegenüber als Master. Sonst weitergeben.
ETB	17H	Wird nur ausgewertet, wenn der Hostcontroller selbst Slave ist. Schaltet dann den Slavezustand aus. Sonst weitergeben
EM	19H	Verbindung zwangsweise trennen (Interfacetrennen). Wird nur vom Hostcontroller gesendet.
ACK	06H	Positive Antwort auf Vermittlungsanfrage oder Trennbestätigung vom Slave. Wird nur vom Hostcontroller empfangen und ausgewertet.
NAK	15H	Negative Antwort auf Vermittlungsanfrage. Wird nur vom Hostcontroller empfangen und ausgewertet.
BEL	07H	Wenn erstes Zeichen im Datenstrom, dann Slavezugriff auf alle Einrichtungen (Broadcast). Wird nur vom Hostcontroller gesendet.

3. Abfrage- und Auswahlkommandos

Abfrage- und Auswahlkommandos werden dann ausgewertet, wenn noch keine endgültige Verbindung aufgebaut ist. Eine Verbindung ist dann aufgebaut, wenn Master und Slave zugewiesen sind. Ein solches Kommando umfaßt zwei Zeichen und ein abschließendes CR. Das erste Zeichen kennzeichnet die Funktion, das zweite die Adresse der Einrichtung.

Adressen:

- Die Adresse 00H darf nicht vergeben werden (reserviert für Diagnosezwecke u. dergl.).
- Die Adresse FFH ist die Adresse des Hostcontrollers.
- Die Bitmuster der reservierten Steuerzeichen CR, CAN, EOT, ENQ, ETB, EM, ACK, NAK, BEL dürfen nicht vergeben werden.
- Demzufolge sind $256 - 11 = 244$ Einrichtungsadressen zugelassen. Diese Adressen müssen fest zugeteilt werden (Speicherung in Flash oder EEPROM).

Die Kommandos:

Kennbuchstabe	Folge	Wirkung
r	r @ CR	Abfrage der Masteranforderung mit anschließender Freigabe
g	g @ CR	Abfrage der Masteranforderung mit anschließender Zuweisung der Busherrschaft (Einrichtung wird Master)
m	m @ CR	Zuweisung der Busherrschaft (Einrichtung wird Master)
s	s @ CR	Auswahl des Slaves

@ = Adresse.

r - sense request & release

Die ausgewählte Einrichtung schaltet sich auf die Rücksendeleitung auf. Ist keine Masteranforderung anhängig, sendet sie NAK und gibt die Rücksendeleitung wieder frei. Ist eine Masteranforderung anhängig, sendet sie ACK und gibt die Rücksendeleitung wieder frei.

g - sense request & grant

Die ausgewählte Einrichtung schaltet sich auf die Rücksendeleitung auf. Ist keine Masteranforderung anhängig, sendet sie NAK und gibt die Rücksendeleitung wieder frei. Ist eine Masteranforderung anhängig, sendet sie ACK und bleibt auf der Rücksendeleitung aufgeschaltet. Die Einrichtung ist jetzt zum Master geworden und beginnt damit, die Masterzugriffe (Übertragungen über die Rücksendeleitung) auszuführen.

m - master grant

Die ausgewählte Einrichtung schaltet sich auf die Rücksendeleitung auf. Die Einrichtung ist jetzt zum Master geworden und beginnt damit, die Masterzugriffe (Übertragungen über die Rücksendeleitung) auszuführen.

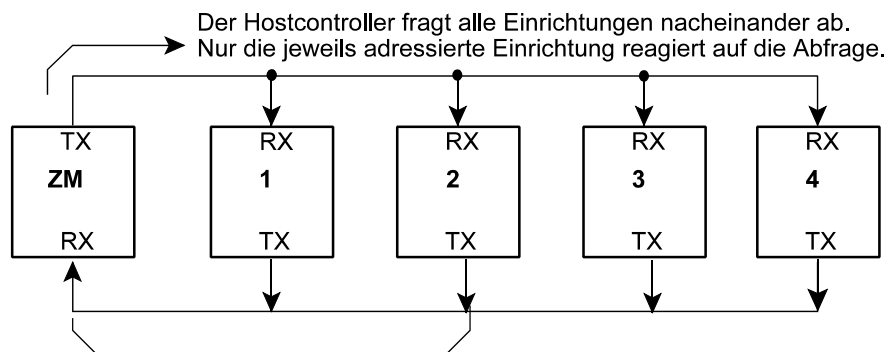
s - slave select

Die ausgewählte Einrichtung wird zum aktuellen Slave und sieht sich somit als Ziel der nachfolgenden Datenübertragungen.

r, s, und m werden nur vom Hostcontroller gegeben, s vom aktuellen Master. m darf nur dann gegeben werden, wenn die Einrichtung eine Masteranforderung anhängig hat.

r und m ermöglichen es dem Hostcontroller, die Anforderungen aller Einrichtungen abzufragen und dann zu entscheiden, welche Einrichtung die Busherrschaft erhält.

s darf nur dann gegeben werden, wenn es nicht schon einen Slave im System gibt.

4. Die Übertragungsvorgänge**Das Anfordern der Busherrschaft**

Der Hostcontroller fragt alle Einrichtungen nacheinander ab. Nur die jeweils adressierte Einrichtung reagiert auf die Abfrage.

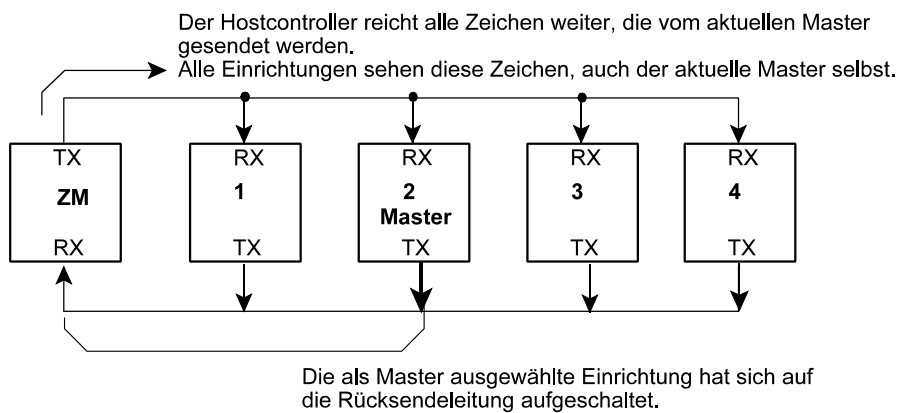
- ACK wird gesendet, wenn eine Masteranforderung anhängig ist.
- NAK wird gesendet, wenn keine Masteranforderung anhängig ist. Dann wird die Rücksendeleitung wieder freigegeben.

