

Data description BC II

Basic Class II

6.111.98.0.01.04

Version R1

Alle Rechte vorbehalten
All rights reserved
Tous droits réservés
Reservados todos los derechos
Tutti i diritti riservati
© 03 / 2009
Bizerba GmbH & Co. KG,
72336 Balingen
Postfach 10 01 64
72301 Balingen/Germany,
Tel. (+49 7433) 12--0, Fax (+49 7433) 12--2696
e--mail: marketing@bizerba.com
Internet: <http://www.bizerba.com>

TABLE OF CONTENTS

Page

1	GENERAL	1-1
1.1	Warranty.....	1-1
1.2	Supplementary documentation.....	1-1
1.3	General modifications.....	1-2
1.4	Changes in field names.....	1-2
1.5	Program transparency.....	1-3
1.6	Changes in the BCII database	1-4
2	PROGRAMS.....	2-1
2.1	Communication programs	2-1
2.2	Help program.....	2-2
3	FILE DESCRIPTION.....	3-1
3.1	HDLC- File structure.....	3-1
3.1.1	File frame	3-5
3.1.2	Useful data	3-6
3.2	HDLC record structure	3-7
3.3	ScLSV1, ScTCP, and ScXML record structure	3-8
3.3.1	Example of a dialog.....	3-10
3.4	Topology.....	3-13
4	DATA RECORD DESCRIPTION	4-1
4.1	Table overview in BCII	4-1
5	MASTER DATA.....	5-1
5.1	ATST gen. Texts master data	5-1
5.2	B2G1 global settings	5-2
5.3	B2LG label configuration for BCII.....	5-4
5.4	COST Barcode data.....	5-9
5.5	ETST Labels	5-10
5.6	FOST Forms.....	5-12
5.7	KFST Header/footer text master data	5-14

5.8	LOST Logos	5-15
5.9	PLST PLU master data	5-16
5.10	FUGL operator key arrangement	5-21

6 CONFIGURATION DATA 6-1

6.1	B2AG general operational procedure	6-1
6.2	B2MS Multiple operator sales	6-4
6.3	B2PL Setting for price labeling	6-6
6.4	B2LC Label configuration	6-7
6.5	B2SC Ticket layout configuration	6-12
6.6	B2PS PLU settings	6-15
6.7	FUCO FUCN local key arrangement	6-17
6.8	B2FC currency settings	6-19
6.9	B2NP Non-PLU settings	6-21
6.10	B2O1 Further settings 1	6-23
6.11	KFTF Keyboard layout	6-25

7 COMMISSIONING 7-1

7.1	B2SU Commissioning BCII	7-1
-----	-------------------------------	-----

8 MESSAGES 8-1

8.1	PROG Program version	8-2
8.2	TIME date/time	8-4
8.3	WARE Scales status	8-5
8.4	B2CR clear RAM reboot scale	8-5

1 GENERAL

1.1 Warranty

Utilization of software

The manufacturer and its subsidiary companies are not liable for the consequences of using this data description or that of the software discussed therein.

If not expressly specified, there is nothing in this manual that represents a license for intellectual property with regard to the operator.

The manufacturer and its subsidiary companies are not liable for problems that arise from using this product. Please note that changes to technical data of the product are subject to change without advance notice.

This data description was prepared with great care. If errors or omissions are still to be found, we would be very grateful if you could inform us.

1.2 Supplementary documentation

The focus of the content of this documentation is on the description of data structures and their content.

To understand individual tables, knowledge about the corresponding scales functions and operating processes is unavoidable. These connections are described in the manuals on

"BCII retail scales programming instructions"

Current supplements are documented in the the release announcements of new versions of the BCII program.

1.3 General modifications

Date	Description
27.04.07/moll Version 1.00	Creation date
10.08.07/moll	WinCWs adjustments
12.11.07/Moll Version 1.00 Revision 2	Switches additional Texts (label layout, receipt layout)
26.03.08/Moll Version 1.00 Revision 3	- new attributes - description KFTF - new table FUCN
14.05.08/Moll Version 2 Revision 1	- 6 digits PLU number - Database 2.0 scalable with commissioning
01.08.08/Moll	- Code structure and 180 ° rotated label
12.09.08/Moll	Database version number minimal barcode height
16.02.09/FR/ Moll	Database version number, Text use by sell by an packed

1.4 Changes in field names

None at the current time

Table name	old field name	new field name	new field name
------------	----------------	----------------	----------------

