

CCI/CSI-Protokoll

Schnittstellen Norm für Ausschankanlagen und Zahlungssysteme

Version 3.5

Status: freigegeben

Inhaltsverzeichnis

1	Hinweise zum Lesen dieser Dokumentation.....	3
1.1	Abkürzungen.....	3
1.2	Begriffe	3
1.3	Zahlendarstellung.....	3
2	Übersicht	4
3	Software Protokoll Kaffeemaschine - Interface	5
3.1	Einführung.....	5
3.2	Definitionen	5
3.3	Protokoll Regeln.....	5
3.3.1	Timeouts	5
3.3.2	Error Handling.....	6
3.3.3	Kommunikations-Beispiele	6
3.4	Telegrammaufbau.....	7
3.4.1	Aufrufe und Rückmeldungen:.....	7
3.4.2	Auswertung der Telegramme:.....	8
3.4.3	Bestätigungen:.....	8
3.4.4	Checksumme:.....	8
3.4.5	Encodierung der Zeichen:	8
3.4.6	Beispiel	9
3.5	Telegrammtypen.....	9
3.5.1	Funktionsumfang des Protokolls: Levels	9
3.5.2	Kompatibilität zu alten Designs	9
3.5.3	Kommando Übersicht.....	10
3.5.4	Vend (nur CCI) – Sperren des Interface’s -	10
3.5.5	Status – Zustandsanfrage beim Interface, Pollen -	11
3.5.6	Credit (nur CCI) – Saldo- und Preisanfrage -	13
3.5.7	Price (nur CCI) – Preisdownload -	14
3.5.8	Inquiry – Verkaufsanfrage -	15
3.5.9	Amount – Betrag abbuchen- (Level 3)	16
3.5.10	Identification – Anfrage Interfacedaten -	17
3.5.11	Machine_Mode (Level 2) – Interfacemodus umstellen -	18
3.5.12	Parameter – Interface parametrisieren- (Level 3)	19
3.5.12.1	Ablauf Schreiben	20
3.5.12.2	Ablauf Lesen	20
3.5.12.3	Parameterliste	20
3.5.12.3.1	Dezimalstelle	20
3.5.12.3.2	Restkredit löschen (Münzprüfer).....	21
3.5.12.3.3	Münzkanäle setzen (Münzprüfer).....	21
3.5.12.3.4	Annahmefreigabe (Münzprüfer).....	21

3.5.12.3.5	Reserviert	21
3.5.12.3.6	Preis Download	21
3.6	Kommunikationsablauf	22
3.6.1	Initialisierung	22
3.6.2	CCI & CSI: Grundzustand (kein Stift/Tag gesteckt)	22
3.6.3	CCI & CSI: Produktetaste an der Kaffeemaschine betätigt	22
3.6.4	Interface verkauft ein Produkt	23
3.6.5	Interface verkauft ein Produkt; Kommando Wiederholungen	23
3.6.6	CSI: Interface verkauft ein Produkt	24
3.6.7	CCI: Das Zahlungssystem antwortet nicht	25
3.6.8	Identifikation anfragen	25
3.6.9	Preis laden	25
3.6.10	CCI & CSI: Während des Pollens tritt eine Störung auf	25
4	Hardwaredefinitionen	25
4.1	Stecker CSS: Maschine - Interface	26
4.2	PWR: Speisung des Interfaces	26
4.3	CSI, Stecker PAR: Interface - Schankanlage	27
5	History	28
6	Anhang	29
6.1	Disclaimer	29
6.2	Kontaktadresse	29

1 Hinweise zum Lesen dieser Dokumentation

1.1 Abkürzungen

CCI	<u>C</u> offee <u>C</u> redit <u>I</u> nterface: Synonym für Zahlungssysteme
CSI	<u>C</u> offee <u>S</u> tandard <u>I</u> nterface: Synonym für Schank Anlagen mit parallel Schnittstelle
VC	<u>V</u> alue <u>C</u> arrier: Datenträger (Mifare, Legic etc.)
PS	<u>P</u> ayment <u>S</u> ystem: Das oder die Zahlungssystem(e), welche(s) mit dem CCI verbunden sind (serielle und/oder parallele Zahlungssysteme)
CSS	<u>C</u> offee <u>S</u> tandard <u>S</u> erial: Stecker zum Anschluss an die Kaffeemaschine
PAR	<u>P</u> arallel <u>R</u> elais: Stecker CSI-Interface <-> Schankanlagen

1.2 Begriffe

Zahlungssystem	Bar- oder bargeldloses Geld- oder Punktesystem
Abrechnungssystem	Gerät zum Auslesen der Verkaufstatistik zwecks Weiterverarbeitung
Interface	Slave Gerät welches über CCI/CSI Protokoll kommuniziert.
Maschine	Synonym für Kaffeemaschine oder Vending Automat

1.3 Zahlendarstellung

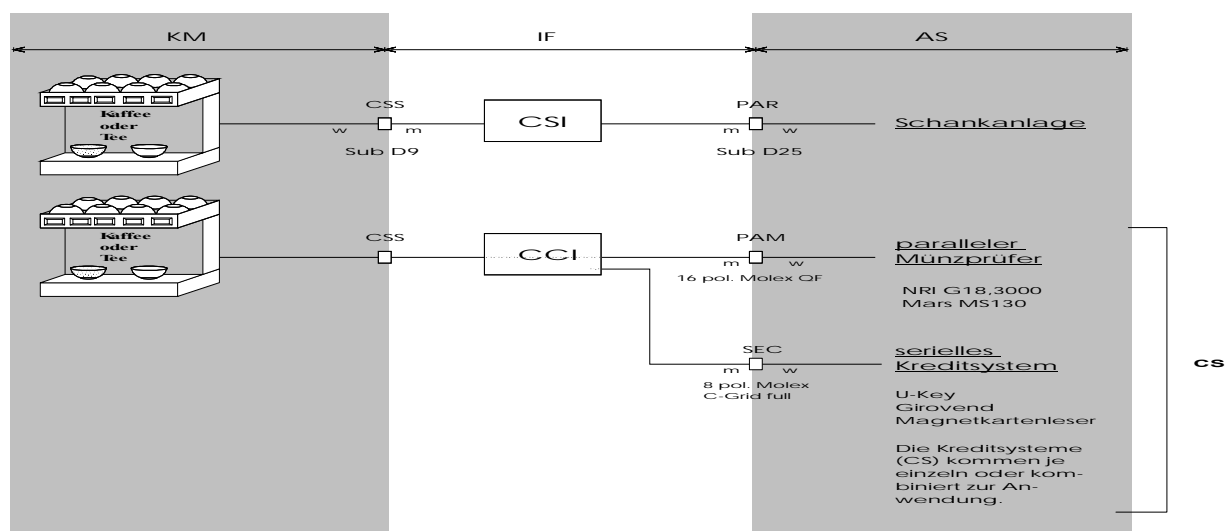
Nicht speziell vermerkte Zahlen sind als Dezimalwerte zu interpretieren. Hexadezimale Zahlen werden in der Form 0xyy dargestellt.

ASCII Zeichen werden in Anführungszeichen dargestellt, z.B. 'A' für den Buchstaben A.

2 Übersicht

Im Sinne einer Vereinheitlichung von Schnittstellen zu externen Zahlungssystemen wurde 1992 von namhaften schweizerischen Unternehmen eine Grundlage des CCI/CSI Protokolls geschaffen.

Basis ist eine RS232 mit Spezifikation des Anschlusskabels. Das Softwareprotokoll beinhaltet ein Set von Kommandos. Die Hersteller von Automaten (oft Kaffeemaschinen) implementieren diejenigen Kommandos, welche sie benötigen, aber mindestens ein Minimum um die Verkäufe korrekt abwickeln zu lassen. Einige Befehle dienen der Komfortsteigerung und können weggelassen werden. Das Protokoll ist derart aufgebaut, dass es für die Kaffeemaschine keine Rolle spielt, welches Interface (CCI oder CSI) angeschlossen ist.



!

Wünscht ein Hersteller eine Erweiterung dieser Spezifikation, ist er verpflichtet, sich beim Herausgeber dieser Dokumentation (siehe Anhang) zu melden. Änderungen und Erweiterungen dürfen nur mit schriftlicher Einwilligung des Herausgebers dieser Spezifikation erfolgen.

3 Software Protokoll Kaffeemaschine - Interface

3.1 Einführung

Das Protokoll ist als Master-Slave Kommunikation konzipiert, wobei die Kommandos im Telegrammstil übertragen werden. Jedes Telegramm vom Master wird mit ACK (Telegramm korrekt empfangen) oder NAK (Fehler) quittiert. Bei einigen Telegrammen folgt eine Antwort mit Dateninhalt. Jedes Telegramm ist mit einer Checksumme gesichert. Die gesamte Übertragung erfolgt in ASCII-Zeichen.