Die Antwort geht nur zum Hostcontroller. Die Einrichtungen bekommen sie nicht zu sehen.

Abfragekommandos:

- r @ CR. Nach Antwort mit ACK wird die Rücksendeleitung wieder freigegeben. Der Hostcontroller muß den Master mit Kommando m @ CR auswählen. Anwendung: dann, wenn der Hostcontroller alle anhängigen Anforderungen erfassen und dann erst über die Zuweisung der Busherrschaft entscheiden will.
- g @ CR. Hat die Einrichtung mit ACK geantwortet, so wird sie automatisch zum Master. Die Rücksendeleitung wird nicht freigegeben.

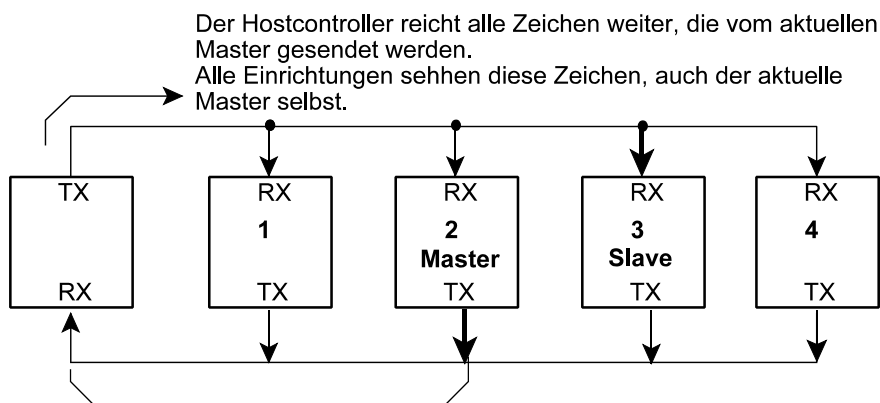
Der Master ist ausgewählt



Die als Master ausgewählte Einrichtung hat sich auf die Rücksendeleitung aufgeschaltet.

Der Hostcontroller reicht alle Zeichen weiter, die vom aktuellen Master gesendet werden. Alle Einrichtungen sehen diese Zeichen, auch der aktuelle Master selbst.

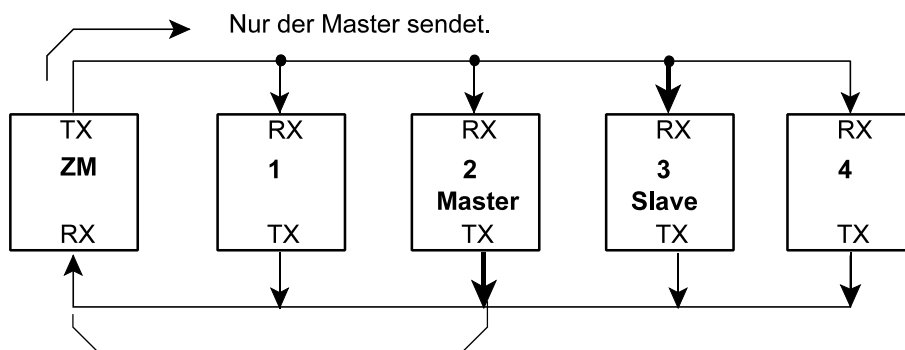
Der Slave wird ausgewählt



Der Master sendet ein Kommando $s @ CR$. Der Hostcontroller reicht es weiter. Die adressierte Einrichtung schaltet sich als Slave zu.

Danach darf der Master weitgehend beliebige Zeichenfolgen übertragen. Sie dürfen aber nicht länger sein, als es die Speicherkapazität des Kommunikationspuffers im Slave zuläßt. Die Zeichenfolgen dürfen keine reservierten Steuerzeichen enthalten.

Master und Slave



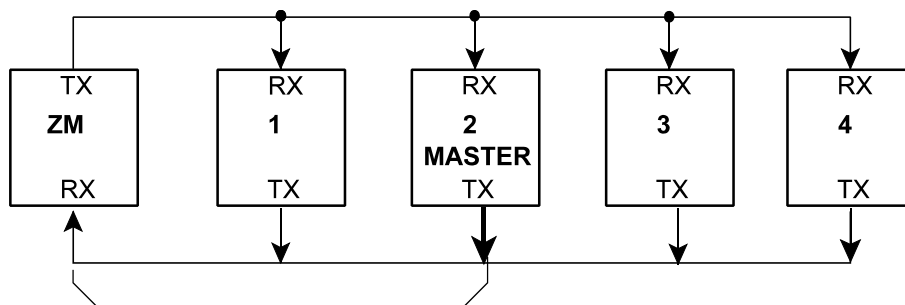
CR beendet die Übertragung eines Datenblocks.

ENQ bewirkt einen Rollentausch von Master und Slave. Der Master wird zum Slave und umgekehrt (Turnaround). Der Hostcontroller ist daran nicht beteiligt.

EBT bewirkt das Trennen vom Slave. Dann kann der Master einen anderen Slave auswählen.

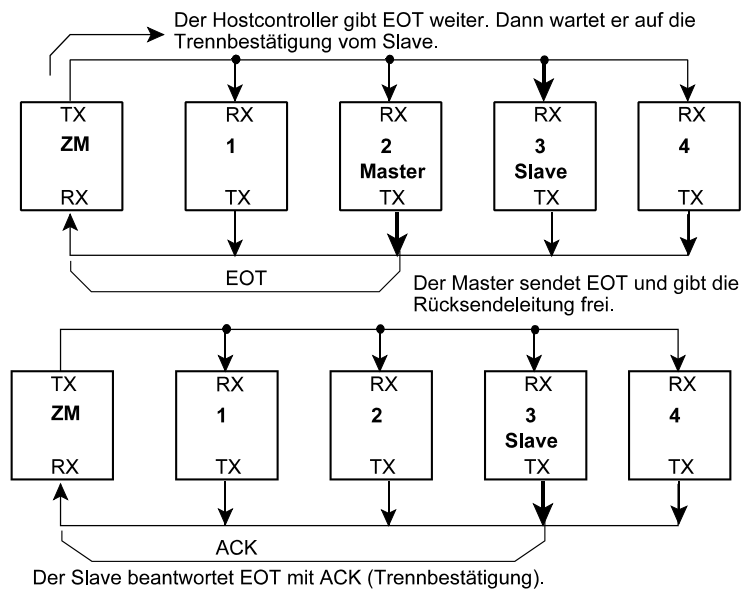
EOT zeigt an, daß der Master die Busherrschaft aufgeben will.

