CCI/CSI Protocol
Interface standard for beveridge dispensers and payment systems
Version 3.5

Status: released

Table of contents

1 Reading this documentation

1.1 Abbreviations

CCI	Coffee Credit Interface: synonym for payment systems
CSI	Coffee Standard Interface: synonym for dispensing systems (Beveridge dispensers) with parallel interface
VC	Value Carrier: RFID transponders (Mifare, Legic etc.)
PS	Payment System: payment system, which is connected with the CCI (serial or parallel)
CSS	Coffee Standard Serial: Connector on interface for the connection to the coffee machine
PAR	PArallel Relais: connector CSI interface with dispensing systems

1.2 Terms

Payment system	Cash- or cashless money or point system
Billing system	Device to read the sales statistics for the purpose of further processing
Interface	Slave device which communicates through CCI/CSI protocol
Machine	Synonym for coffee machine or vending machine

1.3 Number representation

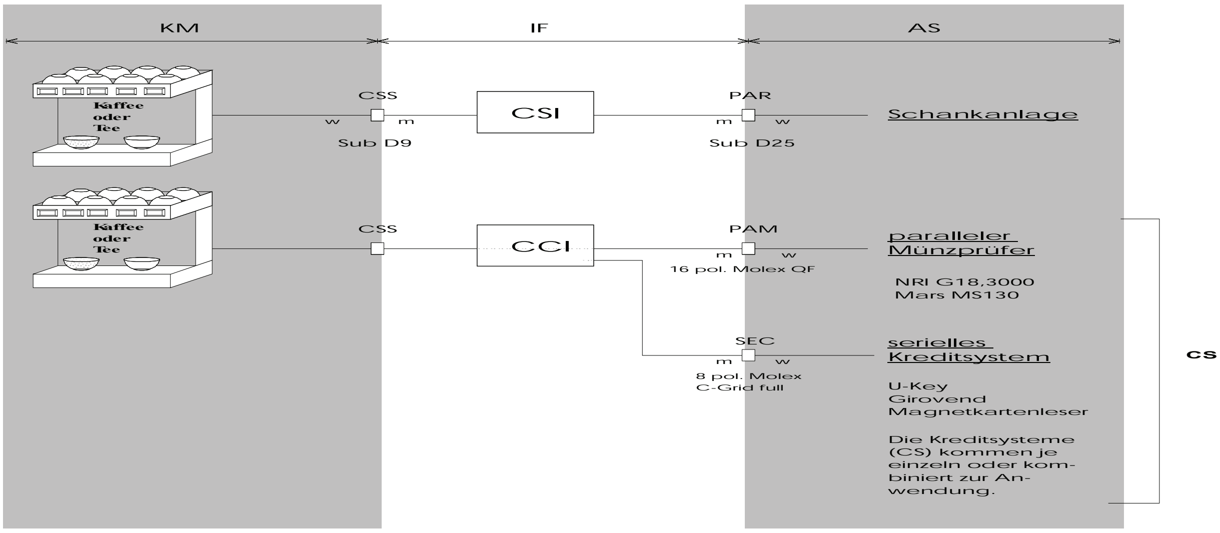
Not specifically noted figures are to be interpreted as decimal values. Hexadecimal numbers are represented in the form 0xyy.

ASCII characters are represented in quotation marks, for example 'A' for the letter A.

2 Overview

In the sense of a unification of interfaces to external payment systems a basis of the CCI/CSI Protocol was created in 1992 by well-known Swiss companies.

It is based on a RS232 interface with specification of the cable. The software protocol is a set of commands. The manufacturer of machines (often coffee machines) implement those commands, which they need to properly handle at least a minimum to handle vends. Some commands are used to increase convenience and can be omitted. The Protocol is so designed, that it doesn't matter which interface (CCI or CSI) is connected.



! A manufacturer wants an extension of this Specification, he is obliged, to inform the Publisher of this document (see annex).
Changes and extensions may only be done with the written consent of the Publisher of this Specification.

3 Software protocol coffee machine - interface

3.1 Introduction

The Protocol is designed as a master-slave communication. Each telegram from master will be acknowledged with ACK (telegram correct received) or NAK (error). With some commands the response contains data content. Each telegram is secured with a checksum. The entire transmission in done in ASCII characters.

3.2 Definitions

Master:	Coffee machine
Slave:	Interface

Transmission parameters:

Baud rate:	9600	In few older machines slower baud rates were used. For new implementations 9600 Baud is fixed. Other baud rates than 9600 are only yet to support backward compatibility.
Number of data bits:	8	
Parity	None	
Number of stop bits	1	

3.3 Protocol rules

Master:		Slave:	Comments:
	Call		- Master polls the interface.
	Confirmation		- Confirmation within TO_CONFIRM
	[*Feedback*]		- optional depending on the telegram. Timeout TO_DATA

All commands can be sent at anytime. It is therefore no certain order prescribed (exception see chapter 3.5.8).

The master should poll the slave regularly-between 100-500ms.

3.3.1 Timeouts

TO_CONFIRM	Confirmation with ACK or NAK:	200 ms
TO_DATA	Feedback with data content:	Default 5 s, i.e. for all commands except for INQUIRY. Rule for the command INQUIRY: The exact time will be reported in the STATUS command (chapter 3.5.5) in field TO_PS (depending on the payment system or by a setting in the interface).

During an active timeout the communication remains quiet, i.e. no other telegrams may be transferred. Nevertheless sent data is ignored.

3.3.2 Error Handling

Telegrams not understood by the slave can be handled in two ways:

- Send NAK
- < silence >, i.e. no answer send and wait for retry

< silence > can be looked as NAK and has the same effect. If clear errors happen, as described below, the use of NAK is preferred.

< silence >:

The slave "responds" with < silence >, because he is either not ready (i.e. not in operating mode) or the response within the agreed time out TO_CONFIRM cannot be sent. The slave must monitor the time TO_CONFIRM; delayed answers are not allowed.

The master repeats the same message typically 10 times, after the TO_CONFIRM timeout has been expired. As the answer will not follow after the end of the timeout, the master may immediately repeat the command.