3.2 Definitionen

Master: Kaffeemaschine
Slave: Interface

Übertragungsparameter:

Baudrate:	9600	In älteren Maschinen wurde vereinzelt mit langsameren Baudraten gearbeitet. Für neue Implementationen wird 9600 Baud festgelegt. Andere Baudraten als 9600 sind nur noch aus Kompatibilitätsgründen zu unterstützen.
Anzahl Datenbits:	8	
Parität	None	
Anzahl Stoppbits	1	

3.3 Protokoll Regeln

<u>Master:</u>	<u>Slave:</u>	<u>Bemerkungen:</u>
⇒	<i>Aufruf</i>	- MASTER pollt das Interface.
⇐	<i>Bestätigung</i>	- Bestätigung innerhalb TO_CONFIRM
⇐	<i>[Rückmeldung]</i>	- optional je nach Telegramm. Timeout TO_DATA

Es dürfen jederzeit alle Kommandos geschickt werden. Es ist also keine bestimmte Reihenfolge vorgeschrieben (Ausnahme siehe Kapitel 3.5.8).

Der Master sollte den Slave regelmäßig im Abstand von 100-500ms pollen.

3.3.1 Timeouts

TO_CONFIRM	Bestätigung mit ACK oder NAK:	200 ms
TO_DATA	Rückmeldung mit Dateninhalt :	Standard 5 s, d.h. für alle Kommandos ausser INQUIRY.

Für das INQUIRY Telegramm gilt folgende Regel:
Die genaue Zeit wird im STATUS (Kapitel 3.5.5) Feld TO_PS zurückgemeldet (abhängig vom Zahlungssystem oder von einer Einstellung im Interface).

Während eines aktiven Timeouts bleibt die Kommunikation ruhig, d.h. es dürfen keine anderen Telegramme übertragen werden. Trotzdem gesendete Daten werden ignoriert.

3.3.2 Error Handling

Vom Slave nicht verstandene Telegramme können auf zwei Arten behandelt werden:

- NAK senden
- <silence>, d.h. keine Antwort senden und Wiederholung abwarten

<silence> ist wie NAK zu betrachten und hat die gleichen Auswirkungen. Wenn klare Fehler auftreten, wie unten geschildert, empfiehlt sich die Verwendung von NAK.

<silence>:

Der Slave reagiert mit <silence>, weil er entweder nicht bereit ist (z.B. nicht im Betriebsmodus) oder die Antwort nicht innerhalb des vereinbarten Timeouts TO_CONFIRM abgeschickt werden kann. Der Slave muss die Zeit TO_CONFIRM überwachen; verspätete Antworten sind nicht erlaubt.

Der Master wiederholt das gleiche Telegramm typischerweise 10 Mal, nachdem das TO_CONFIRM Timeout abgelaufen ist. Da die Antwort nach Ablauf des Timeout nicht mehr ankommt, kann der Master in schnellen Abständen wiederholen.

Falls auch nach der genannten Anzahl Wiederholungen keine Antwort ankommt, sollte der Master in die Initialisierung zurückkehren und die pendente Aktion abbrechen. Zusätzlich gibt der Master eine Fehlermeldung aus. Weitere Aktionen sind applikationsspezifisch und können hier nicht vorgeschrieben werden

Es ist anzustreben, dass sich die Kommunikation automatisch aufbaut, wenn der Slave wieder bereit ist. Dazu sollte der Master in bestimmten Abständen, z.B. alle 10 Sekunden mit VEND 0 oder STATUS pollen. Antwortet der Slave wieder, kann der Master weiterfahren, unter Beachtung des JUST_RESET Flags.

NAK hat nur die Aufgabe, nicht verstandene Kommandos wie falsche Checksumme oder falscher Aufbau (z.B. kein ETB) mitzuteilen. Unlogische oder falsche Dateninhalte müssen bei jedem Telegramm individuell abgefangen werden und sind im Abschnitt der Telegramme dokumentiert.

Durch einen strukturierten Software Aufbau gemäss dem OSI Layer Modell vermeidet man eine Vermischung der unterschiedlichen Fehlersituationen. ACK und NAK werden demnach vom Datalink Layer abgehandelt, und bedeutet, dass ein ACK auch gesendet wird, wenn das Telegramm zwar korrekt ankommt, aber dessen Inhalt auf einem höheren Layer nicht verstanden wird.

Versteht der Master die Antwort des Slave nicht, muss er das Kommando ebenfalls wiederholen. Es gibt allerdings hier kein NAK Mechanismus.

3.3.3 Kommunikations-Beispiele

1. Fehlerfreie Kommunikation:

<u>Master:</u>		<u>Slave:</u>	<u>Bemerkungen:</u>
	Telegramm 1	⇒	MASTER pollt das Interface
⇐	Acknowledge		positive Bestätigung

2. Fehlerhafte Kommunikation mit anschließender Verständigung:

<u>Master:</u>		<u>Slave:</u>	<u>Bemerkungen:</u>
	Telegramm 2	⇒	Master pollt das Interface
⇐	not Acknowledge		nicht verstanden, wiederholen
	Telegramm 2	⇒	Master wiederholt die Anfrage
⇐	not Acknowledge		nicht verstanden, wiederholen
	Telegramm 2	⇒	Master wiederholt die Anfrage
⇐	Acknowledge		jetzt verstanden

3. Beschäftigter Slave:

Der Slave ist vorübergehend nicht bereit.

<u>Master:</u>	<u>Slave:</u>	<u>Bemerkungen:</u>
Telegramm 2	⇒	Master pollt das Interface
⇐ <silence>		Nicht bereit, keine Antwort
Telegramm 2	⇒	Master wiederholt die Anfrage
⇐ <silence>		Nicht bereit, keine Antwort
Telegramm 2	⇒	Master wiederholt die Anfrage
⇐ <silence>		Nicht bereit, keine Antwort
Telegramm 2	⇒	Master wiederholt die Anfrage
⇐ Acknowledge		jetzt bereit

4. Abgeschalteter Slave:

Der Slave ist ausser Betrieb.

<u>Master:</u>	<u>Slave:</u>	<u>Bemerkungen:</u>
Telegramm 2	⇒	Master pollt das Interface
⇐ <silence>		Ausgeschaltet, keine Antwort
Telegramm 2	⇒	Master wiederholt die Anfrage
⇐ <silence>		Ausgeschaltet, keine Antwort
Telegramm 2	⇒	Master wiederholt die Anfrage
⇐ <silence>		Ausgeschaltet, keine Antwort
Telegramm 2	⇒	Master wiederholt die Anfrage
⇐ <silence>		Ausgeschaltet, keine Antwort
VEND 0	⇒	Master kehrt in den Initialisierungsmodus zurück
⇐ <silence>		Ausgeschaltet, keine Antwort
Gleiche Sequenz 10x		
VEND 0	⇒	Master kehrt in den Initialisierungsmodus zurück
⇐ Acknowledge		Interface wurde in Betrieb genommen und antwortet wieder

5. unvollständige Antwort:

<u>Master:</u>	<u>Slave:</u>	<u>Bemerkungen:</u>
STATUS	⇒	Master pollt das Interface
⇐ Acknowledge		Telegramm komplett, Checksumme ok
...		kein Datentelegramm. Master wartet TO_DATA
⇐ STATUS	⇒	Master wiederholt nach TO_DATA + 100ms (Sicherheitsmarge)
⇐ Acknowledge		Telegramm komplett, Checksumme ok
⇐ Datenantwort		jetzt ok

3.4 Telegrammaufbau

3.4.1 Aufrufe und Rückmeldungen:

Byte Telegramm
mit Daten

1	Start	Startzeichen (STX: 0x02)
2	Typ	Kommandotyp
3	Data n	erstes Datenbyte (höchstwertiges Byte)

n+3	Data 1	letztes Datenbyte (tiefst wertiges Byte)
n+4	End of Text	ETX = 0x03
n+5		bcc: Block Check Character: Hexwert in zwei
n+6	Check	ASCII-Character übermittelt
n+7	End of Block	ETB = 0x17

Byte Telegramm
 ohne Daten

1	Start	Startzeichen (STX: 0x02)
2	Typ	Kommandotyp
3	End of Text	ETX = 0x03
4		bcc: Block Check Character: Hexwert in zwei
5	Check	ASCII-Character übermittelt
6	End of Block	ETB = 0x17

3.4.2 Auswertung der Telegramme:

Um die Kommunikationssoftware in Bezug auf Telegrammlängen ausbaufähig zu halten, wird unbedingt empfohlen, die Steuerzeichen ETX oder ETB zu prüfen und dann mit der Auswertung zu beginnen.

Die Gültigkeit der Telegramme ist durch die Checksumme gewährleistet.

3.4.3 Bestätigungen:

Byte Acknowledge

1	ACK	ACK = 0x06
---	-----	------------

Byte not
 Acknowledge

1	NAK	NAK = 0x15
---	-----	------------

3.4.4 Checksumme:

Der bcc berechnet sich aus einer XOR-Verknüpfung aller vorhergehenden Zeichen ohne STX aber incl. ETX:

<bcc> = 0 XOR <char1> XOR <char2> XOR ETX.

Dessen Übertragung erfolgt in zwei Byte ASCII-Charakteren.