1.5 Program transparency

Program version BCII from 1.00

- Database version 1.00
- SxCom protocol version 8.00

Program version BCII from 1.10

- Database version 1.01
- SxCom protocol version 8.00

Program version BCII from 1.20

- Database version 2.00
- SxCom protocol version 8.00

Program version BCII from 1.30

- Database version 9.12
- SxCom protocol version 8.00
-

Program version BCII from 1.40

- Database version 9.13
- SxCom protocol version 8.00

1.6 Changes in the BCII database

BCII Program	Database version	Tables
Changes		
1.00 moll	1.00	Creation version
1.00 Moll 61117100100	1.01	B2LC Field FRAM B2SC Field PRGT
1.10 Moll 61117100110	1.01	B2MS UDSI FZWE FZNW eliminated B2AG RPRG (new) B2PL AISA eliminated MULA (new) FUCN (new) table B2O1 SSEN (new) KSEL (new) B2LG HIGH (max. value) B2LS HIGH (max. value)
1.20 61117100120	2.0	B2SU new PLST PNUM (max value) KLGE (max value) B2PS NDRK (max value) PMI6 (new) PMA6 (new.) Message B2CR
1.30 61117100130	9.12	COST KEZI (eliminated) COWE (eliminated) COST (eliminated) PRZI (eliminated) COOP (eliminated) CVAK (new) COVO (new) B2LG P180 (new) B2LC P180 (new) B2G1 GPCN (new) B2LC EHIG (min. value) B2LG EHIG (min. value) B2SC EHIG (min. value)

1.40	9.13	B2LG B2LC ATPO (new) ATUB (new)
------	------	--

2 Programs

2.1 Communication programs

SWCom is for transmitting data and messages between a computer or several decentralized subsystems (SC/SW-scales, BCII or GX-price labeler).

There are two program versions available

Process control via command line (SWCMAN)

Transmission is started from the computer by accessing the SWCMAN program. The information required for process control is transferred as parameters to the command line when the program is started. Only one file per command can be transmitted.

This version is recommended for realizing customer-specific solutions.

Process control via a batch interpreter (SWC_BAT)

This version is especially used in the CWS and SWDasi application programs and is only available for the DOS and WIN9X/NT operating systems.

Transmission parameters, such as communication definitions, addressed subsystem type, repeat parameters, modem type and transmission commands are transferred via a "USER file" to the program.

Even in the definition or modification of the "USER" command, a check is carried out via a rule file to see if the specified command can be executed on the subsystem. If the corresponding definition is available in the rule file, it is possible to use a command in the application file to start a command sequence in the rule file. This command sequence is executed when the transmission program is started in single commands, depending on error conditions. The repeat parameters can be set to determine how often individual data records are sent or received before a timeout error is produced, or how often the connection should be reestablished after a timeout error or after connection interruption (in modem connections). When the connection has been reestablished, the last processed command is reprocessed or an attempt is made to continue the transmission of the last transmitted data record.

In one program call, several transmission commands can be executed and several fields transferred.

2.2 Help program

SWU: SW file editor (DOS and UNIX operating systems)

The SSU program (**S**ystem **S**cales **U**tility) is an editor for specially-formatted SC/SW data files.

With the SWU, SC/SW data files can be viewed, edited and supplemented.

Syntax: SWU <SC/SW/BCII data file>

Keyboard input:

Arrow up	Moves the section of the window up
Arrow down	Moves the section of the window down
Arrow up	Moves the cursor towards the start of the file
Arrow down	Moves the cursor towards the end of the file
Return	Transfers to edit mode for the current record. If the cursor is on EOF (end of file), a new record is attached unless editing is cancelled by pressing ESC.

Ing.del. Ends the program

ASCII conversion to SWASCII

The conversion program, SWASCII offers a freely programmable conversion from SW/SC/BCII-files to user-defined text files and back.

The description of the text file provided by the user is included in a so-called info file with an "if" file extension.

3 File description

3.1 HDLC- File structure

The useful data is stored in sets in a HEX file. Alongside the actual useful data, this file also contains additional information which is of absolute necessity for conversion into other formats and for further processing by the application. When a file transfer begins, the scales transmit the field identifiers belonging to the BCII/SC/SW database table. Based on this information, SWCom finds out the maximum length of a table. The actual length of a database entry is determined by the length of a variable character string in the last field, if available. The actual length is inserted into the file before each record. Deselected numeric fields within a table are filled up to their maximum length with 0 hex; deselected alphanumeric fields are filled with empty spaces. In addition, conversion from INTEGER data types to MOTOROLA into the INTEL data format take place (BYTE reversal). The field identifiers included in the confirmation telegrams of the scales can only be used to calculate the maximum useful data length and for INTEL-MOTOROLA conversion of HEX data.

Type definition **TYdbmAtt** in the field identifier
(BIZERBA format, from SC-SW version 2.50):

Length	Data type	Description	Comment
4	char[4]	att_symbol	Attribute identifier icon
2	Short	ges_stellen	Number of digits
8	TYint64	minWert	smallest value
8	TYint64	maxWert	largest value
1	Uchar	typ	Attribute type
1	Uchar	key	Key attribute? yes/no
1	Uchar	option	Attribute selected? yes/no
1	Uchar	ptr_on_val	Internal value for SW scales

Explanations:

att_symbol: The 4-digit field name uniquely identifies a specific attribute in the conversion of data from SC/SW files into the SC/SW database. Only use capital letters in the field names.

ges_stellen: In describing a table, it is possible to select wither the
minWert: number of digits or with the minimum and maximum value
maxWert:

ges_digits > 0:

Specifies the maximum value of a whole number.
The storage size of a field is determined from this value
:

```
0 < ges_stellen <= 4 : SHORT      (2 Byte)
5 <= ges_stellen <= 9 : LONG       (4 Byte)
ges_stellen > 9      : INT64      (8 Byte):
                        typedef struct
                        {
                            signed long high;
                            unsigned long low;
                        } TYint64;
```

The components, 'minValue' and 'maxValue', are not taken into consideration.

ges_stellen <= 0:

To describe the attribute values, the fields 'minValue' and 'maxValue' are used. They are only valid for selected attributes (**option = 1**), deselected attributes (**option = 0**) are assigned with 0 Hex. The following table specifies how the components 'minValue' and 'maxValue' are interpreted in relation to attribute type:

Attribute type	minValue	maxValue
STRING	---	maximum string length
VARSTR	---	maximum string length
DATUM	---	---
GEWICHT	smallest value	largest value
GELD	smallest value	largest value
INTEGER	smallest value	largest value
INTBFDB	minimum number of bits	maximum number of bits
INTBFHX	minimum number of bits	maximum number of bits
INTBFEX	minimum number of bits	maximum number of bits
VARDATA	---	maximum data length
STRUCT	---	maximum data length

Attributes with INTBFDB and INTBFHX bit field types are configured with the number of bits they are supposed to contain, while attributes of type INTBFEX are configured with the largest and smallest converted numeric value.

typ:	1: STRING	(character string)
	2: VARSTR	(character string)
	3: DATUM	(long)
	4: GEWICHT	(Not used currently)
	5: GELD	(short, long, int64: depending on maximum)
	6: INTEGER	(short, long, int64: depending on maximum)
	7: INTBFDB	Integer bit field in the SC/SW database (only for internal use)
	8: INTBFEX	Integer bit field for external applications (expanded bit field)
	9: VARDATA	Memory excerpt from the BCII/SC/SW database (only used for data backup) In comparison to VARSTR '\0' bytes can be included in the data. There is no fixed structure. The data is represented in the BCII/SC/SW data format. The size of a VARDAA value is fixed in the first two bytes of the value as a short integer (MOTOROLA format). These two bytes are not counted with respect to size.
	10: INTBFHX	Integer bit field for external applications (non-expanded bit field)
	11: STRUCT	User-specific data format: len typ data : len: [short] Length of useful data in bytes incl. type and excluding type: [short] Structure type of useful data: 0: char array 1: short array 2: long array 3: INT64 array > 3: special user structures data: useful data

key: Specifies whether or not an attribute is a key attribute.
0 = no key attribute
1 = key attribute

option: specifies whether an attribute is selected (configured). Data from a deselected attribute is filled with a 0 hex for the remaining complete length. This ensures that during transmission and in the file, useful data is always processed at its maximum length. The fields, minValue and maxValue are no longer significant in deselected attributes.

0 = Attribute is not selected

1 = Attribute is selected

All further preparation of the files for sending and receipt is accomplished by application programs, such as BIZERBA, CWS, BLD, BSD.

3.1.1 File frame

Description of fields	Data type
Length of header data (in bytes)	ushort
BCII/SC/SW-table name	char[14] ASCII
Target system version	unsigned short
Revision of target system	unsigned short
Date/time target system from 1.1.70	long
Target system program number	char[12] ASCII
Customer name	char[80] ASCII
Special program name	char[40] ASCII
Version SWCom	unsigned short
Revision SWCom	unsigned short
Date/time SWCom from 1.1.70	long
SWCom program number	char[12] ASCII
Parameter field	uchar[80] ASCII
Reserve	char[80] ASCII
Number of target system	unsigned short
Length of field identifier	unsigned short
Field identifier	TYdbmAtt[100] **
Data block length	unsigned short
Number of data blocks	unsigned long

Comment: TYdbmAtt (length is 26 bytes)

3.1.2 Useful data

Useful data (n data records):

Record length 1	ushort
Record 1	
Record length N	ushort
Record N	

Explanations:

File frame:

Data block length: maximum length of a structure

Number of data blocks: Number of records in BCII/SC/SW file:

- > 0: The user updates the "Number of data blocks".
- = 0: SWCom searches the SW-file to determine the "Number of data blocks".

Useful data:

Record length: effective length of a structure (incl. record length).

The length of the header data is:

2946 bytes

3.2 HDLC record structure

The HDLC set structure is **not** supported.
BCII is basically compatible with SC-TCP.