If no response arrives after the named repetitions, the master should go back to the initialization and cancel the pending action. In addition the master issues an error message. More actions are specific and will not be prescribed here

It is an aim that the communication automatically established again, if the slave is ready again. To achieve that, the master must poll periodically e.g. every 10 seconds with VEND 0 or STATUS. If the slave answers back, the master may continue in accordance with the JUST_RESET flag.

NAK has the task to inform the counterpart about not understood commands such as incorrect check sum or incorrect structure (e.g. no ETB). Illogical or incorrect data content must be treated individually with each telegram, and are documented in the section of the telegrams.

A structured software design according to the OSI layer model helps to avoid a mixup of different error situations. ACK and NAK will be dealt by the datalink layer, and means that an ACK is sent if the telegram is correctly, but its contents on a higher layer is not understood.

If the master doesn't understand the response of the slave, he must also repeat the command. There is however no NAK mechanism here.

3.3.3 Communication examples

1. error-free communication:

Master :	Slave :	Comments:
Telegram 1		MASTER polls the interface
Acknowledge		positive acknowledgement

2. faulty communication with subsequent mutual agreement:

Master :	Slave :	Comments:
Telegram 2		Master polls the interface
not acknowledge		not understood, to be repeated
Telegram 2		Master repeats the request
not acknowledge		not understood, to be repeated
Telegram 2		Master repeats the request
Acknowledge		now understood

3. Slave busy:
   The slave is temporarily not ready.

Master :		Slave :	Comments:
	Telegram 2		Master polls the interface
	< silence >		Not ready, no reply
	Telegram 2		Master repeats the request
	< silence >		Not ready, no reply
	Telegram 2		Master repeats the request
	< silence >		Not ready, no reply
	Telegram 2		Master repeats the request
	Acknowledge		now ready

4. Slave disconnected:
   The slave is not in operation.

Master :		Slave :	Comments:
	Telegram 2		Master polls the interface
	< silence >		Device switched off, no response
	Telegram 2		Master repeats the request
	< silence >		Device switched off, no response
	Telegram 2		Master repeats the request
	< silence >		Device switched off, no response
	Telegram 2		Master repeats the request
	< silence >		Device switched off, no response
	Same sequence 10x		
	VEND 0		Master returns the initialization mode
	< silence >		Device switched off, no response
	VEND 0		Master returns the initialization mode
	Acknowledge		Interface was put into operation and responds again

5. Incomplete reply:

Master :		Slave :	Comments:
	STATUS		Master polls the interface
	Acknowledge		Telegram complete, checksum ok
	...		No data followed. Master waits TO_DATA
	STATUS		master repeats after TO_DATA + 100ms (margin)
	Acknowledge		Telegram complete, checksum ok
	Data response		now ok

3.4 Telegram structure

3.4.1 Calls and reponse:

Byte	With data	
1	Start	Start character (STX: 0x02)
2	Type	Command type
3	Data n	first data byte (most significant byte)
n + 3	Data 1	last data byte (least significant byte)
n + 4	End of text	ETX = 0x03
n + 5		Bcc: Block check character: hex value transmitted
n + 6	Check	in two ASCII-character
n + 7	End of block	ETB = 0 x 17

Byte	Telegram without data	
1	Start	Start character (STX: 0x02)
2	Type	Command type
3	End of text	ETX = 0x03
4		Bcc: Block check character: hex value transmitted
5	Check	in two ASCII-character
6	End of block	ETB = 0 x 17

3.4.2 Evaluation of the telegrams:

To keep the communication protocol expandable in terms of telegram lengths, it is recommended to check the control characters ETX or ETB and then begin evaluating.

The validity of the telegrams is ensured by the checksum.

3.4.3 Confirmations:

Byte	Acknowledge	
1	ACK	ACK = 0x06

Byte	not acknowledge	
1	NAK	NAK = 0 x 15

3.4.4 Checksum:

The bcc is calculated from a XOR combination of all previous characters without STX but including ETX:

< Bcc > = 0 XOR < char1 > XOR < char2 > ….XOR ETX.

Bcc is transmitted in two-byte ASCII characters.
Example: bcc 0x8A = transmitted as '8A', so 0 x 38 followed by 0x41.

3.4.5 Encoding of characters:

Each character (data) is transmitted as a decimal ASCII character

Example: the Number of 123 transmitted in three ASCII characters '1', '2', '3' (0 x 31, 0x32, 0x33). The highest-order digit is submitted first, the least significant digit last (big endian).

In bit fields (like the Telegram STATUS, byte IFSTAT and 2 reserved bytes) the high-order bit is set, in order to avoid special characters

3.4.6 Example

Byte	Command with data	Hex value
1	Start	0x02 (= STX)
2	Type = S	0 x 53 (='S')
3	Data 1 = 2	0 x 32 (= '2')
4	Data 2 = 9	0 x 39 (= '9')
5	Data 3 = 0	0 x 30 (= '0')
6	End of text	0x03 (= ETX)
7	Check	0 x 36 (= '6')
8		0x42 (= 'B')
9	End of block	0 x 17 (= ETB)

Note: The data content of the sample has no relevance.

3.5 Telegram types

3.5.1 Features of the Protocol: level

In order to extend the protocol levels are introduced. In principle, the selected level is determined by the master.

An interface that supports a higher level, the support of the lower levels is mandatory. Exception: new commands, which may inform the master that the command is not supported. This means, that optional commands, which are not supported by the slave, are responded with „not supported" accordingly the command specification. The Master must be prepared to this situation.

A machine which supports a deeper level than the interface cannot use the additional features.

In order to work with different levels, the following rules apply:

Level master :	*Level slave :*	*Description:*
x	>x	In the level (> x) enhanced answers are ignored by the master
x	<x	The master doesn't use telegrams of the higher level
x	x	No communication problems, as both have the same level

Unspecified telegrams correspond to level 1.

If the master opposed the rules sends a command, which latter doesn't support due to lower level, the interface responds only with ACK. The data response is not sent.

If the slave returns a higher level than is defined in the latest specification, the master refers to the highest known level.

3.5.2 Legacy problems with old designs

Notes to possible compatibility problems with old machines can be read in the version 3.4 of this specification.