Beispiel: bcc = 0x8A wird als '8A', also 0x38 gefolgt von 0x41 übertragen.

3.4.5 Encodierung der Zeichen:

Jedes Zeichen (Daten) wird als dezimales ASCII-Zeichen übertragen

Beispiel: die Zahl 123 wird in drei ASCII-Zeichen '1','2','3' (0x31,0x32,0x33) übermittelt. Die höchstwertige Ziffer wird zuerst übermittelt, die tiefst wertige zuletzt (Big Endian).

Bei Bitfeldern (z.B. im Telegramm STATUS das Byte IFSTAT und 2 reservierte Bytes) wird das höchstwertige Bit gesetzt, um Sonderzeichen zu vermeiden

3.4.6 Beispiel

Byte	Telegramm mit Daten	Hex-Wert
1	Start	0x02 (=STX)
2	Typ = S	0x53 (= 'S')
3	Data 1= 2	0x32 (= '2')
4	Data 2= 9	0x39 (= '9')
5	Data 3 = 0	0x30 (= '0')
6	End of Text	0x03 (=ETX)
7		0x36 (= '6')
8	Check	0x42 (= 'B')
9	End of Block	0x17 (=ETB)

Hinweis: Der Dateninhalt des Beispiels hat keine Relevanz.

3.5 Telegrammtypen

3.5.1 Funktionsumfang des Protokolls: Levels

Damit das Protokoll erweitert werden kann, werden *Levels* eingeführt. Grundsätzlich wird der gewählte Level durch den Master bestimmt.

Ein Interface, welches einen höheren Level unterstützt, muss zwingend den Umfang der tieferen Levels abdecken. Ausnahme: neue Kommandos, welche in der Antwort dem Master mitteilen können, dass das Kommando nicht unterstützt ist. D.h. optionale Telegramme, welche der Slave nicht unterstützt, werden entsprechend der Spezifikation der entsprechenden Telegramme mit "nicht unterstützt" beantwortet. Der Master muss darauf vorbereitet sein

Eine Maschine, welche einen tieferen Level als das Interface unterstützt, kann die zusätzlichen Funktionen nicht nutzen.

Damit Geräte mit unterschiedlichem Level zusammenarbeiten können, gelten folgende Regeln:

Level Master:	Level Slave:	Beschreibung:
x	>x	Im Level (>x) erweiterte Antworten werden vom Master ignoriert.
x	<x	Die Master benutzt keine Telegramme höheren Levels
x	x	Keine Verständigungsprobleme, weil beide den gleichen Level besitzen

Nicht spezifizierte Telegramme, entsprechen dem Level 1.

Sendet der Master **entgegen** der Vorschrift ein Kommando an das Interface, welches dieses aufgrund eines tieferen Levels nicht unterstützt, antwortet das Interface nur mit ACK, die Datenantwort wird jedoch nicht gesendet.

Liefert der Slave einen höheren Level als in der aktuellen Norm festgelegt, geht der Master vom höchsten bekannten Level aus.

3.5.2 Kompatibilität zu alten Designs

Hinweise zu möglichen Kompatibilitätsproblemen mit alten Maschinen sind in der Version 3.4 dieser Spezifikation nachzulesen.

3.5.3 Kommando Übersicht

Name	Level	Typ	Data Master	Data Slave	Funktion
Vend	1	V	1	-	Abrechnungsgerät sperren/freigeben
Status	1	S	0	4	Slave pollen mit Statusanfrage
Credit	1	C	4	7	Saldo- und Preisanfrage
Price	1	P	10	-	Preis einer Preisliste in den Slave laden
Inquiry	1	I	4	1	Produkt Verkauf
Identification	1	X	-	6/8	Interfacetyp und Version
Machine_Mode	2	M	3	2	Modus setzen (gratis, Test, blockieren)
Parameter	3	E	9	1	Einstellung im Interface lesen und speichern
Amount	3	B	13	1	Betrag abbuchen
reserviert	-	F	-	-	Zukünftige Erweiterung
reserviert	-	a	-	-	von Herstellern privat verwendete Kommandos
Ack	1	-	-	-	Kommando verstanden
NAK	1	-	-	-	Kommando nicht verstanden
<silence>	1	-	-	-	Keine Antwort, weil nicht bereit.

3.5.4 Vend (nur CCI) – Sperren des Interface's –

Master

1	Start	STX
2	Typ = V	'V'
3	Betriebsmode	c: c 0=disable; 1=enable
4	End of Text	ETX
5		
6	Check	
7	End of Block	ETB

Antwort vom Slave

1	Acknowledge	ACK
---	-------------	-----

Mit diesem Kommando kann das Interface veranlasst werden, das Zahlungssystem freizugeben oder zu sperren (z.B. für die Münzannahme). Falls der Master nicht verkaufen kann (Störung, Aufheizen, Freiverkauf u.a.) sollte das Zahlungssystem gesperrt werden.

Nach dem Starten des Interfaces sind die angeschlossenen Zahlungs-Systeme gesperrt. Damit der Betrieb aufgenommen wird, muss **zwingend** ein VEND enable (oder Machine_Mode enable) gesendet werden. Wird während des Betriebes ein VEND disable geschickt, bleibt das Interface solange gesperrt, bis ein VEND enable empfangen wird.

Der Einfluss des VEND Kommandos auf die verschiedenen Interface Zustände ist aus der Tabelle im Kapitel 3.5.11 ersichtlich.

Für neue Implementationen wird empfohlen, anstelle VEND das Kommando MACHINE_MODE (siehe Kapitel 3.5.11) zu verwenden.

3.5.5 Status – Zustandsanfrage beim Interface, Pollen –

Master

1	Start	STX
2	Typ = S	'S'
3	End of Text	ETX
4		
5	Check	
6	End of Block	ETB

Antwort vom Slave

1	Acknowledge	ACK
1	Start	STX
2	Typ = S	'S'
3	X	x=Status ¹ : 0 : no action 3: error Zahlungssystem 1: ready 4: error Interface 2: busy
4	IF_STAT	verschiedene Stati (siehe unten)
5	TO_PS	Timeout_Payment-System: Antwortzeit auf ein <i>Inquiry</i> Telegramm
6		
7	End of Text	ETX
8		
9	Check	
10	End of Block	ETB

Die Statusanfrage 'S' wird zum Polling benutzt.

Ein Produktebezug kann immer versucht werden unabhängig vom aktuellen Status. Preis 0 im Interface: ein Verkauf ist auch bei x=no action möglich.

Bedeutung der Status Meldung x:

- **no action:**
 - CCI: Kein Saldo oder kein Datenträger vorhanden
 - CSI: Kein Kellnerstift gesteckt.
 - Das Interface ist im Ruhezustand.
- **ready:**
 - CCI: Das Interface ist bereit für einen Produktverkauf (Saldo > 0, Datenträger vorhanden)
 - CSI: Ein Kellnerstift wurde gesteckt.
- **busy:**
 - CCI: nicht verwendet
 - CSI: Die Meldung an die Schankanlage über die Relais läuft.
 - Das Interface kann in diesem Zustand keine weitere Produkthanfrage entgegennehmen. Es muss gewartet werden, bis der Zustand wechselt.
- **error Zahlungssystem:**
 - CCI: Störung Zahlungssystem. Diese Meldung darf nur für Anzeigezwecke verwendet werden.
 - CSI: - (nicht benutzt)

¹ nicht alle Zustände werden von allen Interfaces unterstützt.

Im Status ready sind möglicherweise keine Verkäufe möglich. Produktanfragen (Kommando 'I') können mit *zuwenig Kredit* beantwortet werden.

- **error Interface:**

Im Interface ist ein Fehler aufgetreten, so dass dessen Funktion ganz oder teilweise nicht mehr gewährleistet ist. Auf der Maschine sollte eine entsprechende Meldung ausgegeben werden.

Bedeutung der Statusmeldung IF_STAT:

7	6	5	4	3	2	1	0
1	X	X	CREDIT_HIDDEN	JUST_RESET	AB_GELD	SERVICE	GRATIS

GRATIS		CCI	Das Interface ist auf Gratisverkauf eingestellt.
SERVICE		CCI	Das Interface befindet sich im Programmiermodus. In diesem Zustand können Testverkäufe erfolgen.
AB_GELD		CCI	Der Münzwechsler meldet, dass zuwenig Geld in den Tuben vorhanden sind, um Wechselgeld zu geben. Die Maschine sollte dies anzeigen.
JUST_RESET		CCI/ CSI	Das Interface ist neu gestartet worden. Die Maschine muss eine neue Initialisierung ausführen. Dies betrifft die Kommandos STATUS, IDENTIFICATION, PRICE und VEND/ MACHINE_MODE. Das Flag wird nach Senden eines VEND (siehe Kapitel <i>Vend</i> (nur CCI)) oder MACHINE_MODE Kommandos zurückgesetzt, aber nur wenn seit dem Booten einmal ein Status erfragt wurde.
CREDIT_HIDDEN	Level 2	CCI	Der gemeldete Saldo darf auf der Maschine nicht angezeigt werden.
Bit7		CCI/ CSI	Immer "1",
X (Bit5, 6)		CCI/ CSI	Reserviert(Franke)

Bedeutung der Statusmeldung TO_PS:

Je nach angeschlossenem Zahlungssystem dauert der Abbuchvorgang unterschiedlich lange. Mit diesen Byte wird angezeigt, wie lange die Antwort auf ein Inquiry verzögert sein kann.