Der Hostcontroller als Slave



Ein Master kann den Hostcontroller als Slave ansprechen.

Das Ende der Busherrschaft

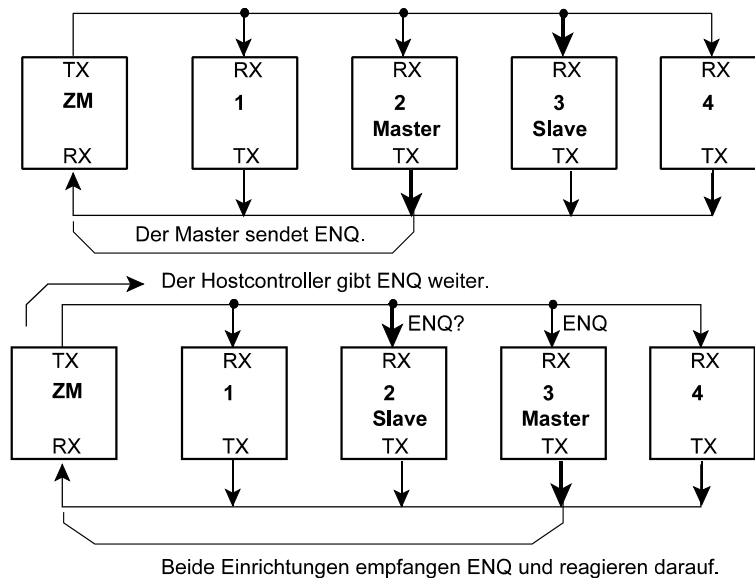


Der Hostcontroller gibt EOT weiter. Dann wartet er auf die Trennbestätigung vom Slave.

Der Master sendet EOT und gibt die Rücksendeleitung frei. Für ihn hat es sich damit erledigt.

Der Slave beantwortet EOT mit ACK (Trennbestätigung). Damit gibt er den Slavezustand auf. Der Übertragungsvorgang ist beendet. Der Hostcontroller setzt die zyklische Abfrage fort.

Rollentausch (Turnaround) zwischen Master und Slave



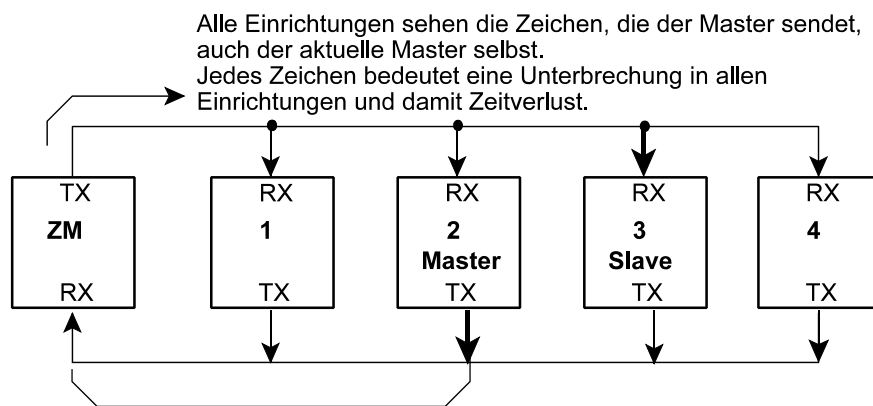
Der Master soll zum Slave werden, der Slave zum Master.

Der Master sendet ENQ und gibt die Rücksendeleitung frei. Er wird aber erst dann zum Slave, nachdem er das gesendete ENQ wieder empfangen hat³.

Der Slave empfängt ENQ, schaltet sich auf die Rücksendeleitung auf und wird zum Master.

Beide Einrichtungen empfangen ENQ und reagieren darauf.

Den Zeitverlust gering halten



Alle Einrichtungen sehen die Zeichen, die der Master sendet, auch der aktuelle Master selbst. Jedes Zeichen bedeutet eine Unterbrechung in allen Einrichtungen und damit Zeitverlust.

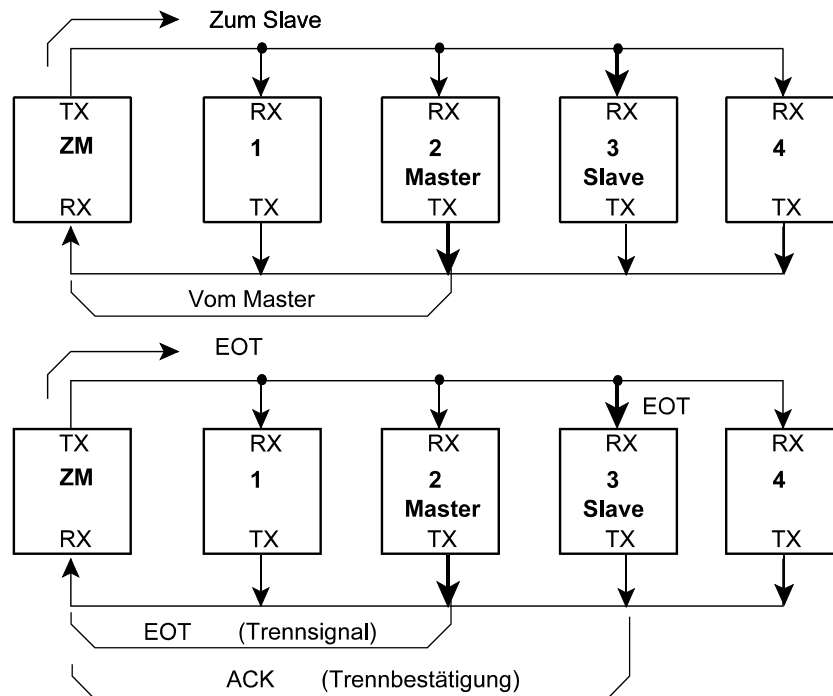
Eigentlich muß der aktuelle Master gar nichts empfangen. Die extreme Lösung wäre das Sperren der Unterbrechungsauslösung. Dann wäre es aber nicht mehr möglich, ihn vom Hostcontroller aus zu beeinflussen (z. B. zur Fehlerbehandlung).