3.5.3 Command Overview

Name	Level	Type	master data	slave data	Function
Vend	1	V	1	-	Billing device lock / release
Status	1	S	0	4	Polling the Slave
Credit	1	C	4	7	Balance and price inquiry
Price	1	P	10	-	load price of a price list into the slave
Inquiry	1	I	4	1	Sell product
Identification	1	X	-	6/8	Interfacetype and version
Machine_Mode	2	M	3	2	Set mode (free vend, test, block)
Parameter	3	E	9	1	Read and save setting in the interface
Amount	3	B	13	1	Debit amount
reserved	-	F	-	-	Future expansion
reserved	-	a	-	-	Commands used privately by manufacturers
ACK	1	-	-	-	Command understood
NAK	1	-	-	-	Command not understood
< silence >	1	-	-	-	No answer, because not ready

3.5.4 Vend (only CCI) - lock the interface -

Master

1	Start	STX
2	Type = V	'V'
3	Operating mode c	c: 0=disable; 1=enable
4	End of text	ETX
5	Check	
6		
7	End of block	ETB

Reply by the slave

1	Acknowledge	ACK

With this command the interface can be forced to release or lock the payment systems (for example for the acceptance of coins). If the master cannot distribute products (disorder, heating, free sale and others) the payment system should be locked.

The connected payment systems are locked after powering up the interface. In order to switch to operation mode, a VEND enable **absolutely** must be sent (or Machine_Mode enable). A Vend disable sent during operation causes the lock of the interface until a VEND enable is sent.

The influence of the VEND commands on the different interface conditions is shown in the table in the chapter 3.5.11 Machine_Mode (level 2) - switch interface mode -.

For new implementations, it is recommended to use the command MACHINE_MODE, instead VEND (see chapter 3.5.11 Machine_Mode (level 2) - switch interface mode -).

3.5.5 Status - State request to the interface, Polling-

Master

1	Start	STX
2	Type = S	' S '
3	End of text	ETX
4	Check	
5		
6	End of block	ETB

Reply by the slave

1	Acknowledge	ACK

1	Start	STX
2	Type = S	' S '
3	X	x = status: 0: no action 3: error payment system 1: ready 4: error interface 2: busy
4	IF_STAT	different States (see below)
5	TO_PS	Timeout_Payment system: response time to an INQUIRY telegram
6		
7	End of text	ETX
8	Check	
9		
10	End of block	ETB

The status request 'S' is used for polling .
A status request can always be sent independent of the current status. Price 0 in the interface: a vend is also possible at x = no action.

Meaning of the status message x:

- **no action:**
  CCI: No saldo or transponder present
  CSI: No waiter pin plugged.
  The interface is in the idle state.

- **ready:**
  CCI: The interface is ready for a product sales (balance > 0, value carrier available)
  CSI: A waiter pen was plugged.

- **busy:**
  CCI: not in use
  CSI: The message on the dispensing system via the relay passes.
  The interface can accept no further inquiry in this State. It must wait until the state changes.

error payment system:
CCI: disturbance payment system. This message is for display purposes only be used.
CSI: (not used)

In status *ready* maybe no sales are possible. Product inquiries (command 'I') can be answered with *credit low*.

- **error interface:**
  An error occurred in the interface so that its function is no longer guaranteed (partly or completely). A message should be displayed on the machine.

Meaning of the status message IF_STAT:

7	6	5	4	3	2	1	0
1	X	X	CREDIT_HIDDEN	KEN RESET	AB_GELD	SERVICE	FREE

FREE		CCI	The interface is set to free vend.
SERVICE		CCI	The interface is in the programming mode. Test sales can be processed in this state.
AB_GELD		CCI	The coin changer reports low tube coins. The machine should prompt a message.
JUST_RESET		CCI /. CSI	The interface has been restarted. The machine must perform a new initialization. This concerns the command STATUS, IDENTIFICATION, PRICE and VEND / MACHINE_MODE. The flag is reset after sending a VEND (see chapter Vend (only CCI)) or MACHINE_MODE, but only if a STATUS request has been performed since rebooting.
CREDIT_ HIDDEN	Level 2	CCI	The reported balance on the machine may not be displayed.
Bit7		CCI /. CSI	Always "1",
X (bit 5, 6)		CCI /. CSI	Reserved (Franke)

Meaning of the status message TO_PS:
The bookings process takes different time depending on the payment system. With this byte the interface indicates how long the answer to an inquiry can be delayed.

The time in seconds is calculated as follows: timeout [s] TO_PS - = '0'.
Exception: the value 80 H ('Ç') is equivalent to the default setting (see 3.3.1 Timeouts).
Allowed value range: 0 x 31 ('1') < TO_PS <. 0xC8 (equivalent to 1[1]... 152 (s).

[1] In section 3.3.1 Timeouts, the default timeouts are defined. The computational result can provide values which are less than the default timeout. It is advisable to set these smaller values to the default timeout, ie the default timeout is the minimum timeout , which is valid for Inquiry. Results greater than the default time-out "overwrite" this .

Example:

1 TO_PS = 'l' (0x6C)-> timeout = TO_PS - 0 x 30 = 60 seconds
2 TO_PS = '1' (0 x 31)-> timeout = TO_PS - x 30 = 1 second-> set default timeout 5 seconds = 0

Reserved byte (byte 6):

7	6	5	4	3	2	1	0
1							

CCI, CSI Always "1"

3.5.6 Credit (only CCI) - saldo and price request -

Master

1	Start	STX
2	Type C =	'C'
3	N3	three-digit article number (relevant only for exec = 1)
4	N2	
5	N1	
6	e	e = exec: 0: total balance) 1: product price to item number nnn 2: delete rest credit
7	End of text	ETX
8	Check	
9		
10	End of block	ETB

Reply by the slave

1	Acknowledge	ACK

1	Start	STX
2	Type C =	'C'
3	N6	Balance or price (6 digits).
4	N5	
5	N4	
6	N3	
7	N2	
8	N1	
9	p	Decimal point position
10	End of text	ETX
11	Check	
12		
13	End of block	ETB

With this command the actual balance is requested (exec = 0), or the unit price of the product from which the product number has been delivered (exec = 1). This data can be used for the display of the machine.
Further, the current credit can be deleted, i.e. after a certain time after a vend (to the detriment of the customer).
Article number: The numbering starts with 1. This interface must support at least as many selections as the machine has products.