Die Zeit in Sekunden berechnet sich wie folgt: Timeout [s] = TO_PS - '0'.

Ausnahme: Der Wert 80H ('C') entspricht der Defaulteinstellung (siehe 3.3.1).

Erlaubter Wertebereich: 0x31 ('1') < TO_PS < 0xC8 (entspricht 1²...152s).

Beispiel:

1. TO_PS = '1' (0x6C) -> Timeout = TO_PS - 0x30 = 60 Sekunden
2. TO_PS = '1' (0x31) -> Timeout = TO_PS - 0x30 = 1 Sekunde -> gesetzt auf Standard Timeout = 5 Sekunden

Reserviertes Byte (Byte 6):

7	6	5	4	3	2	1	0
1							

Bit 7 CCI,CSI Immer "1"

² Im Kapitel 3.3.1 sind die Standard-Timeouts definiert. Das rechnerische Resultat kann Werte liefern, welche unter dem Default Timeout liegt. Es empfiehlt sich, diese kleineren Werte auf das Default Timeout zu setzen, d.h. das Standard Timeout ist das minimale Timeout, welches für INQUIRY gültig ist. Resultate grösser als das Standard Timeout „überschreiben“ dieses.

3.5.6 Credit (nur CCI) – Saldo- und Preisanfrage –

Master

1	Start	STX
2	Typ = C	'C'
3	n3	dreistellige Artikelnummer (nur bei exec=1 relevant)
4	n2	
5	n1	
6	e	e=exec: 0: totaler Saldo 1: Produktpreis zu Artikelnummer nnn 2: Restkredit löschen
7	End of Text	ETX
8		
9	Check	
10	End of Block	ETB

Antwort vom Slave

1	Acknowledge	ACK
1	Start	STX
2	Typ = C	'C'
3	n6	Saldo oder Preis (6 stellig).
4	n5	
5	n4	
6	n3	
7	n2	
8	n1	
9	p	Dezimalpunkt Position
10	End of Text	ETX
11		
12	Check	
13	End of Block	ETB

Mit diesem Kommando wird entweder das vorhandene Guthaben angefragt (exec=0), oder der Einzelpreis des Produktes von welchem die Produktnummer übermittelt wurde (exec=1). Diese Daten können für die Anzeige an der Maschine Verwendung finden.

Weiter kann der aktuelle Kredit gelöscht werden, z.B. nach einer bestimmten Zeit nach einem getätigten Verkauf (zu Ungunsten des Kunden).

Artikelnummer: Die Nummerierung beginnt bei 1. Das Interface muss mindestens so viele Wahlen unterstützen, wie die Maschine Produkte hat.

'000000'-'FFFF00' Produktpreis oder Saldo. Ist das Interface auf Freiverkauf eingestellt, werden alle Preise mit '000000' gemeldet

'FFFFFF' Artikel Überlauf: Es wird eine Artikelnummer angefragt wird, welche nicht existiert, weil der Bereich überschritten ist.

'FFFFFFE' (Level 2) Das angefragte Produkt ist gesperrt und kann nicht verkauft werden

'Sperr' (Level 1)

'FFFFFD' Der Preis kann nicht berechnet werden

'FFFFFC' ungültiger exec Wert

'FFFFFB' (Level 3) Einheits-Jeton (exec=0): Mit diesem Jeton kann ein beliebiges Produkt unabhängig vom Preis gekauft werden. Der Master sollte eine passende Meldung anzeigen, anstelle des Saldos. Nach dem Verkauf ist der Saldo 0

(siehe auch Kap. 3.5.12.3.3)

Die Dezimalpunkt-Position gibt an, an welcher Stelle von rechts sich dieser befindet:

p=0 kein Punkt vorhanden
p=2 zwei Nachkommastellen (CHF, EUR)

Falls der Saldo gelöscht werden soll (exec=2), wird bei der Antwort der bereits aktuelle Saldo, also 0 als Antwort gesendet, sofern dies möglich ist (Datenträger Saldo kann z.B. nicht gelöscht werden) und vom Interface unterstützt wird

3.5.7 Price (nur CCI) - Preisdownload -

Master:

1	Start	STX
2	Typ = P	'P'
3	l	Preisliste 0-n
4	n3	Dreistellige Artikelnummer.
5	n2	
6	n1	
7	p6	Preis in der kleinsten Einheit
8	p5	
9	p4	
10	p3	
11	p2	
12	p1	
13	End of Text	ETX
14		
15	Check	
16	End of Block	ETB

Antwort vom Slave

1	Acknowledge	ACK
---	-------------	-----

Ein Verkauf mit dem Kommando *Inquiry* verwendet die im Interface gespeicherten Preise. Mit dem Kommando *Price* ist es möglich, die Preisverwaltung in die Maschine zu verlegen. *Price* erlaubt die in der Maschine gespeicherte Preise ins Interface zu übermitteln.

Vom Interface kann nur *ACK* (verstanden) oder *NAK* (nicht verstanden) retourniert werden, wobei damit nur gemeint ist, ob das Telegramm richtig empfangen wurde (Checksumme etc.). Damit wird der Inhalt noch nicht auf Plausibilität überprüft. Insbesondere wird z.B. eine **ungültige Produktnummer nicht mit einem Fehler quittiert**.

Für neue Implementationen wird der Befehl AMOUNT (siehe 3.5.9) empfohlen anstelle PRICE, sofern das Interface keine Statistik führt (Einschränkung mehrere Preislisten bei AMOUNT). Das Kommando PARAMETER (siehe 3.5.12) implementiert die PRICE Funktion ebenfalls in einer verbesserten Variante (Handshake siehe 3.5.12.3.6).

Wertebereiche:

- Preisliste Die Nummer entspricht der Preislisten Nummerierung im Interface.
0: Barverkäufe
1-n: bargeldlose Bezüge
n ist die höchste vom Interface unterstützte Preisliste.
- Produktnummer Beginnend bei 1
- Preise Betrag in der kleinsten monetären Einheit

Richtig empfangene Preise werden sofort nullspannungssicher im Speicher abgelegt und überschreiben die bereits vorhandenen Einstellungen.

Der Dezimalpunkt kann mit dem CREDIT Kommando erfragt werden.

3.5.8 Inquiry - Verkaufsanfrage -

Master

1	Start	STX
2	Typ = I	'I'
3	n	dreistellige Artikelnummer.
4	n	
5	n	
6	e	exec: '0': nur Überprüfung '1': abbuchen wenn okay
7	End of Text	ETX
8		
9	Check	
10	End of Block	ETB

Antwort vom Slave

1	Acknowledge	ACK
---	-------------	-----

1	Start	STX
2	Typ = I	'I'
3	x	x = status: '0': credit low '1': credit okay
4	End of Text	ETX
5		
6	Check	
7	End of Block	ETB

Mit dem Kommando *INQUIRY* wird versucht, das Produkt mit der Artikelnummer (Tasten- oder Wahlnummer) abzubuchen. Das Interface vergleicht dazu das Guthaben mit den gespeicherten Preis. Diese Preise können über das Kommando *PRICE* oder via Konfiguration ins Interface gelangen.

Bei exec=0 wird nur geprüft, ob das Guthaben ausreichend ist, ohne dass tatsächlich bei einem Zahlungssystem abgebucht wird. Da die Überprüfung nicht direkt im Zahlungssystem durchgeführt wird, kann die Überprüfung ein falsches Resultat ergeben.

Bei exec=1 wird (genügend Saldo vorausgesetzt) die Abbuchung ausgeführt. Der Master startet die Produktausgabe nach Empfang von *credit okay*. *credit low* bedeutet, dass zum Zeitpunkt der Buchung nicht genügend Guthaben zur Verfügung gestanden hat.

Für die Antwort mit Dateninhalt (nicht ACK oder NAK) gilt ein anderes Timeout (siehe Kapitel 3.3.1).

Artikelnummer:

Der gültige Produktbereich ist abhängig von Maschine und Interface und wird hier nicht vorgegeben. Liegt eine Anfrage vor mit einer ungültigen Produktenummer, wird *credit low* zurückgesendet. Die gleiche Antwort erhält man bei einem gesperrten Produkt.

Absicherung bei nicht verstandener Antwort:

Damit keine Mehrfachabbuchungen bei Nichtverstehen der Antwort auf *INQUIRY* vorkommen können, muss auf ein *INQUIRY* Kommando stets *STATUS* folgen. Das Interface interpretiert dies als Quittung auf die Antwort *credit okay*.