Alle Einrichtungen, die kein Slave sind, müssen nur erkennen, daß der Übertragungsvorgang beendet ist oder daß eine Slaveauswahl stattfindet. Alles andere hat keine Bedeutung.

Das Steuerzeichen EM dient zum Interfacetrennen. Ein ggf. laufender Übertragungsvorgang wird beendet. Eine typische Anwendung ist die Fehleranalyse und -behebung. EM wird nur vom Hostcontroller gesendet. Daraufhin gibt der Master die Rücksendeleitung frei. Alle Einrichtungen gelangen in einen Sonderzustand, so daß der Hostcontroller Slavezugriffe ausführen kann. Um dem Hostcontroller eine Fehleranalyse zu ermöglichen, soll der ursprüngliche Zustand der Einrichtungen sowenig wie möglich verstellt werden. Die einfachste Form der Fehlerbehandlung ist die Maschinenfehleraufzeichnung (Logout, Error Dump).

3: Denn sonst würde es zeitweilig zwei Slaves geben, die beide das ENQ empfangen.

Ein Schreibablauf

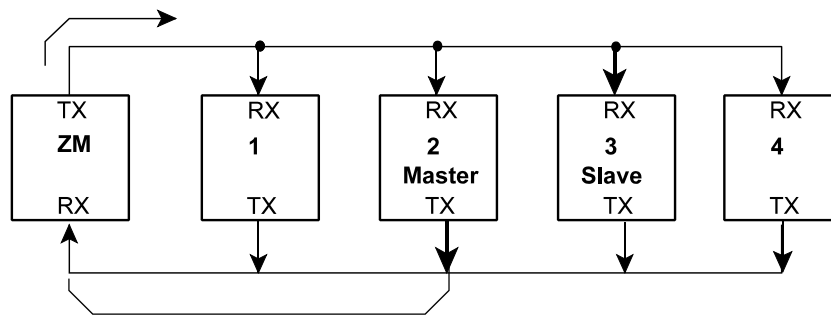


1. Der Master wählt den Slave aus. s--@--CR
2. Der Master schickt den Datenstrom hinterher. Es ist typischerweise eine Folge Kommando – Adresse – Daten – CR.
3. Ist alles erledigt, sendet der Master EOT und gibt danach die Rücksendeleitung frei.
4. Der Slave beendet den gesamten Vorgang, indem er sich auf die Rücksendeleitung aufschaltet, ACK signalisiert und die Rücksendeleitung wieder freigibt. Das ACK geht nur zum Hostcontroller.

Ein Leseablauf

Ein Master kann nur senden, ein Slave nur empfangen. Jeder Lesevorgang muß deshalb in zwei Zugriffe unterteilt werden, (1) dem Schreiben des Lesekommandos und der Adresse und (2) dem Zurücksenden der gelesenen Daten (Split Transaction).

Der Rollentausch hat den Vorteil, daß der ursprüngliche Slave nicht ausdrücklich wissen muß, an wen er die Antwort zu schicken hat, und daß alles zwangsweise ununterbrochen nacheinander abläuft. Es ist eine In-Order-Transaction, keine Out-of-Order-Transaction (die um einiges schwieriger zu implementieren wäre).



1. Der Master wählt den Slave aus. $s @ CR$.
2. Der Master schickt eine Folge Kommando – Leseadresse – $(CR)^4$ – ENQ, gibt danach die Rücksendeleitung frei und macht sich zum Slave (Rollentausch, Turnaround).
3. Der bisherige Slave hat ENQ empfangen, macht sich zum Master, und schaltet sich auf die Rücksendeleitung auf.
4. Der neue Master sendet die Lesedaten. Je nach Lesekommando beendet er dann den Übertragungsvorgang oder signalisiert erneut ENQ.

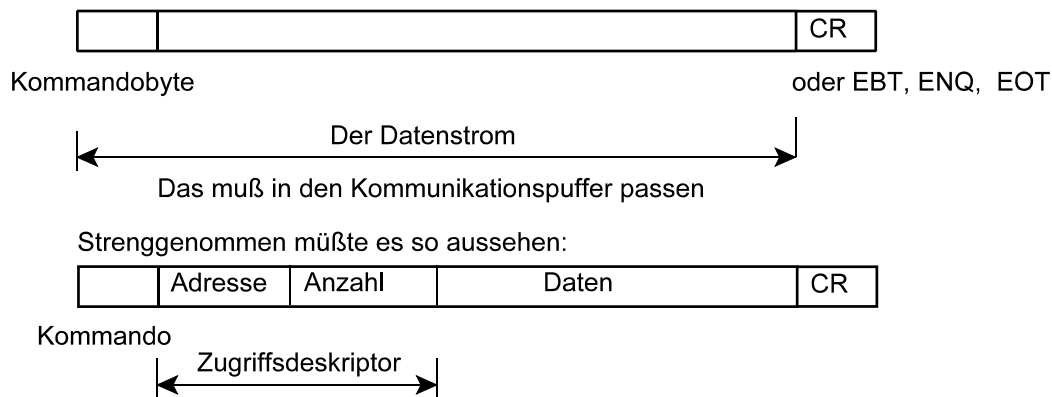
5. Informationsstrukturen der Kommunikation

Schreiben

Eine Schreibdatenstruktur besteht aus einem Kommandobyte, den Daten und dem Abschlußzeichen. Kommandobyte und Daten müssen in den Kommunikationspuffer des Slave passen. Längere Datenströme müssen entsprechend zerlegt werden. Die Synchronisation ist auf einer höheren Protokollebene zu organisieren. Die jeweilige Kommandowirkung des Schreibzugriffs (im Slave) ergibt sich aus der Anwendung. Die universelle Grundsatzlösung besteht darin, den Schreibdaten einen Zugriffsdeskriptor voranzustellen, der eine Anfangsadresse und eine Längenangabe enthält. Man kann aber nicht mit absoluten Adressen arbeiten, denn die wären beim Assemblieren immer wieder zu berücksichtigen; der Master müßte immer die Adressen im Slave kennen. Typische Auswege:

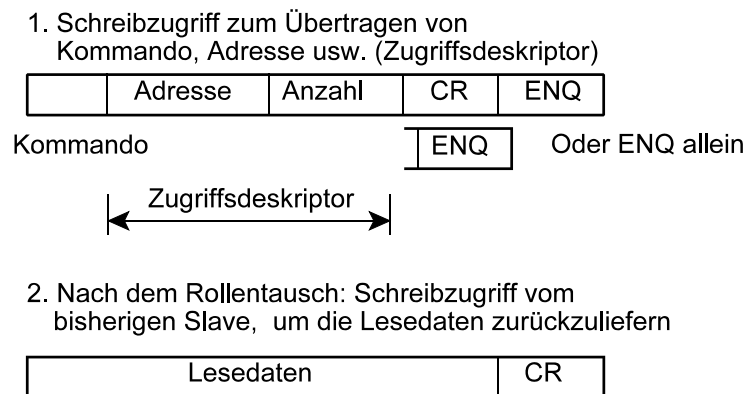
- a) Das Ziel (Destination) wird im Kommando ausgedrückt (z. B. Schreiben in einen Anzeigepuffer).
- b) Es werden keine Adressen übertragen, sondern Ordinalzahlen, wenn nicht sogar Bezeichner in Form einer Zeichenkette (Prinzip der Objektorientierung; vgl. auch die DLL-Aufrufe in Windows).

4: CR kann gegeben werden, muß aber nicht.



Lesen

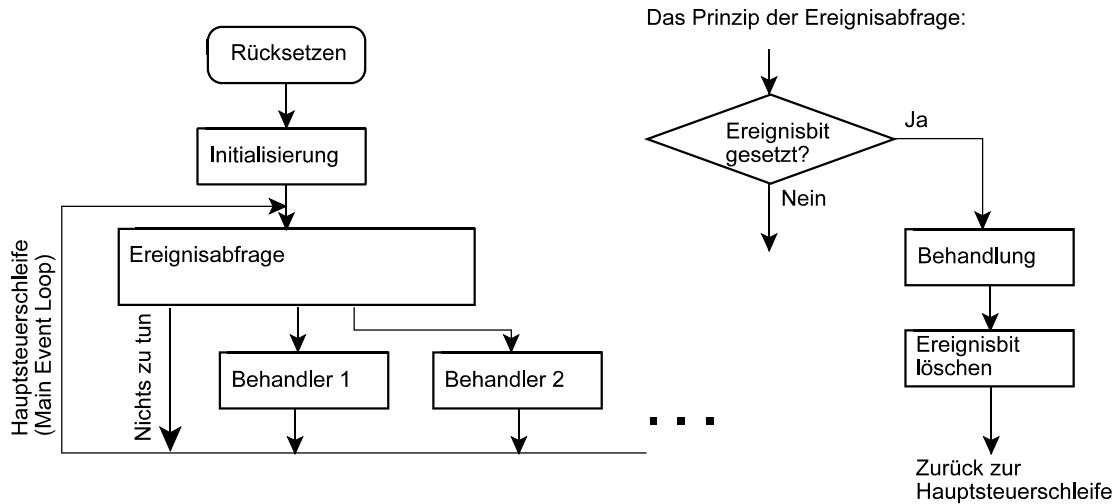
Ein Lesevorgang zerfällt in zwei Schreibabläufe, wobei zwischen beiden Abläufen Master und Slave ihre Rollen tauschen (Turnaround). Der Schreibablauf überträgt das Lesekommando und (falls erforderlich) die näheren Zugriffsangaben (Zugriffsdeskriptor; hierfür gilt das gleiche wie beim Schreibkommando). Das Schreiben wird mit ENQ abgeschlossen. Der Master wird nun zum Slave und wartet auf das Anliefern der Lesedaten. Der bisherige Slave wird zum Master und sendet die Lesedaten zurück. Es ist möglich, die Datenübertragung mit ENQ anzuschließen, um zum ursprünglichen Master zurückzuschalten. Diese Ablaufvariante muß ggf. im Lesekommando codiert werden.



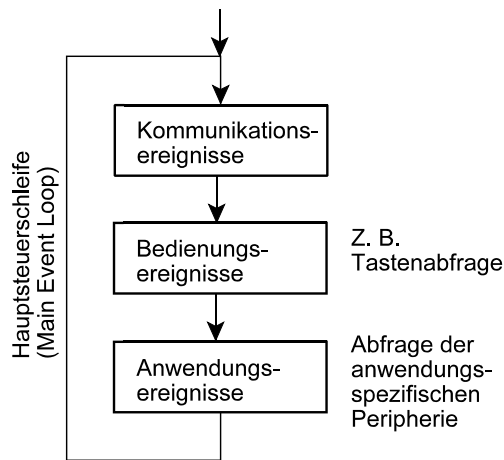
6. Die grundsätzliche Programmorganisation

Nach dem Rücksetzen wird die einrichtung initialisiert. Dann wird eine sog. Hauptsteuerschleife (Main Event Loop) endlos durchlaufen. Die Hauptsteuerschleife fragt Ereignisse ab und ruft ggf. die entsprechenden Ereignisbehandler auf.