'000000'-'FFFF00'	Product price or balance If The interface is set to freevend, all prices are reported with '000000'
'FFFFFF'	Article overflow: An article number is requested, which does not exist, because the range has been exceeded.
'FFFFFE' (Level 2) 'Sperr' (Level 1)	The requested product is locked and may not be sold
'FFFFFD'	The price cannot be calculated
'FFFFFC'	exec value invalid
'FFFFFB' (Level 3)	Unit token (exec=0): With the token any product can be purchased regardless of the price. The master should display an appropriate message, instead of the balance. After the vend, the balance is set to 0 (see section 3.5.12.3.3 Set coin channels (coin))

The decimal point position indicates the position of the point (from right):

p=0	no point available
p=2	two decimal places (CHF, EUR…)

If the balance must be deleted (exec=2), the new balance 0 must be returned, if this is possible (tags balance i.e. cannot be deleted) and supported by the interface.

3.5.7 Price (only CCI) -Price download-

Master :

1	Start	STX
2	Type = P	'P'
3	l	Price list 0-n
4	N3	Article number
5	N2	
6	N1	
7	P6	Price in the smallest unit
8	P5	
9	P4	
10	P3	
11	P2	
12	P1	
13	End of text	ETX
14	Check	
15		
16	End of block	ETB

Reply by the slave

1	Acknowledge	ACK

A Vend with the command *Inquiry* uses the prices that are configured in the interface. With the command *Price* it is possible to relocate the price management into the machine. *Price* allows to transfer the prices, that are stored in the machine into the interface.

The interface may return only *ACK* (understood) or *NAK* (not understood). The meaning of that is whether the telegram was received correctly (check sum, etc.) or not. With that, the content is not on proved. E.g. an **invalid product number cannot not marked with an error**.

For new implementations, the command AMOUNT (3.5.9 Amount – Value charge - (level 3)) is recommended instead of PRICE, if the interface holds no statistics (constraint multiple price lists in

AMOUNT). The command PARAMETER (3.5.12 Parameter – Configure Interface - (level 3)) implements the PRICE Function in an improved way (handshake see 3.5.12.3.6 Price download).
Value ranges:

• Price list	The number refers to the numbering in the Interface. 0: cash vends 1-n: cashless vends n is the highest pricelist that is supported by the interface
• Product number	Starting with 1
• Prices	value in the smallest currency unit

Properly received prices immediately are stored in memory and overwrite the existing settings.

The decimal point can be obtained with the CREDIT command.

3.5.8 Inquiry - Sale inquiry -

Master

1	Start	STX
2	Type I =	'I'
3	n	article number
4	n	
5	n	
6	e	exec: '0': only verification '1': withdraw if okay
7	End of text	ETX
8	Check	
9		
10	End of block	ETB

Reply by the slave

1	Acknowledge	ACK

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

With the command INQUIRY the product with the article number is attempted to vend (button- or selection number). The interface compares the balance with the corresponding price. These prices can be loaded into the interface via the command PRICE or configuration.

With exec=0 the interface only checks whether the credit is sufficient without deducting the price from a payment system. Because the verification is not processed directly in the payment system the result may be wrong.

When exec=1 the debiting is done (assuming enough balance available). The master starts the product delivery after reception of *credit okay*. *credit* low means that at the time of booking there was not enough balance.

For the reply with data content (not ACK or NAK) a different time-out applies (see 3.3.1 Timeouts).

Article number:

The valid range is dependent on the machine and interface and is not further defined here.

In case of a request with an invalid product number, *credit okay* is returned. You get the same answer if the product is blocked.

Safeguarding in case of response not understood:
To avoid multiple charges for not understood responses to the INQUIRY command, after the INQUIRY always the command STATUS must follow. The interface interprets this as a receipt for the response *credit okay*.

If the machine once didn't understand this response, it will again send the command INQUIRY. Because the STATUS request misses (between the two INQUIRY) the interface this time will not withdraw the price, but only repeat again the answer.

The same mechanism applies, if INQUIRY *only verification* is requested.

The mechanism in case of failure with repeats obeys the rules as described in the chapter 3.3.2 Error Handling. An explicit example of possible error case is on page 24 apparent.

3.5.9 Amount – Value charge - (level 3)

Master

1	Start	STX
2	Type B =	'B'
3	N3	n: article number
4	N2	Range: '001'-'999'. (not supported: '000')
5	N1	n1 = LSB, MSB = n3
6	P6	The debit amount
7	P5	p1=LSB, MSB=p6
8	P4	Range:'000000'-'999999'
6	P3	
7	P2	
8	P1	
9	e	exec: 1: only verification 0: withdraw if okay Note: Compared to the inquiry command the meaning of exec is swapped!
2	l	Reserve (default '0')
2	m	Reserve (default '0')
6	End of text	ETX
7	Check	
8		
9	End of block	ETB

Reply by the slave

1	Acknowledge	ACK

1	Start	STX
2	Type B =	'B'
3	x	Status: '0'=credit low '1'=credit okay
7	End of text	ETX
8		

9	Check	
10	End of block	ETB

With the AMOUNT command a certain amount is debited from the interface. This command is especially useful when the master manages the prices.

The article number (selection) is used for statistical purposes when the interface holds a sale statistic. The master may send the article number correctly or fill '000' (number not supported).

x	Description
'0'	*credit low* : The available balance is not enough to charge the amount.
'1'	*credit okay*: With exec=0 the amount has been charged, the product can be delivered. With exec=1 the balance would be sufficient to charge the amount p

Important: Equal to the INQUIRY command, also after each AMOUNT a STATUS must follow (see section 3.5.8 Inquiry - Sale inquiry -)

3.5.10 Identification - Request interface data -

Master

1	Start	STX
2	Type = X.	'X'
3	End of text	ETX
4	Check	
5		
6	End of block	ETB

Response from the Slave

1	Acknowledge	ACK

1	Start	STX
2	Type = X.	'X'
3	i	i = interface type: 0: reserved 1: CSI (Coffee Standard Interface) (level 1) 2: type CCI (Coffee Credit Interface), i.e. an interface for managing payment systems 3: GABI (GAstro Bedienungsinterface) 4-6: reserved for third-party systems 7: MSS (Melitta System Service) 8: Data logger (reduced command set) 9, a...: reserved for new CCI/CSI applications
4	T2	TT = payment system: [12]
5	T1	00: coin 60: MDB coin changer 61: MDB bill validator 62: MDB card reader For more payment systems refer to the manual of the corresponding interface
6	v3	vvv = interface SW version
7	v2	Example: vvv=261 (v2. 61). Longer version numbers are reduced to three digits (E.g. 2.90.6-> 290)
8	v1	

9	L2	Level: Functionality of the implemented specification	L2: bit mask which features are implemented.
10	L1		l1: 01=level 1 02=level 2 03=level 3 00, 04..09: reserved for future expansion
11	End of text	ETX	
12	Check		
13			
14	End of block	ETB	

With the command IDENTFICATION information on the type and configuration of the interfaces are requested. These can be evaluated in the master (E.g. for an automatic machine configuration).