Hat die Maschine einmal diese Antwort nicht verstanden, wird sie unmittelbar nochmals das INQUIRY Kommando wiederholen. Weil die STATUS Anfrage gefehlt hat (zwischen den beiden INQUIRY) wird das Interface diesmal nicht mehr abbuchen, sondern nur noch einmal die Antwort wiederholen.

Derselbe Mechanismus gilt, wenn INQUIRY *nur Überprüfung* angefragt wird.

Der Mechanismus im Fehlerfall mit Wiederholungen gehorcht den Regeln wie sie im Kapitel 3.3.2 beschrieben sind. Ein explizites Beispiel zu möglichen Fehlerfällen ist auf Seite 23 ersichtlich.

3.5.9 Amount – Betrag abbuchen– (Level 3)

Master

1	Start	STX
2	Typ = B	'B'
3	n3	n: PLU, Wahlnummer
4	n2	Bereich: '001'-'999'. (nicht unterstützt: '000')
5	n1	n1=LSB, n3=MSB
6	p6	Betrag zum Abbuchen
7	p5	p1=LSB, p6=MSB
8	p4	Bereich: '000000'-'999999'
6	p3	
7	p2	
8	p1	
9	e	exec: 1: nur Überprüfung 0: abbuchen wenn okay Hinweis: Gegenüber dem Inquiry Kommando ist die Bedeutung von exec vertauscht!
2	l	Reserve (default '0')
2	m	Reserve (default '0')
6	End of Text	ETX
7		
8	Check	
9	End of Block	ETB

Antwort vom Slave

1	Acknowledge	ACK
1	Start	STX
2	Typ = B	'B'
3	x	Status: '0'=credit low '1'=credit okay
7	End of Text	ETX
8		
9	Check	
10	End of Block	ETB

Mit dem AMOUNT Befehl wird ein Betrag vom Interface abgebucht. Dieser Befehl ist insbesondere dann nützlich, wenn der Master die Preise selber verwaltet.

Die Wahlnummer wird für Statistikzwecke verwendet, wenn das Interface eine Verkauf Statistik führt. Der Master hat die Wahl, diese Nummer korrekt aufzufüllen, oder ,000' zu melden (Wahlnummer nicht unterstützt).

x Beschreibung



'0' *credit low*: Der verfügbare Saldo reicht nicht aus, um den Betrag abzubuchen.
 '1' *credit okay*:
 bei exec=0 wurde der Betrag abgebucht, das Produkt kann ausgegeben werden.
 Bei exec=1 wäre der Saldo ausreichend, um den Betrag p abzubuchen

Wichtig: Wie beim INQUIRY Kommando muss auch hier nach jedem AMOUNT ein STATUS folgen (siehe Kapitel 3.5.8)

3.5.10 Identification – Anfrage Interfacedaten –

Master

1	Start	STX
2	Typ = X	'X'
3	End of Text	ETX
4		
5	Check	
6	End of Block	ETB

Antwort vom Slave

1	Acknowledge	ACK
---	-------------	-----

1	Start	STX
2	Typ = X	'X'
3	i	i = Interfacetyp: 0: reserviert 1: CSI (Coffee Standard Interface) (Level 1) 2: Typ CCI (Coffee Credit Interface), d.h. ein Interface zur Steuerung von Zahlungssystemen 3: GABI (Gastro Bedienungsinterface) 4-6: reserviert für Fremdsysteme 7: MSS (Melitta System Service) 8: Datalogger (reduzierter Kommandosatz) 9, a...: reserviert für neue CCI/CSI Anwendungen
4	t2	tt = Zahlungssystem: ³
5	t1	00: Münzprüfer 60: MDB Münzwechsler 61: MDB Notenleser 62: MDB Card Reader Weitere Zahlungssystem-Nummern entnehme man dem Handbuch des entsprechenden Interfaces
6	v3	v vv = Interface SW-Version
7	v2	Beispiel: vv = 261 (v2.61). Längere Versionsnummern werden auf drei Stellen reduziert (z.B. 2.90.6 -> 290)
8	v1	
9	l2	Level: Funktionsumfang der implementierten Spezifikation l2: Bitmaske, welche Features implementiert sind.

³ Die Liste kann erweitert werden ohne Änderung der Dokumentation.

10	l1	
11	End of Text	ETX
12		
13	Check	
14	End of Block	ETB

l1:
 01=Level 1
 02=Level 2
 03=Level 3
 00,04..09: reserviert für
 zukünftige Erweiterungen

Mit dem Kommando IDENTIFICATION werden Information über Art und Konfiguration des Interfaces angefordert. Diese können im Master ausgewertet werden (z.B. für eine automatische Konfiguration der Maschine).

Das Zahlungssystem tt ist nur noch aus Kompatibilitätsgründen angegeben, hat aber an Bedeutung verloren. Das CSI sendet in tt immer 99 zurück.

i	Interface Typ	Beschreibung
'1'	CSI	reduzierter Kommandosatz
'2'	CCI	kompletter Kommandosatz mit Level 1-3

3.5.11 Machine Mode (Level 2) - Interfacemodus umstellen -

Master

1	Start	STX
2	Typ = M	'M'
3	MMODE	Verkaufsmodus der Maschine: 1: Normalbetrieb 2: Gratisbetrieb 3: Test/ Servicebetrieb 4: Maschine ist ausser Betrieb 5: Service + Data_Entry (Level 3)
4		reserviert HGZ (Defaultwert ,0')
5		80H reserviert für zukünftige Anwendungen
6	End of Text	ETX
7		
8	Check	
9	End of Block	ETB

Antwort vom Slave

1	Acknowledge	ACK
---	-------------	-----

1	Start	STX
2	Typ = M	'M'
3		0: reserviert für zukünftige Anwendungen
4		80H: Dito
5	End of Text	ETX
6		
7	Check	
8	End of Block	ETB

Mit diesem Telegramm wird dem Interface mitgeteilt, in welchem Modus sich die Maschine befindet. Dies ist notwendig, wenn das Interface eine Statistik über die Verkäufe führen soll, und damit informiert werden muss, ob ein Getränk z.B. in einem Testmodus ausgegeben wurde. Damit im

Interface eine komplette Statistik geführt werden kann, ist es notwendig, dass alle Verkäufe über das Interface abgewickelt werden, d.h. auch wenn die Maschine z.B. im Gratisbetrieb ist. Die Maschine darf somit Produktausgaben nicht direkt abwickeln, ohne das Interface zu informieren.

MMODE = 4 ist redundant mit der Meldung *VEND disable*. Der Slave darf keine Verkäufe zulassen, und muss die Zahlungssysteme sperren.

Hinweis: Die Betriebsarten Service und Gratis können unabhängig von VEND enable/ disable gebraucht werden. D.h. VEND disable stellt das Interface auf Blockiert, VEND enable behält hingegen den Zustand (Gratis, Service), ausser das Interface selber befinde sich im Blockiert Zustand.

Das Telegramm muss bei jedem Wechsel des Betriebszustandes von der Maschine gesendet werden. Ebenfalls beim Initialisieren (Power on) der Maschine oder des Interfaces (Antwort auf Statustelegramm beachten).

Bei Widersprüchen der Modis zwischen Maschine und Interface hat das Interface Priorität.

Tabelle mit den Modi bezogen auf das Vend Kommando:

Zustand alt	Kommando	Zustand neu
Blockiert	Vend disable	Blockiert
Blockiert	Vend enable	Bereit/ normal
Bereit/ normal	Vend enable	Bereit/ normal
Bereit/ normal	Vend disable	Blockiert
Service	Vend disable	Blockiert
Gratis	Vend disable	Blockiert
Gratis	Vend enable	Gratis

3.5.12 Parameter – Interface parametrisieren- (Level 3)

Master

1	Start	STX
2	Typ = E	'E'
3	D	Richtung: '0'=schreiben, '1'=lesen
4	P3	P: 3-stellige Parameter Nummer Hex
5	P2	Bereich: 001-0xFFFF
6	P1	P1 = LSB, p3 = MSB
7	V4	4-stelliger Wert Hex
8	V3	Bereich: 0001-0xFFFF
9	V2	
2	V1	v1 = LSB, v4 = MSB
6	End of Text	ETX
7		
8	Check	
9	End of Block	ETB

Antwort vom Slave

1	Acknowledge	ACK
1	Start	STX
2	Typ = E	'E'
3	X	Status: '0'=not supported '1'=saved '2'=error '3'=read_ok '4'=product index exceeded

4	V4	NUR BEI LESEN:
5	V3	4-stelliger Wert Hex
6	V2	Bereich: 0001-0xFFFF
7	V1	v1 = LSB, v4 = MSB
8	End of Text	ETX
9		
10	Check	
11	End of Block	ETB

Mit dem PARAMETER Kommando können Einstellungen im Interface vom Master oder einem PC Programm durchgeführt werden.

Jeder Parameter wird mit einer Nummer identifiziert. Die Liste der Parameter ist unten aufgeführt und kann erweitert werden. In zukünftigen Schnittstellen können neue Einstellungen definiert werden, welche von Interfaces anderen Typs nicht unterstützt werden. Dazu ist das Interface in der Lage, einen Fehler zurückzugeben, wenn entweder ein Speicherfehler aufgetreten ist, oder die Parameter Nummer nicht unterstützt wird.