3.3 ScLSV1, ScTCP, and ScXML record structure

Table : PLST	global table	Number of attributes: 39
--------------	--------------	--------------------------

Description: PLU master data

```

|----- Table name
|----- Number of fields
|----- Local: scales-specific
|----- global: valid throughout the system

```

|----- Field description

Description							
Desc.	KEY	OPT	MinValue	MaxValue	Offs	Laen	Data type
<i>delete flag</i>							
WALO	KEY	FIX	0	1	0	2	INTEGER

Desc.	KEY	OPT	MinValue	MaxValue	Offs	Laen	Data type
-------	-----	-----	----------	----------	------	------	-----------

```

|----- Attribute type:
|----- all data types
|----- are represented in
|----- ASCII format.
|----- INTBFDB Fields
|----- are omitted
|----- Field length
|----- (in data types
|----- STRING and VARSTR
|----- length
|----- is reduced by 1 byte,
|----- numerical fields do not
|----- have a leading zero)
|----- Offset (no meaning here)
|----- Maximum value
|----- Minimum value
|----- Attribute can be
|----- configured
|----- OPT: configurable
|----- FIX: available and fixed
|----- EXP: expanded
|----- bit field (=FIX)
|----- Key attribute
|----- (all KEY attributes
|----- have to be available in
|----- the data record,
|----- the order is arbitrary)
|----- Attribute icon
|----- (= field name in capital
|----- letters)

```

A data record constitutes three components:

1

11

11

| Command
see "SV

3.3.1 Example of a dialog

Host sends VAT master data to the scales:

Host request:

Table name:

Name	Information separator
char[6]	char
MWST	ESC

Control part:

Command	Target address	Information separator
char	char[2]	char
S	00	ESC

Useful data:

1. Field

Field name	Value	Information separator
char[4]	char	char
WALO	0	ESC

2. Field

Field name	Value	Information separator
char[4]	char	char
MWNU	1	ESC

3. Field

Field name	Value	Information separator
char[4]	char[4]	char
MWST	1500	ESC

4. Field

Field name	Value	Information separator
char[4]	char	char
MWWI	0	ESC

5. Field

Field name	Value	Information separator
Char[4]	char[20]	char
MWTE	MEHRWERTSTEUER_1	ESC

SW confirmation:**Table name:**

Name	Information separator
Char[6]	char
MWST	ESC

Control part:

Command	Target address	Information separator
Char	char[2]	char
s	00	ESC

Useful data:

Field name	Value	Information separator
Char[4]	char	char
QUIT	0	ESC

ScLSV1:

The ScLSV1 protocol and serial transmission is **not** supported.

ScTCP:

In ScTCP an end identifier is attached to the end of each data record ("BLK <ESC>" a block counter is also possible as field content).

```
„MWST <ESC>S00<ESC>WALO0<ESC>MWNU1<ESC>MWST1500<ESC>MWWI0<ESC>M
WTEMEHRWERTSTEUER_1<ESC>BLK <ESC>“
„MWST <ESC>s00<ESC>QUIT0<ESC> BLK 0<ESC>“
```

ScXML:

Within the data records (<sx_r>) , the data representation can be the same as in ScLSV1. If the table name and the control part in the header (<sx_header>) of the ScXML file is available, the respective data records only contain the useful data of the data record.

Example 1:

```
...
  <sx_data>
    <sx_r>MWST <ESC>S00<ESC>WAL00<ESC>MWN01<ESC>MWST1500<ESC>MWWI
0<ESC>MWTEMEHRWERTSTEUER_1<ESC></sx_r>
  </sx_data>
```

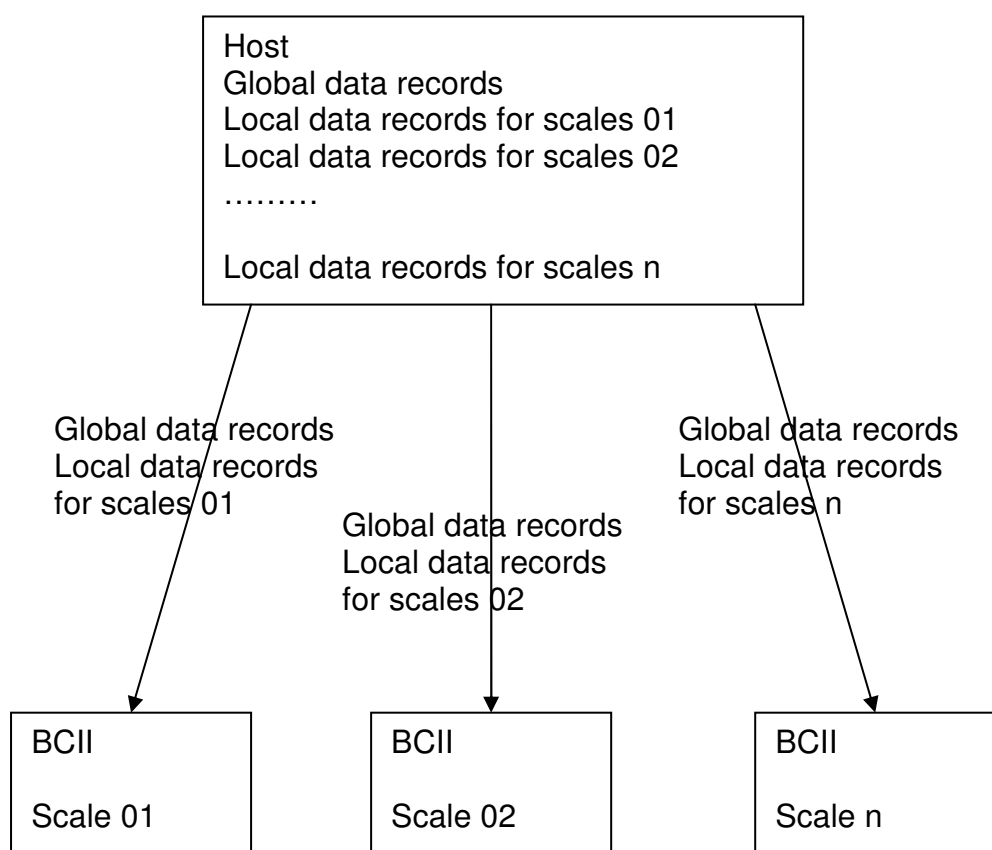
Example 2:

```
...
  <sx_header>
    <sx_table>MWST</sx_table>
    <sx_cmd>MWST</sx_cmd>
    <sx_id>MWST</sx_id>
  </sx_header>
  <sx_data>
    <sx_r>WAL00<ESC>MWN01<ESC>MWST1500<ESC>MWWI0<ESC>MWTEMEHRWERT
STEUER_1<ESC></sx_r>
  </sx_data>
```