The payment system tt is specified only for compatibility reasons, but has lost importance. The CSI sends back in tt always 99.

i	Interface type	Description
'1'	CSI	reduced command set
'2'	CCI	complete command set with level 1-3

3.5.11 Machine_Mode (level 2) - switch interface mode -

Master

1	Start	STX
2	Type = M	' M '
3	MMODE	Vend mode of the machine: 1: normal operation 2: free vend 3: Test/service mode 4: Machine is out of order 5: Service + Data_Entry (level 3)
4		reserved HGZ (default value '0'))
5		80 H reserved for future applications
6	End of text	ETX
7	Check	
8		
9	End of block	ETB

Reply by the slave

1	Acknowledge	ACK

1	Start	STX
2	Type = M	' M '
3		0: reserved for future use
4		80 H: Ditto
5	End of text	ETX
6	Check	
7		
8	End of block	ETB

With this telegram the interface communicates the mode in which the machine is located. This is necessary if the interface holds statistics, and thus must be informed whether a beverage has been

issued in test mode.
For a complete statistic in the interface, it is necessary that all sales are processed through the interface, i.e. also when the machine is in free vend. The machine must not directly handle product deliveries without informing the interface.

MMODE = 4 is redundant with the message VEND *disable*. The slave must allow no sales, and must lock the payment systems.

Note: The operating modes service and free can be used independently from VEND enable/disable. I.e. VEND *disable* blocks the interface, VEND *enable* however keeps the state (free, service) except the interface itself is in the state blocked.

The telegram must be sent with every change of the operating status of the machine. Also when initializing (power on) the machine or the interfaces (see response to status telegram).

In case of contradictions of the modes between machine and interface, the interface has priority.

Table modes related to vend command:

Condition old	**Command**	**New condition**
Blocked	Disable Vend	Blocked
Blocked	Enable Vend	Ready / normal
Ready / normal	Enable Vend	Ready / normal
Ready / normal	Disable Vend	Blocked
Service	Disable Vend	Blocked
Free	Disable Vend	Blocked
Free	Enable Vend	Free

3.5.12 Parameter – Configure Interface - (level 3)

Master

1	Start	STX
2	Type = E	'E'
3	D	Direction: '0'=write, '1'=read
4	P3	P: 3-digit parameter number hex
5	P2	Range: 001-0xFFF
6	P1	P1 = LSB, MSB = p3
7	V4	4-digit value hex
8	V3	Range: 0001-0xFFFF
9	V2	
2	V1	v1=LSB, MSB=v4
6	End of text	ETX
7	Check	
8		
9	End of block	ETB

Reply by the slave

1	Acknowledge	ACK

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

4	V4	ONLY WHEN READING:
5	V3	4-digit value hex
6	V2	Range: 0001-0xFFFF
7	V1	v1=LSB, MSB=v4
8	End of text	ETX
9	Check	
10		
11	End of block	ETB

With the command PARAMETER settings in the interface can configured by the master or a PC program.

Each parameter is identified by a number. The list of parameters is listed below and can be extended. New settings can be defined in future interfaces, which are not supported by interfaces of other type. To do so the interface may return an error, if either a memory error has occurred, or the parameter number is not supported.

On a write command it is expected by the slave, that it takes all necessary tasks to securely store the parameter. The master must not repeat the command in case of an error (write failed, response status=2-error) except for NAK or < silence >.

Conversely the master must not check written parameters using read command. The slave has already made sure that the parameter is actually stored.

3.5.12.1 Process writing

- The master sends the PARAMETER of command with the parameter number and the corresponding value.
- The interface responds with ACK, checks the number and value and writes the value to the non-volatile memory
- The interface reports the results in the byte x within TO_PS
  x = '0' *not_supported:* the parameter number is invalid or is not supported by the interface
  x = '1' *saved*: the interface could write correctly the data
  x = '2' *error*: the interface could not save the value, or the parameter value was not allowed (failed range validation)

3.5.12.2 Process reading

- The master sends the parameters command with the parameter number. The value is not relevant.
- The interface responds with ACK, checks the number and reads the value from the non-volatile memory
- The interface reports the results in the byte x and the corresponding data within TO_DATA
  x = '0' *not_supported:* the parameter number is invalid or is not supported by the interface, or the read command is not supported
  x = '3': 'read_ok'', the interface could read the data correctly, and returns the data

Note: The answer telegram has different length, depending on whether the data are appended.

3.5.12.3 Parameter list

3.5.12.3.1 Decimal place

Set the decimal point position for the command CREDIT (details and application see chapter 3.5.6 Credit (only CCI) – saldo and price request -)

Parameter number	Value	Description

'001'	0x00	no decimal place
	0x02	2 decimal (default)

3.5.12.3.2 delete remaing credit (coin)

This flag specifies whether the interface deletes any residual credit after a sale, and thus caused an overpayment. Otherwise, the balance remains and can be consumed by another selection.
The current balance can be deleted at any time by the master with the CREDIT command (see chapter 3.5.6 Credit (only CCI) – saldo and price request -)

Parameter number	Value	Description
'004'	0x00	Delete remaining credit
	0x01	Residual credit will stop (default)

3.5.12.3.3 Set coin channels (coin)

When using a parallel coin acceptor, the value of each coin channel in the interface must be deposited. This must be agreed with the programming of the coin acceptor, i.e. the channels must have the same value.
A special value is used for a token. More info see chapter 3.5.6 Credit (only CCI) – saldo and price request -)

Value ranges:

- Coins: '0000'-'FFF0'
- Unit token: 'FFFF'

Parameter number	Value	Description
'011'	0 x 00-0xFFFF	Value coin channel A (default '000A')
'012'	0 x 00-0xFFFF	Value coin channel B (default '0014')
'013'	0 x 00-0xFFFF	Value coin channel B (default '0032')
'014'	0 x 00-0xFFFF	Value coin channel C (default '0064')
'015'	0 x 00-0xFFFF	Value coin channel D (default ' 00C8')
'016'	0 x 00-0xFFFF	Value coin channel E (default ' 01F4')

3.5.12.3.4 Acceptance limit (coin)

This value limits the coin acceptance to the set amount by locking the coin acceptor.