Es wird vom Slave erwartet, dass es auf einen Schreibbefehl das notwendige unternimmt, um den Parameter sicher zu speichern. Der Master muss im Fehlerfall das Kommando (Schreiben misslungen, Antwort Status=2 – error) also nicht wiederholen (ausser bei NAK oder <silence>).

Umgekehrt muss der Master einen geschriebenen Parameter nicht mittels Lesebefehl kontrollieren. Der Slave hat bereits sichergestellt, dass der Parameter tatsächlich gespeichert ist.

3.5.12.1 Ablauf Schreiben

- Der Master sendet das PARAMETER Kommando mit der Parameter Nummer und dem dazugehörigen Wert.
- Das Interface antwortet mit ACK, prüft die Nummer und Wert und schreibt den Wert in den nicht-flüchtigen Speicher
- Das Interface meldet das Resultat im Byte x innerhalb TO_PS
x = '0' *not_supported*: die Parameternummer ist ungültig, bzw. wird vom Interface nicht unterstützt
x = '1' *saved*: das Interface konnte die Daten richtig schreiben
x = '2' *error*: das Interface konnte den Wert nicht speichern, oder der Parameterwert war unzulässig (Bereichsüberprüfung fehlgeschlagen)

3.5.12.2 Ablauf Lesen

- Der Master sendet das PARAMETER Kommando mit der Parameter Nummer. Der Wert ist nicht relevant.
- Das Interface antwortet mit ACK, prüft die Nummer und liest den Wert aus dem nicht-flüchtigen Speicher
- Das Interface meldet das Resultat im Byte x sowie die entsprechenden Daten innerhalb TO_DATA
x = '0' *not_supported*: die Parameternummer ist ungültig, bzw. wird vom Interface nicht unterstützt, oder der Lesebefehl wird nicht unterstützt
x = '3' *read_ok*: das Interface konnte die Daten richtig lesen und gibt die Daten zurück

Hinweis: Das Antwort Telegramm weist unterschiedliche Länge auf, je nachdem ob die Daten angehängt werden oder nicht.

3.5.12.3 Parameterliste

3.5.12.3.1 Dezimalstelle



Setzen der Dezimalpunkt Position für das Kommando CREDIT (Details und Anwendung siehe Kap. 3.5.6)

Parameternummer	Wert	Beschreibung
'001'	0x00	keine Dezimalstelle
	0x02	zwei Nachkommastellen (default)

3.5.12.3.2 Restkredit löschen (Münzprüfer)

Dieses Flag gibt an, ob das Interface einen allfälligen Restkredit nach einem Verkauf löscht, und damit eine Überzahlung verursacht. Andernfalls bleibt der Restbetrag stehen und kann durch erneute Wahl verbraucht werden.

Der aktuelle Saldo kann auch jederzeit vom Master mit dem CREDIT Kommando gelöscht werden (siehe Kap. 3.5.6)

Parameternummer	Wert	Beschreibung
'004'	0x00	Restkredit löschen
	0x01	Restkredit bleibt stehen (default)

3.5.12.3.3 Münzkanäle setzen (Münzprüfer)

Beim Betrieb mit parallelem Münzprüfer muss die Wertigkeit jedes Münzkanals im Interface hinterlegt sein. Diese muss abgestimmt sein, mit der Programmierung des Münzprüfers selber, d.h. die Kanäle müssen die gleiche Wertigkeit haben.

Ein spezieller Wert wird für einen Jeton verwendet. Weitere Info siehe Kap. 3.5.6

Wertebereiche:

- Münzen: '0000'-'FFF0'
- Einheitsjeton: 'FFFF'

Parameternummer	Wert	Beschreibung
'011'	0x00-0xFFFF	Wertigkeit Münzkanal A (default '000A')
'012'	0x00-0xFFFF	Wertigkeit Münzkanal B (default '0014')
'013'	0x00-0xFFFF	Wertigkeit Münzkanal B (default '0032')
'014'	0x00-0xFFFF	Wertigkeit Münzkanal C (default '0064')
'015'	0x00-0xFFFF	Wertigkeit Münzkanal D (default '00C8')
'016'	0x00-0xFFFF	Wertigkeit Münzkanal E (default '01F4')

3.5.12.3.4 Annahmefbegrenzung (Münzprüfer)

Dieser Wert begrenzt den Münzeinwurf beim eingestellten Betrag, indem der Münzprüfer gesperrt wird.

Parameternummer	Wert	Beschreibung
'040'	0x00-0xFFFF	Annahmefbegrenzung in der kleinsten Einheit. Bei 0 wird keine Limite überprüft. Bsp: '07D0' entspricht 20.00

3.5.12.3.5 Reserviert

Parameternummer	Wert	Beschreibung
'041' - '063'	-	Reserviert Hug-Witschi AG

3.5.12.3.6 Preis Download

Dieser Parameter ist identisch mit dem Kommando PRICE (Kapitel 3.5.7). Bei einer Überschreitung des maximalen Preisindex wird ein Fehler x=4 zurückgegeben.

Parameternummer	Wert	Beschreibung
'064' - '0C7'	0x00-0xFFFF	'064' = Wahl 1, '065' = Wahl..... 'C7' = Wahl 100

3.6 Kommunikationsablauf

Für ein Minimalprotokoll kann man sich auf die Kommandos VEND/MACHINE_MODE (Interface freischalten), INQUIRY (Verkaufsanfrage) und STATUS (Pollen) beschränken.

Die unten aufgeführten Abläufe sollen nicht starre Schemas sein, sondern Beispiele, wie sie häufig auftreten.

3.6.1 Initialisierung

Kaffeemaschine	Interface	Kommentar
1 Identification		[optional Typ und Art des Interfaces abfragen]
2	ACK+Daten	
3 Machine_mode		[optional den Mode initialisieren]
4	ACK+Daten	
5 Price		[optional Preise downloaden]
6	ACK	
7 Vend (c=1)		Zahlungssystem freischalten (enable)
8	ACK	
9 Status		Master beginnt mit dem Pollen
10	ACK	
11	Status (x=0)	keine weiteren Vorkommnisse
(fertig)		(warten bis zum nächsten Polling)

3.6.2 CCI & CSI: Grundzustand (kein Stift/Tag gesteckt)

Kaffeemaschine	Interface	Kommentar
1 Status		Master pollt das Interface
2	ACK	
3	Status (x=0)	keine besonderen Vorkommnisse
(fertig)		(warten bis zum nächsten Polling)

3.6.3 CCI & CSI: Produktetaste an der Kaffeemaschine betätigt

Es ist kein Stift bzw. kein VC gesteckt.

Kaffeemaschine	Interface	Kommentar
1 Inquiry (nnn=001, e=1)		Der Master will Produkt Nr. 1 (Preis Nr.1) abbuchen. Bei genügend Guthaben wird abgebucht.
2	ACK	
3	Inquiry (x=0)	nicht genügend Kredit um zu verkaufen, bzw. (noch) keine Freigabe von der Schankanlage
(fertig)		(warten bis zum nächsten Polling)

Falls der Preis 0 ist, wird immer abgebucht, weil die Bedingung Kredit \geq Produktpreis erfüllt ist.

3.6.4 Interface verkauft ein Produkt

Der Benutzer wirft Geld ein und will das Produkt beziehen.

Kaffeemaschine	Interface	Kommentar
1 Status		Master pollt das Interface
2	ACK	
3	Status (x=0)	keine Besonderen Vorkommnisse
4 Status		Master pollt das Interface
5	ACK	
6	Status (x=1)	Eine Münze eingeworfen oder VC gesteckt
7 Status		Master pollt das Interface
8	ACK	
9	Status (x=1)	Es ist Kredit vorhanden
10 Inquiry (nnn=021,e=1)		Taste wurde betätigt. Produkt Nr. 21 soll verkauft werden.
11	ACK	
12	Inquiry (x=0)	nicht genügend Kredit um zu verkaufen
13 Inquiry (nnn=021,e=1)		Master will weiterhin verkaufen.
14	ACK	
15	Inquiry (x=0)	nicht genügend Kredit um zu verkaufen
16 Status		Benutzer hat Taste losgelassen um mehr Geld einzuwerfen
17	ACK	
18	Status (x=1)	Es ist Kredit vorhanden
19 Inquiry (nnn=021,e=1)		Taste wurde wieder betätigt.
20	ACK	
21	Inquiry (x=1)	Jetzt genügend Kredit. Interface hat abgebucht
(fertig)		(warten bis zum nächsten Polling)

3.6.5 Interface verkauft ein Produkt; Kommando Wiederholungen

Gleicher Ablauf wie oben bei 3.6.4, aber mit Ausbleiben der Antwort und entsprechenden Wiederholungen.