The ScXML file format can be used within the host interface.

3.4 Topology

The BCII is **not** a master scale. Data is **not** distributed by a BCII scales!
If the data should be distributed across several BCII, this distribution must be executed by setting up successive connections from the host interface to the scales of the host interface itself.



4 Data record description

The descriptions "global table" and "local table" in this context relate to the addressing of the scales when data is transmitted and is not related to how memory is organized in the scales.

Tables beginning with B2 are only found in BCII. The other tables form subsets of the SC tables.

Address: 0 = global table
 1 - 99 = Local table

4.1 Table overview in BCII

ATST	global table	General text master data
COST	global table	Barcode structure
B2G1	global table	global settings 1
B2LG	global table	Label configurations
ETST	global table	Labels (BLD)
FOST	global table	Forms (BLD)
KFST	global table	Header/footer text master data
LOST	global table	Logos (BLD)
PLST	global table	PLU master data
FUGL	global table	Operator Key assignment
B2AG	Local table	General operating settings
B2MS	Local table	Settings for multiple operators
B2PL	Local table	Settings for price labeling
B2LC	Local table	Label layout configuration
B2SC	Local table	Ticket layout configuration
B2PS	Local table	PLU settings
FUCO FUCN	Local table	Direct PLU key assignment Direct tare Key assignment
B2FC	Local table	Foreign currency settings
B2NP	Local table	Non-PLU settings
B2O1	Local table	Further settings
KFTF	Local table	Keyboard layout
PROG	Message	Program version
„UHR „	Message	Set time
WARE	Message	Scales reference

5 Master data

5.1 ATST gen. Texts master data

Table : ATST	global table	Number of attributes: 3
--------------	--------------	-------------------------

Description : General texts master data

<i>Description</i>					
Desc.	KEY	OPT	MinValue	MaxValue	Data type
<i>delete flag</i>					
WALO	KEY	FIX	0	1	INTEGER
<i>General text number</i>					
ATNU	KEY	FIX	1	999999999	INTEGER
<i>General text</i>					
ATTE	-	FIX	0	1501	VARSTR

Explanations:

General texts are texts listing ingredients or recipes to supplement a product. Similarly, additional texts for sell-by date or rounding etc.

WALO: **delete flag :**
 0 = Data record is being modified or recreated
 1 = Data record is deleted

ATNU: **General text number**

ATTE **General text.**
 The text has to be UTF8 coded!!
 Within the text, the font number is supplied via ^; Change font. New line via '\n'.

5.2 B2G1 global settings

Table : B2G1	global table	Number of attributes: 5
--------------	--------------	-------------------------

Description : global settings

<i>Description</i>					
Desc.	KEY	OPT	MinValue	MaxValue	Data type
<i>delete flag</i>					
WALO	KEY	FIX	0	1	INTEGER
<i>Rounding methods for customer total</i>					
RTMO	-	OPT	0	9	INTEGER
<i>Rounding of product total</i>					
RAMO	-	OPT	0	9	INTEGER
<i>Second currency (€) factor</i>					
SCFM	-	OPT	0	3	INTEGER
<i>Second currency € rate</i>					
SCCA	-	OPT	0	999999	INTEGER
<i>Global code structure number</i>					
GBCN	-	OPT	0	99	INTEGER

WALO: **delete flag :**
0 = Data record is being modified or recreated
1 = Data record is initialised with standard values

RTMO: **Rounding methods for customer total Rounding methods for customer total**

- 0** no rounding
- 1** Rounding to 5
- 2** Rounding to 10
- 3** Rounding to 50
- 4** Rounding to 100
- 5** mathematical rounding to 5
- 6** mathematical rounding to 10
- 7** mathematical rounding to 25
- 8** mathematical rounding to 50
- 9** mathematical rounding to 100

RAMO: Rounding of product total Rounding of product total
0 no rounding
1 Rounding to 5
2 Rounding to 10
3 Rounding to 50
4 Rounding to 100
5 mathematical rounding to 5
6 mathematical rounding to 10
7 mathematical rounding to 25
8 mathematical rounding to 50
9 mathematical rounding to 100

The second currency is only active for non-€ scales. The functions are for countries that are still in the € transition period.