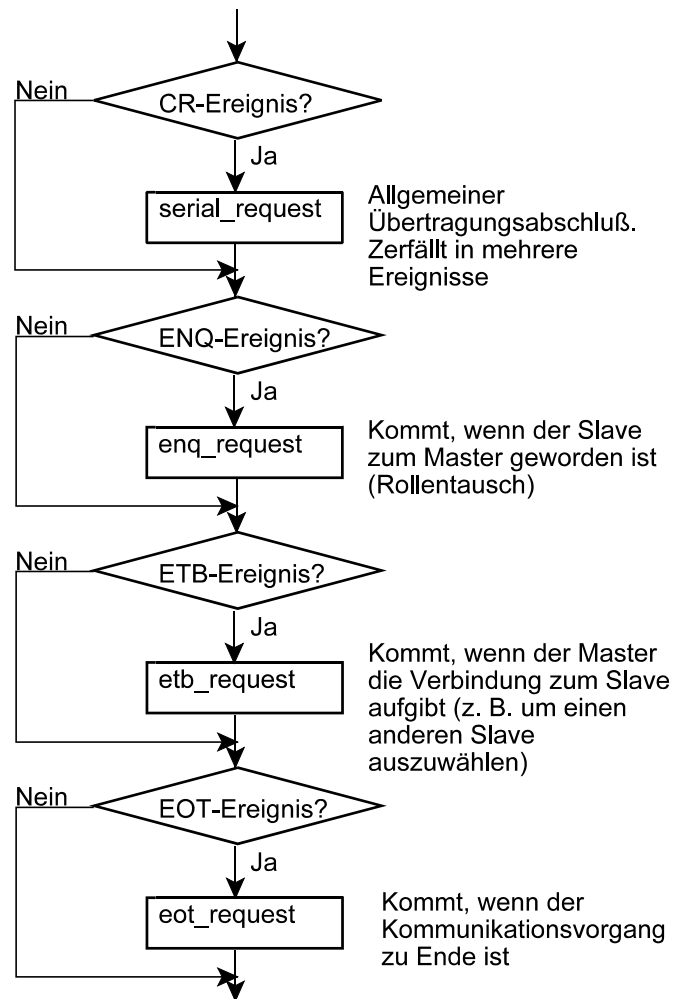
Jeder Ereignisbehandler kehrt zur Hauptsteuerschleife zurück. Es gibt anwendungsspezifische Ereignisse und Ereignisse der Mehrprozessorkommunikation. Letztere werden mit Merkbits signalisiert. Alle Ereignisbehandler rufen Unterprogramme auf, die anwendungsspezifisch auszugestalten sind. Die Behandlung endet mit dem Löschen des jeweiligen Ereignisbits.



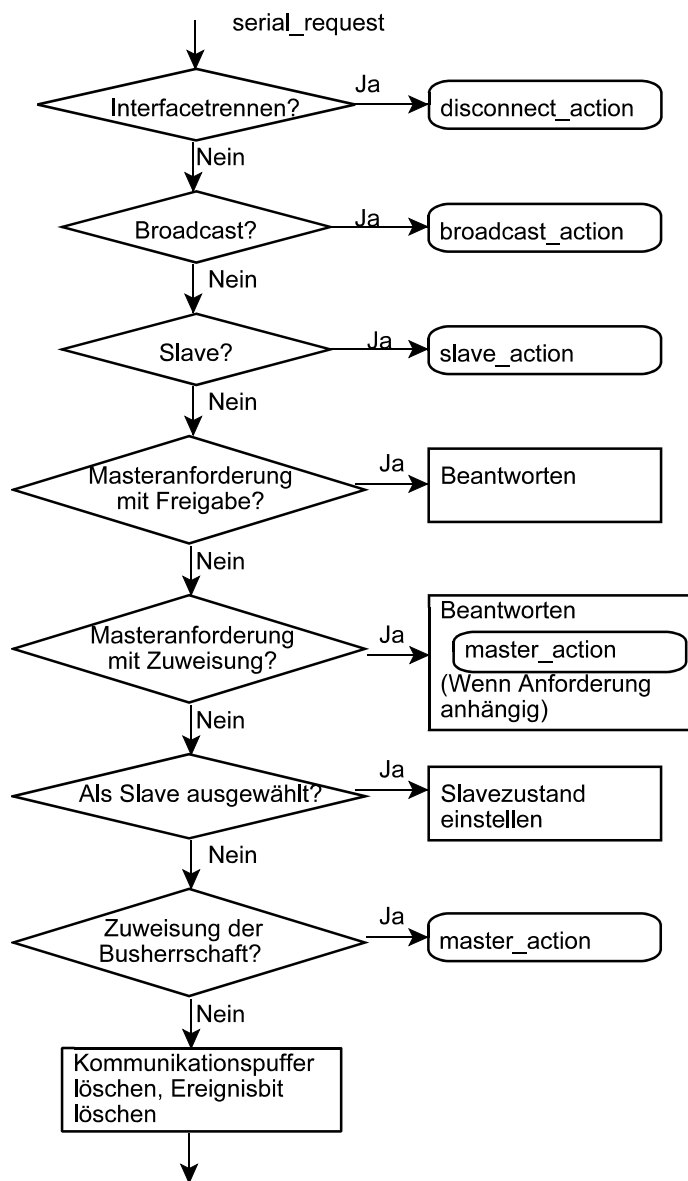
Die Organisation der Ereignisabfrage:



Die Kommunikationsereignisse in der Einrichtung:

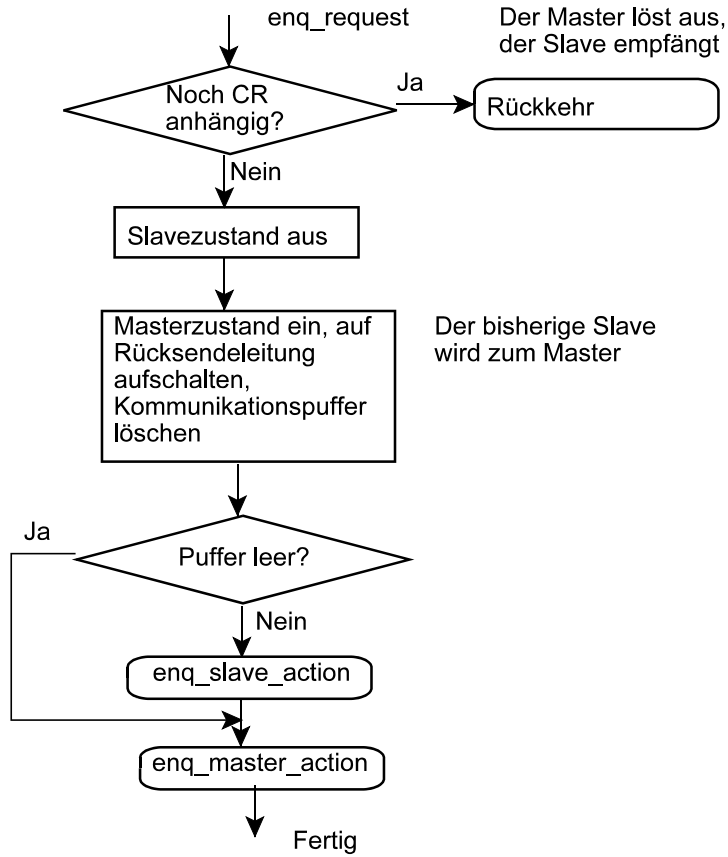


CR – der allgemeine Übertragungsabschluß:



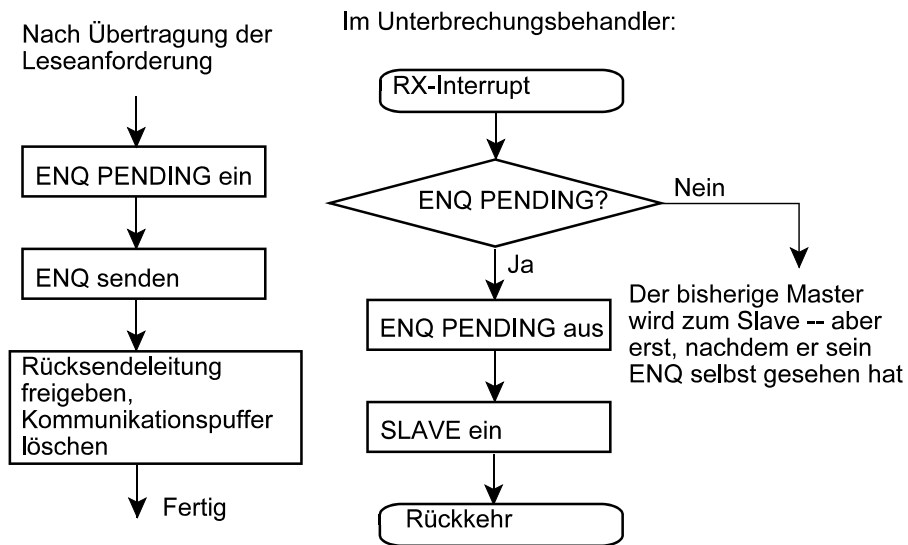
ENQ – der Rollentausch im Slave:

Dieser Ablauf betrifft den bisherigen Slave, der ein ENQ empfangen hat.

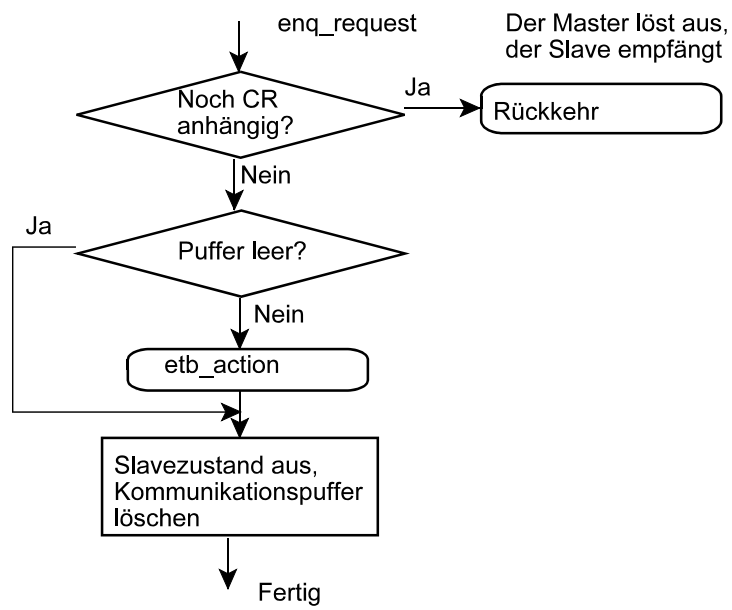


ENQ – der Rollentausch im Master:

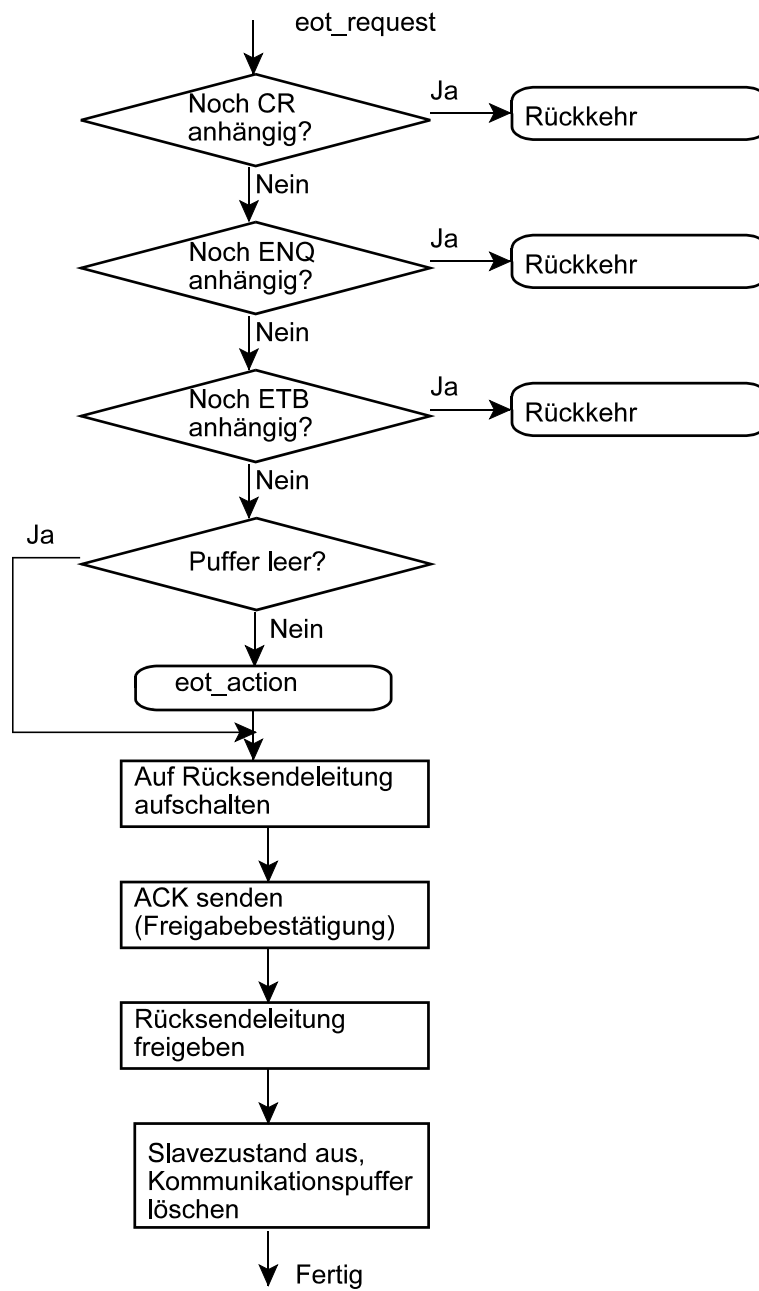
Dieser Ablauf betrifft den bisherigen Master, der ein ENQ sendet und dann zum Slave wird.