Kaffeemaschine	Interface	Kommentar
1 Status		Master pollt das Interface
2	ACK	
3	Status (x=1)	Debitkarte vorgehalten, Saldo wird am Leser angezeigt
4 Inquiry (nnn=021,e=1)		Taste wurde betätigt. Produkt Nr. 21 soll verkauft werden.
5	<silence>	Antwort kann nicht gesendet werden innerhalb TO_CONFIRM.
6 Inquiry (nnn=021,e=1)		Wiederholung des Kommandos
7	ACK	
8	Inquiry (x=1)	Preis wurde abgebucht. Master kann Produkt starten. Der Preis wird nur einmal abgebucht, weil zwischen INQUIRIES keine STATUS Anfrage gesendet wurde.

9	Inquiry (nnn=021, e=1)	Fehler Maschine: Kommando darf nicht zweimal hintereinander gesendet werden, wenn die Antwort vom Interface korrekt übermittelt wird ausser sie hätte die Antwort vom Interface nicht verstanden.
10	ACK	
11	Inquiry (x=1)	Preis wurde schon zuvor abgebucht. Hier wird nur die Meldung wiederholt, ohne dass das Interface nochmals abbucht.
12	Status	Bestätigung, dass die INQUIRY Antwort verstanden wurde. Alle anderen Kommandos führen zu einer unerlaubten Situation (Verkauf ist nicht abgeschlossen)
13	<silence>	Antwort kann nicht gesendet werden innerhalb TO_CONFIRM
14	Status	Korrekte Wiederholung der STATUS Anfrage. Ein Wechsel des Kommandos würde auch zu einer unerlaubten Situation führen (Verkauf nicht abgeschlossen)
15	ACK	
16	Status (x=1)	Es ist Kredit vorhanden
	(fertig)	(warten bis zum nächsten Polling)

3.6.6 CSI: Interface verkauft ein Produkt

Der Kellner den Stift wird gesteckt und das Produkt bezogen.

Kaffeemaschine	Interface	Kommentar
1 Status		Master pollt das Interface
2	ACK	
3	Status (x=0)	keine besonderen Vorkommnisse
4 Status		Master pollt das Interface
5	ACK	
6	Status (x=1)	Stift gesteckt
7 Status		Master pollt das Interface
8	ACK	
9	Status (x=1)	Stift gesteckt. Schankanlage gibt die Freigabe.
10 Inquiry (nnn=015,e=1)		Taste wurde betätigt. Produkt Nr. 15 soll ausgegeben werden.
11	ACK	
12	Inquiry (x=1)	Produkt wird abgebucht (Relais gesetzt)
13 Status		Die Ausgabe läuft. Master beginnt weiter mit pollen
14	ACK	
15	Status (x=2)	Die Meldung an die Schankanlage ist noch nicht abgeschlossen
16 Status		Master beginnt weiter mit pollen
17	ACK	
18	Status (x=0)	Produkt fertig abgebucht. Kein Stift gesteckt.
19 Inquiry (nnn=035, e=1)		Eine andere Taste wurde betätigt
20	ACK	
21	Inquiry (x=0)	Kein Stift gesteckt
	(fertig)	(warten bis zum nächsten Polling)

3.6.7 CCI: Das Zahlungssystem antwortet nicht

Das Interface kann das Kreditsystem nicht pollen.

	Kaffeemaschine	Interface	Kommentar
1	Status		Master pollt das Interface
2		ACK	
3		Status (x=3)	Fehler seitens des Kreditsystems. Ein Inquiry würde mit x=0 beantwortet
	(fertig)		(warten bis zum nächsten Polling)

3.6.8 Identifikation anfragen

	Kaffeemaschine	Interface	Kommentar
1	Identification		Master fragt nach dem Innenleben des Interfaces
2		ACK	
3		Identification(i=2, tt=62, vvv=110, ll=02)	Interface-Typ=CCI, Cashless MDB, Software-Version Interface=v1.10, Level 2
	(fertig)		(warten bis zum nächsten Polling)

3.6.9 Preis laden

	Kaffeemaschine	Interface	Kommentar
1	Price (l=0, nnn=035, pppppp=000150)		Master sendet folgenden Preis zum Interface: Produktenr. 35, Preis 150 der Preisliste 0). Bei einem Dezimalpunkt von 2 sind das z.B. CHF 1.50)
2		ACK	
	(fertig)		(warten bis zum nächsten Polling)

3.6.10 CCI & CSI: Während des Pollens tritt eine Störung auf

Das Kommando muss wiederholt werden.

	Kaffeemaschine	Interface	Kommentar
1	Status		Master pollt das Interface
2		NAK	z.B. Checksumme stimmt nicht
3	Status		Master pollt das Interface
4		NAK	z.B. Kommando nicht im Befehlssatz enthalten
5	Status		Master pollt das Interface
6		ACK	ok
7		Status (x=0)	keine besonderen Vorkommnisse
	(fertig)		(warten bis zum nächsten Polling)

4 Hardwaredefinitionen



Im Folgenden sind die Steckertypen und -belegungen für die Interfaces definiert.

4.1 Stecker CSS: Maschine - Interface

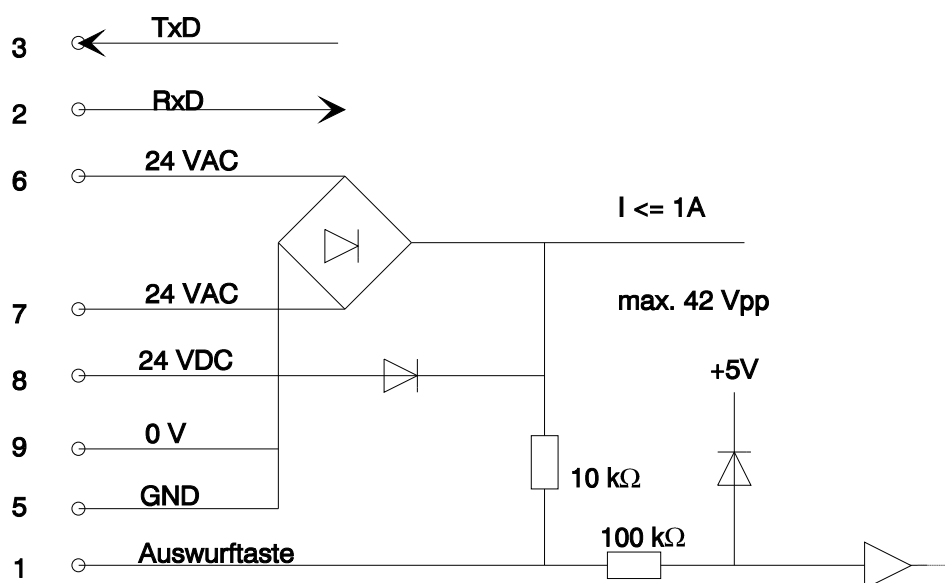
Typ: Sub DB9, Interfaceteil male

Der Maschinen Hersteller ist verpflichtet, diesen Stecker für den Anschluss von CCI/ CSI Geräten zur Verfügung zu stellen.

Bitte beachten: Die Steckerbelegung wurde derart ausgelegt, dass die Kaffeemaschine direkt an einen PC angeschlossen werden kann. Dabei muss aber die Speisung des Interfaces abgeklemmt werden!

KM:

IF:



Restliche Pins sind nicht belegt

Bitte beachten Sie die zulässigen Spannungstoleranzen der angeschlossenen Zahlungssysteme. Die 24V AC und DC werden möglicherweise unregelmäßig vom Interface an diese weitergegeben.

4.2 PWR: Speisung des Interfaces

Dieser Stecker wird alternativ für die Speisung des Interfaces ausgeführt.