SCFM: Second currency (€) factor
0 .. 3 = factor 1..1000
Example
ZW = second currency (€)
NW = National currency
SCCA input = 1955.83
SCFM = 3
=> 1000 ZW (€) = 1955.83 NW

SCCA: Second currency (€) calculation rate
e.g. 195583
results in 1955.83 in the national currency

GBCN: Globale code structure number
00 for PLUS' take code struct number from ECO1 field
01..99 take for all PLUS the same number 1 to 99

5.3 B2LG label configuration for BCII

Table : B2LG	global table	Number of attributes: 20
--------------	--------------	--------------------------

Description : Label configurations for BCII

Description					
Desc.	KEY	OPT	MinValue	MaxValue	Data type
<i>delete flag</i>					
WALO	KEY	FIX	0	1	INTEGER
<i>Configuration number</i>					
STNR	KEY	FIX	1	3	INTEGER
<i>Predefined label layout</i>					
ALID	-	OPT	0	99	INTEGER
<i>BLD label layout</i>					
BLDI	-	OPT	0	99	INTEGER
<i>Label height</i>					
HIGH	-	OPT	28	148	INTEGER
<i>Label distance</i>					
PGAP	-	OPT	8	40	INTEGER
<i>Text sell-by date</i>					
PSBD	-	OPT	0	2	INTBFHX
<i>EAN – Code</i>					
EANC	-	OPT	0	2	INTBFHX
<i>EAN code height</i>					
EHIG	-	OPT	5	22	INTEGER
<i>Tare printout</i>					
TARE	-	OPT	0	1	INTBFHX
<i>Header text printout</i>					
HEAD	-	OPT	0	99	INTEGER
<i>Additional text for sell-by date</i>					
SBDT	-	OPT	0	999999999	INTEGER
<i>Header logo</i>					
LOGO	-	OPT	0	9999	INTEGER
<i>Start of text field</i>					
TFBE	-	OPT	0	84	INTEGER
<i>Start of data field</i>					
DFBE	-	OPT	0	84	INTEGER
<i>Data field monitoring on/off</i>					
DFIC	-	OPT	0	1	INTBFHX
<i>Text format for texts</i>					
TLAY	-	OPT	0	110	INTEGER
<i>Price range and texts</i>					
FRAM	-	OPT	0	7	INTBFHX

<i>Operator, device- or digi code</i>					
SDDI	-	OPT	0	3	INTEGER
<i>Print foreign currency</i>					
FCSE	-	OPT	0	4	INTEGER
<i>label 180° rotated</i>					
P180	-	OPT	0	1	INTBFHX
<i>Text packed on</i>					
ATPO	-	OPT	0	999999999	INTEGER
<i>Text use by</i>					
ATUB	-	OPT	0	999999999	INTEGER

Explanations:**WALO: delete flag :**

0 = Data record is being modified or recreated

1 = Data record is deleted

STNR: Label configuration number:

3 label configurations can be stored under the label configuration number and imported.

Compare label configuration administration in the programming instructions.

ALID: Label type

0 Label from BLD. See ALID

1.. 99 predefined label.

There are pre-defined label layouts in the device. If you are working with BLD (Bizerba Label Designer) you can select freely-definable. Pre-defined labels have a text and a data area. The text area contains text, e.g. the product text. The data area contains data on the product, weight, sales price sum EAN code, etc. Text and data field areas can be adjusted. No difference is made between text and data field range on BLD labels.

BLDI: BLD label layout :

1..99 Select BLD label layout (ALID has to be 0)

HIGH: Label height :

28..148 mm 54 and 55 mm are not permitted

PGAP: Label distance :

8 .. 40 Dots (8 Dots = 1mm)

PSBD: Packaged-on and sell-by date:

Bit field

Bit no.:

1 0

|

|

1 - Sell-by date is printed

|

1 packaged date is printed

EANC: EAN code print-out on label:

Bit field

Bit no.:

1 0

|

|

1 - Total EAN code is printed

|

1 product EAN code is printed

EHIG: EAN code height:

5..22 mm

TARE: Tare printout:

Bit field

Bit no.:

0

|

1 -Tare is printed

HEAD: Header text printout :

0 without header text

1..99 Number of the header text to be printed

(Text from KFST)

SBDT: Text for sellby :

0 standard text

1..999999999 Number of text to be printed

LOGO: Logo print-out in header row :

0 without header logo

1..9999 Number of the header logo to be printed

TFBE: Start of text field :

0..84

Predefined labels are separated into a text and a data field area. The text field contains the product text. The text field can be adjusted up or down in the label:

0: The text field begins right after the label header.

1..84: The text field is adjusted downwards in mm.