ETB – der Slave wird aufgegeben:



EOT – das Ende des Übertragungsvorgangs:



Die Ereignisbehandler

Es sind Unterprogramme, die anwendungsspezifisch zu gestalten sind. Wenn gar nichts zu tun ist, bestehen sie nur aus einem RET-Befehl.

Bezeichnung	Das Ereignis wird ausgelöst, ...	Was ist zu tun?
slave_action	wenn die Einrichtung als Slave ausgewählt ist und CR empfangen wurde	Nachsehen, was im Kommunikationspuffer steht, und entsprechend handeln (z. B. Kommandoausführung)
master_action	wenn der Einrichtung die Busherrschaft zugesprochen wurde	Dasjenige ausführen (als Master), weswegen man die Busherrschaft angefordert hat.
enq_slave_action	wenn die Einrichtung als Slave ausgewählt ist, ENQ empfangen wurde, der Kommunikationspuffer nicht leer ist und die Einrichtung die Busherrschaft übernommen hat	Nachsehen, was im Kommunikationspuffer steht, und entsprechend handeln (z. B. Kommandoausführung)*.
enq_master_action	wenn die Einrichtung als Slave ausgewählt ist, ENQ empfangen wurde die Einrichtung die Busherrschaft übernommen hat und ggf. enq_slave_action ausgeführt wurde	Dasjenige ausführen (als Master), was zuvor (im Slavezustand) vom empfangenen Kommando angewiesen wurde
etb_action	wenn die Einrichtung als Slave ausgewählt ist, ETB empfangen wurde und der Kommunikationspuffer nicht leer ist	Nachsehen, was im Kommunikationspuffer steht, und entsprechend handeln (z. B. Kommandoausführung)*
eot_action	wenn die Einrichtung als Slave ausgewählt ist, EOT empfangen wurde und der Kommunikationspuffer nicht leer ist	Nachsehen, was im Kommunikationspuffer steht, und entsprechend handeln (z. B. Kommandoausführung)*
broadcast_action	wenn CR empfangen wurde und das erste Zeichen im Kommunikationspuffer ein BEL ist. Kann nur vom Hostcontroller gegeben werden.	Nachsehen, was im Kommunikationspuffer steht, und entsprechend handeln (z. B. Kommandoausführung)
disconnect_action	wenn CR empfangen wurde und der Interfacetrennzustand eingestellt ist. Kann nur vom Hostcontroller gegeben werden.	Nachsehen, was im Diagnosepuffer steht, und entsprechend handeln (z. B. Kommandoausführung)

*: ENQ, EBT und EOT sind als Abschluß eines Datenstroms zulässig. Man kann aber auch den Datenstrom mit CR abschließen und danach eines dieser Zeichen einzeln senden. Wird ein solches einzelnes Zeichen empfangen, so ist der Kommunikationspuffer leer (combuffcount = 0); die eigentliche Behandlung wurde dann schon durch CR ausgelöst und ist nun erledigt.

Die allgemeine Programmschnittstelle (API)

Merkmal oder Funktion	Bezeichnung
die Größe des Kommunikationspuffers	combuffersize
der Füllstand des Kommunikationspuffers	combuffcount
der Inhalt des Kommunikationspuffers	combuff
die Größe des Diagnosepuffers	diagbuffersize
der Füllstand des Diagnosepuffers	diagbuffcount
der Inhalt des Diagnosepuffers	diagbuff
das Anfordern der Busherrschaft	Bit master_request im Kommunikationszustandsregister COMSTA setzen
die Ausführung eines Kommandos, wenn die Einrichtung der aktuelle Slave ist	Unterprogramm slave_action, enq_slave_action, ebt_action, eot_action
der Empfang von Lesedaten nach einem Rollentausch vom Master zum Slave	Unterprogramm turnaround_slave_action
die Ausführung der jeweiligen Übertragungsabsicht als Masterer, nachdem die angeforderte Busherrschaft zugewiesen wurde	Unterprogramm master_action
das Zurücksenden beispielsweise von Lesedaten als Master nach einem Rollentausch	Unterprogramm turnaround_master_action
die Ausführung eines Kommandos, das vom Hostcontroller an alle Einrichtungen gesendet wurde	broadcast_action
die Ausführung eines Fehlerbehebungs- oder Diagnosekommandos, das vom Hostcontroller an alle Einrichtungen gesendet wurde	disconnect_action
einen Slave auswählen	Slaveadresse nach a, Unterprogramm slave_connect
ein Zeichen senden ohne Warteüberlappung	Zeichen nach a, Unterprogramm fulltx
ein Zeichen senden ohne zu warten	Zeichen nach a, Unterprogramm fasttx
auf Sendebereitschaft warten und ein Zeichen senden	Zeichen nach a, Unterprogramm waittx
als Master den Übertragungsvorgang beenden	Unterprogramm master_terminate
als Master einen Rollentausch auslösen	Unterprogramm master_to_slave

Steuer- und Zustandsregister im Hostcontroller:

COMSTA:

7	6	5	4	3	2	1	0
		REQ_ PENDING	MASTER_ SELECTED		SLAVE		

EVENTS:

7	6	5	4	3	2	1	0
			NEXT_ DEVICE	ETB_REQ	ENQ_REQ	EOT_REQ	CR_REQ

MODE:

7	6	5	4	3	2	1	0
							POLL_ REQ

DSTA:

7	6	5	4	3	2	1	0
						LCDBUFF COPY	NL

Steuer- und Zustandsregister im Hostcontroller in der Einrichtung:

COMSTA:

7	6	5	4	3	2	1	0
IF_DSC		TURNED_TO _SLAVE	ENQ_ PENDING	NO_ SLAVE	SLAVE	MASTER_ GRANT	MASTER_ REQUEST

EVENTS:

7	6	5	4	3	2	1	0
				ETB_REQ	ENQ_REQ	EOT_REQ	CR_REQ

MODE:

7	6	5	4	3	2	1	0

DSTA:

7	6	5	4	3	2	1	0
						LCDBUFF COPY	NL