Parameter number	Value	Description
'040'	0 x 00-0xFFFF	Accepting limit in the smallest unit. With 0, no limit is checked. Example: ' 07D0' is equivalent to 20.00

3.5.12.3.5 Reserved

Parameter number	Value	Description
'041' - '063'	-	Reserved hug-Witschi AG

3.5.12.3.6 Price download

This parameter is identical to the command PRICE (chapter 3.5.7 Price (only CCI) -Price download-). If this exceeds the possible price index, an error is x = 4 returned.

Parameter number	Value	Description
'064' - ' 0C7'	0 x 00-0xFFFF	'064' = selection 1, '065' selection =... 'C7' = 100

3.6 Communication process

For a minimal Protocol one can restrict to the commands VEND/ MACHINE_MODE (interface to unlock), INQUIRY (vend request) and STATUS (Poll).

The below sequences should be no rigid schemes, but often occured examples.

3.6.1 Initialization

	Coffee machine	Interface	Comment
1	Identification		[optional query type of the interface]
2		ACK + data	
3	Machine_mode		[optionally initialize the mode]
4		ACK + data	
5	Price		[optional download prices]
6		ACK	
7	Vend (c=1)		unlock payment system (enable)
8		ACK	
9	Status		Master starts with polling
10		ACK	
11		Status (x = 0)	no further incidents
	(finished)		(wait until the next polling)

3.6.2 CCI & CSI: idle state (No Pin/tag inserted)

	Coffee machine	Interface	Comment
1	Status		Master polls the interface
2		ACK	
3		Status (x = 0)	no unusual occurrences
	(finished)		(wait until the next polling)

3.6.3 CCI & CSI: product button pressed on the coffee machine

There is no PIN or no VC.

	Coffee machine	Interface	Comment
1	Inquiry (nnn=001, e=1)		The Master wants to deduct no. 1 (price No. 1). With enough balance, the amount will be deducted.
2		ACK	
3		Inquiry (x=0)	not enough credit or (still) no release of the dispensing system
	(finished)		(wait until the next polling)

If the price is 0, the interface always will release the product.

3.6.4 Interface sells a product

The user throws in money and wants to obtain the product.

	Coffee machine	Interface	Comment
1	Status		Master polls the interface
2		ACK	
3		Status (x=0)	no unusual occurrences
4	Status		Master polls the interface
5		ACK	
6		Status (x=1)	A coin inserted or plugged VC
7	Status		Master polls the interface
8		ACK	
9		Status (x=1)	There is credit
10	Inquiry (nnn=021, e=1)		Button has been pressed. Product no. 21
11		ACK	
12		Inquiry (x=0)	not enough credit to sell
13	Inquiry (nnn=021, e=1)		Master wants to continue to sell.
14		ACK	
15		Inquiry (x=0)	not enough credit to sell
16	Status		User has released button to throw more money
17		ACK	
18		Status (x=1).	There is credit
19	Inquiry (nnn=021, e=1)		Press the button again
20		ACK	
21		Inquiry (x=1).	Now enough credit. Interface has debited
	(finished)		(wait until the next polling)

3.6.5 Interface sells a product; command retries

Same sequence as above at 3.6.4 Interface sells a product, but with a lack of response and corresponding repetitions.

	Coffee machine	Interface	Comment
1	Status		Master polls the interface
2		ACK	
3		Status (x=1)	debit card presented, saldo is shown on the display
4	Inquiry (nnn=021, e=1)		Button has been pressed. Product no. 21
5		< silence >	Reply cannot be sent within TO_CONFIRM.
6	Inquiry (nnn=021, e=1)		Retry the command
7		ACK	
8		Inquiry (x=1)	Price was deducted. Master starts product delivery. **The price is charged only once, because between INQUIRIES no STATUS request was sent**
9	Inquiry (nnn=021, e=1)		**Error machine** : Command must not sent twice in a row when the answer is properly submitted by the interface except it has not understood the response from the interface.
10		ACK	
11		Inquiry (x=1).	Price was charged earlier. Here, only the message repeats without that deducts the interface again.
12	Status		Confirm that the INQUIRY was sunk in response. **All other commands result in an illegal situation (sale is not completed)**
13		< silence >	Reply cannot be sent within TO_CONFIRM
14	Status		Correct repetition of the STATUS request. A change of the command would also cause an illicit situation (sale not completed)
15		ACK	
16		Status (x=1)	There is credit
	(finished)		(wait until the next polling)

3.6.6 CSI: interface sells a product

The waiter inserts the Pen and then applies the product.

	Coffee machine	Interface	Comment
1	Status		Master polls the interface
2		ACK	
3		Status (x = 0)	no unusual occurrences
4	Status		Master polls the interface
5		ACK	
6		Status (x = 1).	Pen set
7	Status		Master polls the interface
8		ACK	
9		Status (x = 1).	Pin inserted. Dispensing system is released.
10	Inquiry (nnn=015, e=1)		Button has been pressed. Product no. 15 should be issued
11		ACK	
12		Inquiry (x = 1).	Product is charged (relay used)

13	Status		The delivery runs. Master continues with polling
14		ACK	
15		Status (x = 2)	The message to the dispensing system is not yet completed
16	Status		Master polls
17		ACK	
18		Status (x = 0)	Product delivery finished. No Pen inserted anymore
19	Inquiry (nnn=035, e=1)		Another key is pressed
20		ACK	
21		Inquiry (x = 0)	No Pen inserted
	(finished)		(wait until the next polling)

3.6.7 CCI: the payment system does not respond

The interface cannot poll the credit system

	Coffee machine	Interface	Comment
1	Status		Master polls the interface
2		ACK	
3		Status (x=3)	Error on the part of the credit system. An inquiry would be answered with x=0
	(finished)		(wait until the next polling)

3.6.8 Identification request

	Coffee machine	Interface	Comment
1	Identification		Master asks about the inner workings of the interface
2		ACK	
3		Identification(i=2, TT =62, vvv = 110, LL=02)	Interface type=CCI, cashless MDB, software version interface=v1.10, level 2
	(finished)		(wait until the next polling)

3.6.9 Download price

	Coffee machine	Interface	Comment
1	Price (l=0, nnn=035, pppppp=000150)		Master sends following price to the interface: Product no. 35, price 150 of price list 0. With a decimal point of 2 are E.g. (EUR 1.50)
2		ACK	
	(finished)		(wait until the next polling)

3.6.10 CCI & CSI: a disturbance occurs during pollen

The command must be repeated.