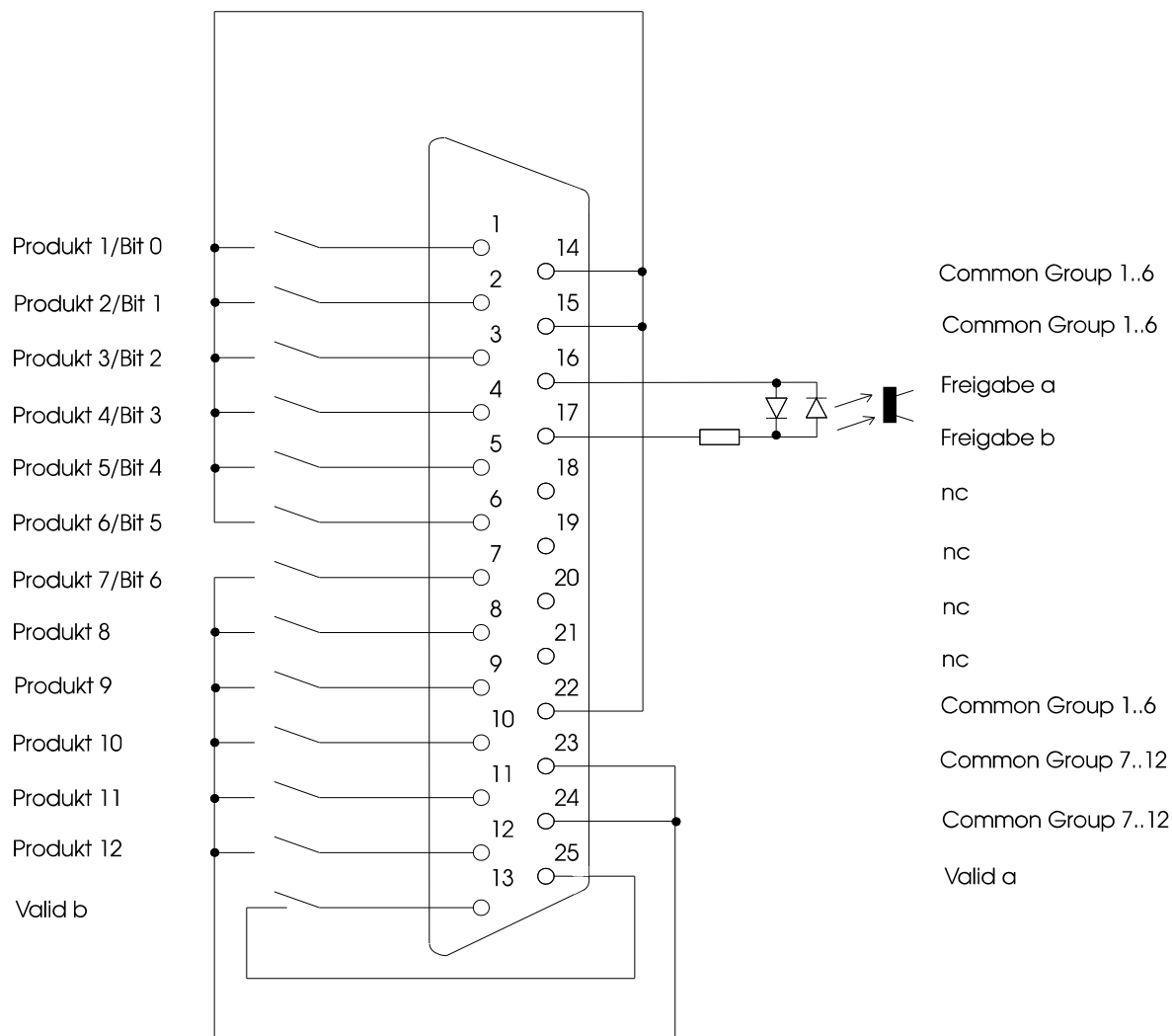
Typ: 4 Pol Molex Mini Fit 39-30-1040

Nicht empfohlen! Dieser Stecker hat kaum Verbreitung gefunden.

Pin 1	24 VAC
Pin 2	24 VAC
Pin 3	+24 VDC
Pin 4	GND (24 VDC)

4.3 CSI, Stecker PAR: Interface - Schankanlage

Typ: Sub D25, Interfaceteil male



Die Relaiskontakte sind aktiv closed.

Die Freigabe kann mit 12-36 V AC/DC betrieben werden, und ist ebenfalls aktiv high.

5 History

<i>Version</i>	<i>Änderungen:</i>	<i>Datum:</i>	<i>Status:</i>
1.4	erste offizielle Version		freigegeben
1.5	Erweiterungen für CCI-2: <ul style="list-style-type: none"> • Neues Telegramm MACHINE_MODE definiert • Telegramm IDENTIFICATION angepasst • Electronic-Purse berücksichtigt: Timeoutdefinition und STATUS-Telegramm erweitert • Levels eingeführt • diverse textliche Korrekturen 	2.9.96	
1.6	<ul style="list-style-type: none"> • IDENTIFICATION- Antwort erweitert für den Level 2 • STATUS: Flag JUST_RESET präziser definiert • STATUS, TO_PS limitiert • STATUS: <i>error Zahlungssystem</i> genauer definiert • Timeout ACK, NAK vergrössert • Timeout Rückmeldung mit Dateninhalt vergrössert und neu definiert. • Fehlermeldungen auf CREDIT • diverse textliche Anpassungen. Genauere Beschreibung der Themen <i>Telegrammtypen</i>, <i>STATUS</i>, <i>CREDIT</i> • Kontaktadresse eingefügt • Steckerbelegungen für das CCI-2 • MACHINE_MODE: Byte reserviert • Kontaktadresse des Herausgebers eingeführt • Layoutänderung, Überschriften etc. 	8.4.97	provisorisch
1.7	<ul style="list-style-type: none"> • Flag CREDIT_HIDDEN im Status eingefügt • Kleinere textliche Korrekturen 	23.4.98	
1.7a	<ul style="list-style-type: none"> • Korrektur bei der Beschreibung von TO_PS und dem Standard Timeout TO_DATA • Textliche Korrekturen im CREDIT Kommando • Adressen der Mitglieder angehängt 	5.6.98	Freigegeben
1.9	<ul style="list-style-type: none"> • Benutzer Interface via CCI Protokoll mit Ein- und Ausgabemöglichkeiten 		Nicht implementiert
2.0	<ul style="list-style-type: none"> • Korrektur und Anpassung diverser Teile an die DIVA Eigenheiten. • Spezifizierung und Präzisierung des Error Handlings 		
3.0	<ul style="list-style-type: none"> • neue Kommandos PARAMETER, REPORT, AMOUNT für Level3 eingeführt • nie implementierte Kommandos EXPANSION und DISPLAY_REQUEST aus Doku Version 1.9 und 2.0 gestrichen • Nachbesserung der Spec. im Kapitel 3.5.1 – Levels • Kapitel <i>Stromaufnahme...</i> gestrichen (nicht mehr aktuell) • Kapitel <i>Schnittstellen Interface – Schankanlage</i> gestrichen. Beschreibung gehört in das CSI Handbuch • Kapitel <i>Sleep Modus</i> gestrichen (Interface spezifisch) • CREDIT: Erweiterung Statistikabfrage 	November 2004	
3.1	<ul style="list-style-type: none"> • Kommando CREDIT: Einheitsjeton nur in Level 3 • PARAMETER: auch lesen der Einstellungen ist möglich, sowie Annahmefreigabe 0 geändert • reservierte Kommandotypen aufgelistet 	Dez. 04	
3.2	<ul style="list-style-type: none"> • kleinere Korrekturen 		

- | | | |
|-----|--|----------------|
| 3.3 | <ul style="list-style-type: none"> • neues Kommando <i>Report</i> gestrichen • Steckerbelegungen gestrichen, die nicht zur Norm gehören • Diverse Korrekturen | Freigegeben |
| 3.4 | <ul style="list-style-type: none"> • Kommando Amount: Typ der Antwort muss ‚B‘ sein nicht ‚E‘ • Kommando Parameter: D muss ASCII sein, nicht binär | |
| 3.5 | <ul style="list-style-type: none"> • Überarbeitung des gesamten Textes. Aktualisierung gemäss „heute gelebtem Standard“ • AMOUNT: Bereich Preis korrigiert. Null ist ebenfalls möglich. Bereich neu: ‚000000‘-‚999999‘ • Status Telegramm: Status modifiziert hinsichtlich den error Zuständen • Kapitel Übersicht. Disclaimer ergänzt | September 2015 |

6 Anhang

6.1 Disclaimer

Die CSNORM und die dort enthaltenen Informationen werden der Öffentlichkeit zugänglich gemacht. Durch das Herunterladen oder Kopieren von Inhalten, Bildern, Fotos oder anderen Dateien werden keinerlei Rechte übertragen.

Die Urheber- und alle anderen Rechte an Inhalten, Bildern, Fotos oder anderen Dateien mit und in dieser CSNORM gehören ausschliesslich der Hug-Witschi AG.

Für die Reproduktion jeglicher Elemente ist die schriftliche Zustimmung der Urheberrechtsträger im Voraus einzuholen.

Obwohl die Hug-Witschi AG mit aller Sorgfalt auf die Richtigkeit der veröffentlichten CSNORM achtet, kann hinsichtlich der inhaltlichen Richtigkeit, Genauigkeit, Aktualität, Zuverlässigkeit und Vollständigkeit dieser CSNORM und allen vorgängigen Versionen keine Gewährleistung übernommen werden.

Haftungsansprüche gegen die Hug-Witschi AG wegen Schäden materieller oder immaterieller Art, welche aus dem Zugriff oder der Nutzung bzw. Nichtnutzung der CSNORM, durch Missbrauch der Verbindung oder durch technische Störungen entstanden sind, werden ausgeschlossen.

Die Hug-Witschi AG behält sich ausdrücklich vor, jederzeit die Inhalte der CSNORM ohne Ankündigung ganz oder teilweise zu ändern, zu löschen oder zeitweise nicht zu veröffentlichen.

Jegliche Änderungen, Anpassungen der CSNORM dürfen nur durch den Eigentümer Hug-Witschi AG gemacht werden.

Copyright durch Hug-Witschi AG. Alle Rechte vorbehalten.

6.2 Kontaktadresse

Hug-Witschi AG	Tel. +41 31 740'44'44
Auriedstrasse 10	Fax. +41 31 740'44'45
CH-3178 Böisingen	eMail info@hugwi.ch
Schweiz	