(0 position starts after the label header)

In BLD labels there is no subdivision of this type.

DFBE: Start of data field :

0..84

Predefined labels are separated into a text and a data field area. The data field contains product data, including unit price, weight, selling price, total, EAN code, etc. The data field can be adjusted up or down in the label.

0: The data field is moved to the end of the label.

1..84: The data field is always adjusted downwards in mm.

(0 position starts after the label header)

In BLD labels there is no subdivision of this type.

DFIC: Data field monitoring on/off:**Bit field**

Bit no.:

0

|

1 - Data field monitoring is on

When data field monitoring is activated, this prevents texts from the text field being written to the data field.

Text is cut-off on the label.

In the case of labels on tickets, the print-out is simply extended.

If monitoring is deactivated, texts from the text field might continue into the data field.

TLAY: Text format for product texts:

0 The texts are represented according to the formatting in the PLU text or in line with the text sizes provided in the layout.

1..110 Select a defined font for text.

FRAM: Price range and texts

Bit field

Bit no.:

5

4

3

2

1

0

|

|

|

|

|

|

|

|

|

|

|

|

|

|

|

|

|

|

|

|

|

|

|

|

|

|

|

|

|

|

|

|

|

|

|

|

|

|

|

|

|

|

|

|

|

|

|

|

|

|

|

|

|

|

|

|

|

|

|

|

|

|

|

|

|

|

|

|

|

|

|

|

|

|

|

|

|

|

|

|

|

|

|

|

|

|

|

|

|

|

|

|

|

|

|

|

|

|

|

|

|

|

|

|

|

|

|

|

|

|

|

|

|

|

|

|

|

|

|

|

|

|

|

|

|

|

|

|

|

|

|

|

|

|

|

|

|

|

|

|

|

|

|

|

6

|

1-Print. Additional text
from ATST

SDDI: Operator no., machine parameter check digit on label

0 without output
1 with operator no.
2 with key machine figure
3 with digit code

The label contains a field intended for output. The selected number is inserted into the field.

FCSE: Foreign currency

Bit no.:

3	2	1	0	
			1-	Only print total
		1-----		Print selling price and total
	1-----			Print unit price, selling price and total
1-----				Print conversion rate

P180: rotate label 180°

0: no rotation
1: rotation

ATPO: Text packed on

0 Standard Text
1..999999999 number of text to be printed

ATUB: Text use by

0 Standard text
1..999999999 number of text to be printed

5.4 COST Barcode data

Table : COST	global table	Number of attributes: 7
--------------	--------------	-------------------------

Description : Barcode data

Beschreibung					
Bez.	KEY	OPT	MinWert	MaxWert	Datentyp
<i>Delete flag</i>					
WALO	KEY	FIX	0	1	INTEGER
<i>Codestructc number</i>					
COSN	KEY	FIX	1	99	INTEGER
<i>Code build structure active</i>					
CVAK	-	FIX	0	1	INTBFEX
<i>Code build structure</i>					
COVO	-	FIX	0	501	VARSTR

Explanations:

- WALO:** **delete flag :**
 0 = Data record is being modified or recreated
 1 = Data record is deleted
- COSN:** **code structure-number**
 01 .. 99
- CVAK:** **Code-build structure active:**
 0 = not active
 1 = active
 BCII only allows active!
 The old creation of code structures isn't supported anymore
- COVO:** **Code build structure**

5.5 ETST Labels

Table : ETST	global table	Number of attributes: 11
--------------	--------------	--------------------------

Description : Labels

Description					
Desc.	KEY	OPT	MinValue	MaxValue	Data type
<i>delete flag</i>					
WALO	KEY	FIX	0	1	INTEGER
<i>Label number</i>					
ETNU	KEY	FIX	1	99	INTEGER
<i>Form number</i>					
FONU	-	FIX	1	99	INTEGER
<i>Left coordinate</i>					
LEFT	-	FIX	0	16	INTBFHX
<i>Upper coordinate</i>					
TOP	-	FIX	0	16	INTBFHX
<i>Right coordinate</i>					
RGHT	-	FIX	0	16	INTBFHX
<i>Lower coordinate</i>					
BOTT	-	FIX	0	16	INTBFHX
<i>Distance</i>					
ETAB	-	FIX	7	75	INTEGER
<i>Label width</i>					
ETBR	-	FIX	0	16	INTBFHX
<i>Decorative labels-offset</i>					
ETVS	-	FIX	0	16	INTBFHX
<i>Label text</i>					
ETTE	-	FIX	0	21	STRING

Explanations:

WALO: **delete flag :**
 0 = Data record is being modified or recreated
 1 = Data record is deleted

This table is only for internal use (data backup of BLD and BIZERBA store management)!