	Coffee machine	Interface	Comment
1	Status		Master polls the interface
2		NAK	E.g. checksum is wrong
3	Status		Master polls the interface
4		NAK	E.g. command not include in the command set
5	Status		Master polls the interface
6		ACK	OK
7		Status (x=0)	no unusual occurrences
	(finished)		(wait until the next polling)

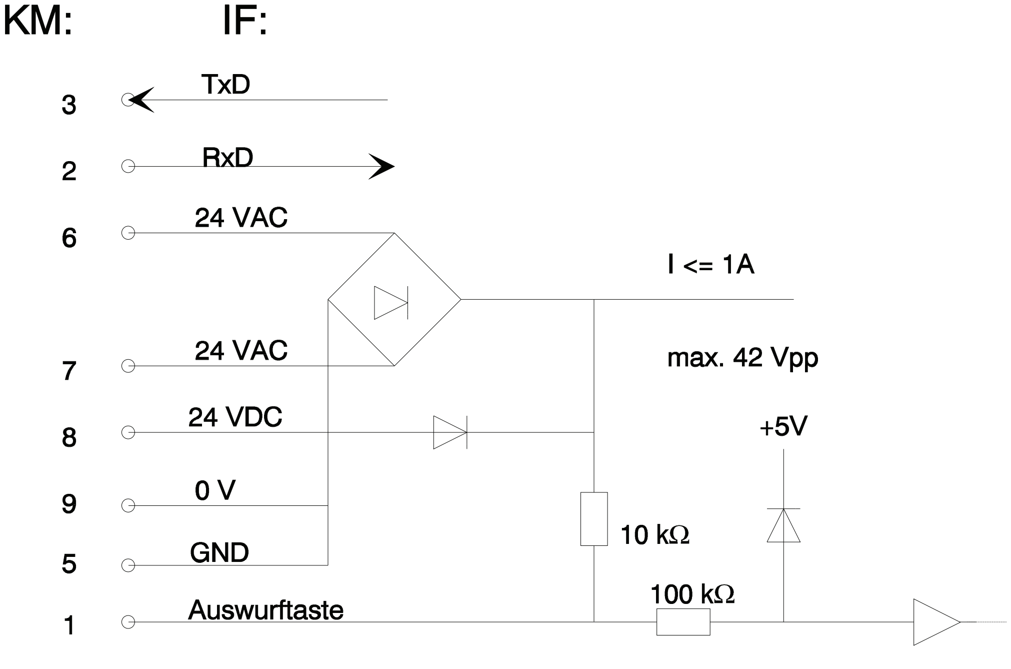
4 Hardware definitions

The connector types and-pinouts for the interfaces are defined in the following:

4.1 Plug CSS: machine - interface

Type: Male Sub DB9, interface part male

The machine manufacturer is obliged to provide this connector for the connection of CCI / CSI devices.
Please note: The pin assignment was designed so that the coffee machine can be connected directly to a PC. Doing so, the power of the interface must be disconnected!



Remaining pins are not assigned.
Please note the max. Voltage of connected payment systems. The 24V AC and DC maybe are propagated unregulated by the interface to these.

4.2 PWR: power of the interface

This plug is executed alternatively for the power supply of the interface.

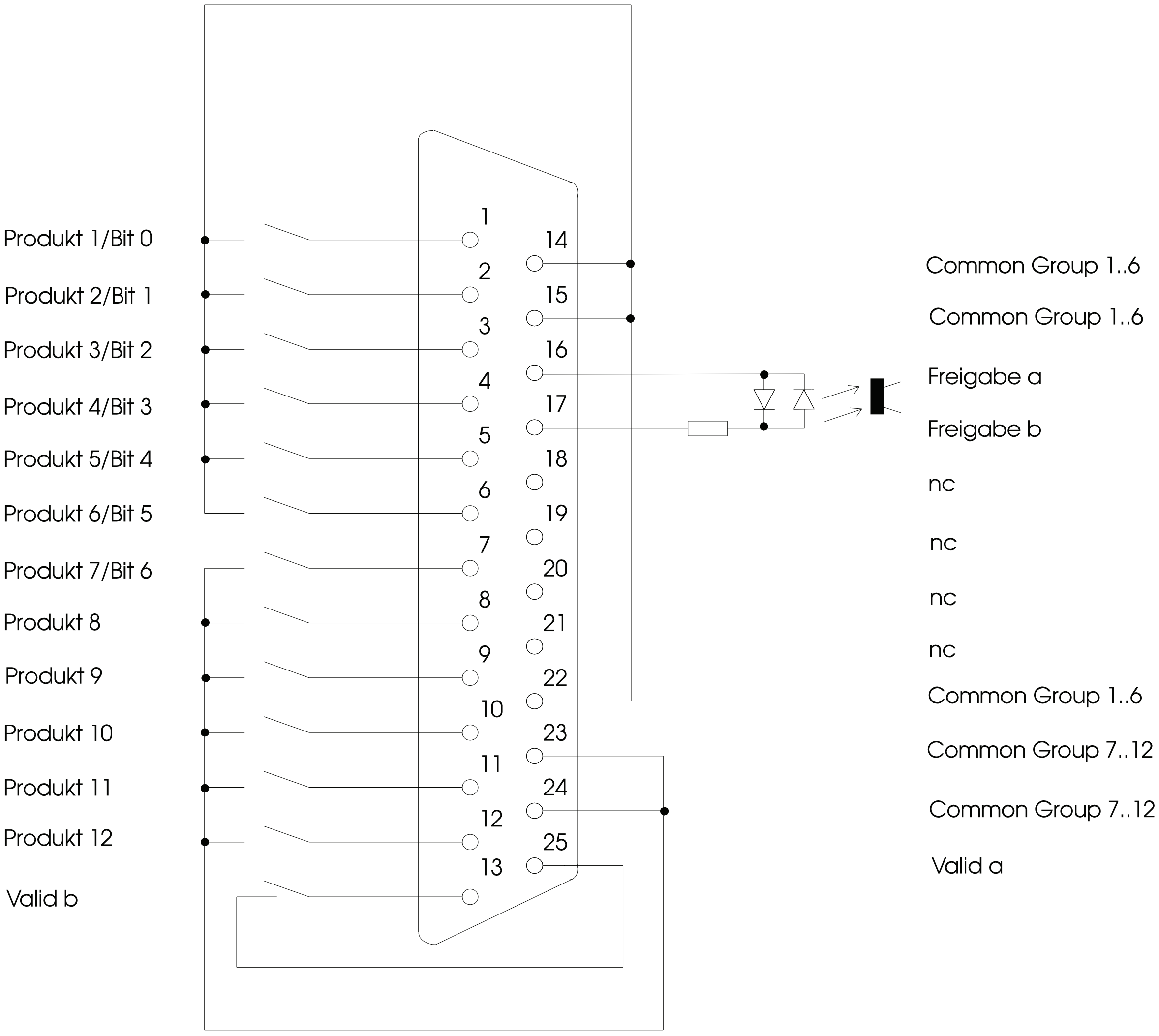
Type: 4 pole Molex mini fit 39-30-1040.

Not recommended! This plug can rarely be found in actual products.

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

4.3 CSI, plug PAR: interface - dispensing system

Type: Male sub DB25, interface part male



The relay contacts are active closed.

The release signal can be operated with 12-36 V AC/DC, and is also active high~~.~~

5 History

Version	*Changes:*	*Date:*	*Status:*
1.4	first official version		released
1.5	Extensions for CCI-2: • Defines new telegram MACHINE_MODE • Adapted telegram IDENTIFICATION • Electronic purse included: extended Timeoutdefinition and STATUS data telegram • Levels introduced • several textual corrections	2.9.96	
1.6	• IDENTIFICATION-response extended to the level 2 • STATUS: Flag JUST_RESET defines more precisely • STATUS, TO_PS limited • STATUS: *error payment system* more precisely defined • Timeout ACK, expanded NAK • Timeout feedback with data contents expanded and defined new. • Errors on CREDIT • various textual adjustments. More detailed description of topics *telegram types, STATUS, CREDIT* • Contact address • Pin assignments for the CCI-2 • MACHINE_MODE: Byte reserved • Contact address of the editor introduced • Layout change, headings etc.	8.4.97	Provisional
1.7	Added flag CREDIT_HIDDEN in the status Minor textual corrections	23.4.98	
1.7A	• Correction of description by TO_PS and the default timeout TO_DATA • Textual corrections in the CREDIT command • Addresses of members appended	5.6.98	Released
1.9	User interface via CCI Protocol with input and output options		Not implemented
2.0	• Correction and adaptation miscellaneous parts to the DIVA peculiarities. • Specification and clarification of the error handling		
3.0	• introduced new command parameters, REPORT, AMOUNT to Level3 • never implemented commands EXPANSION and DISPLAY_REQUEST of documentary version 1.9 and 2.0 • Rectification of the spec. in the chapter 3.5.1 -Level • Chapter *current consumption...* deleted (no longer current) • Chapter *interfaces interface - dispensing system* deleted. Description belongs in the CSI Handbook • Chapter *sleep mode* deleted (interface specific) • CREDIT: Extension statistics query	November 2004	
3.1	• Command CREDIT: unit chip only in level 3 • PARAMETERS: also read the settings is possible, as well as acceptance limit 0 changed • reserved command types listed	Dec. 04	

3.2	minor fixes		
3.3	• new command *report* deleted • Pin assignments are deleted, does not belong to the standard • Various corrections		Released
3.4	• Command amount: Type which must answer 'B' not 'E' be • Command parameter: D-must be ASCII, not binary		
3.5	• Revision of the text. Update according to "today lived standard" • AMOUNT: range fixed price. NULL is also possible. New range: '000000'-' 999999' [MW6] • Status telegram: status modified with regard to the error conditions • Chapter overview. Added Disclaimer	November 2015	

6 Addendum

Disclaimer

The CSNORM and verified the information provided to be made available to the public. By downloading or copying of texts, illustrations, photos or other files shall not transfer any.

The copyrights and all other rights relating to texts, illustrations, photos or other files and in this CSNORM belong exclusively to Hug-Witschi AG.

For Any reproduction the written consent of the copyright holder in advance must be obtained. Although the Hug-Witschi AG takes all possible care to ensure the accuracy of the published CSNORM, no guarantee can be taken over for the correctness, accuracy, timeliness, reliability or completeness of this CSNORM and all antecedent versions.

Liability claims against the Hug-Witschi AG for material or immaterial nature arising from access to, use or non-use of CSNORM, through misuse of the connection or technical breakdowns are excluded.

The Hug-Witschi AG expressly reserves the right at any time to change the contents of CSNORM without notice wholly or partly, to delete or temporarily not to publish.

Any changes, adjustments may be made CSNORM only by the owner Hug-Witschi AG.

Copyright by Hug-Witschi AG. All rights reserved.

Address

Hug-Witschi AG
Auriedstrasse 10
CH-3178 Bösingen
Switzerland

Tel. + 41 31 740'44 ' 44
Fax. + 41 31 740'44 ' 45
eMail info@hugwi.ch

[1] This time was compared to the spec. v1. 5 and older doubled.
[2] Doubled this time, because now payment systems can be connected to 1 s (previous timeout) when posting from exceed.
[6] not all States are supported by all interfaces.
[9] In the chapter 3.3.1 the default timeouts are defined. The calculated result can deliver values, which is below the DefaultTimeout. It is advisable, the default timeout to put on these smaller values, i.e. the default timeout is the minimum timeout, which is valid for INQUIRY. Results greater than the default time-out "override" this.
[12] The list can be expanded without changing the documentation.