ETST is not implemented in BCII Version 1.00

5.6 FOST Forms

Table : FOST	global table	Number of attributes: 12
--------------	--------------	--------------------------

Description : Forms

Description					
Desc.	KEY	OPT	MinValue	MaxValue	Data type
<i>delete flag</i>					
WALO	KEY	FIX	0	1	INTEGER
<i>Form number</i>					
FONU	KEY	FIX	1	99	INTEGER
<i>Field number</i>					
FENU	KEY	FIX	1	99	INTEGER
<i>Content & type</i>					
INTY	-	FIX	0	32	INTBFHX
<i>Left coordinate</i>					
LEFT	-	FIX	0	16	INTBFHX
<i>Upper coordinate</i>					
TOP	-	FIX	0	16	INTBFHX
<i>Right coordinate</i>					
RGHT	-	FIX	0	16	INTBFHX
<i>Lower coordinate</i>					
BOTT	-	FIX	0	16	INTBFHX
<i>Size</i>					
GRSS	-	FIX	-1	100	INTEGER
<i>Font</i>					
FONT	-	FIX	0	16	INTBFHX
<i>Field attribute</i>					
FEAT	-	FIX	0	32	INTBFHX
<i>Fixed text</i>					
FIXT	-	FIX	0	501	VARSTR

Explanations:

WALO: delete flag :
0 = Data record is being modified or recreated
1 = Data record is deleted

This table is only for internal use (BLD and BIZERBA store management)!
FOST is not implemented in BCII Version 1.00

5.7 KFST Header/footer text master data

Table : KFST	global table	Number of attributes: 3
--------------	--------------	-------------------------

Description : Header/footer text master data
--

<i>Description</i>					
Desc.	KEY	OPT	MinValue	MaxValue	Data type
<i>delete flag</i>					
WALO	KEY	FIX	0	1	INTEGER
<i>Header/footer text number</i>					
KFNU	KEY	FIX	1	99	INTEGER
<i>Text for header/footer row</i>					
KFTE	-	FIX	0	501	VARSTR

Explanations:

WALO: **delete flag :**
 0 = Data record is being modified or recreated
 1 = Data record is deleted

KFNU: **Header/footer row number**

KFTE: **Header/footer text.**
 The text has to be UTF8 coded!!
 New line via '\n'.

List of supported formatting sequences:

^ number ; font size,
 Value range: ^1; ... ^110;
Non-defined formatting sequences are output as text.

Macros: -

5.8 LOST Logos

Table : LOST	global table	Number of attributes: 4
--------------	--------------	-------------------------

Description : Logos

Description					
Desc.	KEY	OPT	MinValue	MaxValue	Data type
<i>delete flag</i>					
WALO	KEY	FIX	0	1	INTEGER
<i>Logo number</i>					
LOGN	KEY	FIX	1	9999	INTEGER
<i>Logo record number</i>					
LOSA	KEY	FIX	0	9999	INTEGER
<i>Logo data</i>					
LODT	-	FIX	0	504	VARDAT

Explanations:

WALO: **delete flag :**
 0 = Data record is being modified or recreated
 1 = Data record is deleted

This table is only for internal use (BLD and BIZERBA store management)!
LOST is not implemented in BCII Version 1.00

5.9 PLST PLU master data

Table : PLST	global table	Number of attributes: 16
--------------	--------------	--------------------------

Description : PLU master data

Description					
Desc.	KEY	OPT	MinValue	MaxValue	Data type
<i>delete flag</i>					
WALO	KEY	FIX	0	1	INTEGER
<i>PLU number</i>					
PNUM	KEY	FIX	1	999999	INTEGER
<i>Unit price 1</i>					
GPR1	-	FIX	0	999999	GELD
<i>Product group</i>					
WGNU	-	FIX	1	9999	INTEGER
<i>Tare value weight</i>					
TARE	-	OPT	0	9999	INTEGER
<i>Shelf-life 1</i>					
HBA1	-	OPT	0	999	INTEGER
<i>EAN code</i>					
ECO1	-	OPT	0	14	VARSTR
<i>General text number</i>					
ALT1	-	OPT	0	999999999	INTEGER
<i>General text number</i>					
ALT2	-	OPT	0	999999999	INTEGER
<i>General text number</i>					
ALT3	-	OPT	0	999999999	INTEGER
<i>General text number</i>					
ALT4	-	OPT	0	999999999	INTEGER

<i>Price overwrite</i>					
PREU	-	EXP	0	1	INTBFEX
<i>PLU class</i>					
KLAR	-	EXP	0	14	INTBFEX
<i>Weight class</i>					
KLGE	-	EXP	0	6	INTBFEX
<i>Data of origin</i>					
HKDR	-	EXP	0	2	INTBFEX
<i>Text field numbers</i>					
TFZU	-	OPT	0	44	STRUCT
<i>Logo numbers</i>					
LFZU	-	OPT	0	10	STRUCT
<i>Product text</i>					
PLTE	-	FIX	0	501	VARSTR

Explanations:

WALO: **delete flag :**
 0 = Data record is being modified or recreated
 1 = Data record is deleted

PNUM: **PLU number**
 1..999999

GPR1: **Unit price 1**
 0.999999

WGNU: **Product group number**
 1..9999

TARA: **Tare value**
 0..9999

Datum 1 Packaged date is current date
 Datum 2 Sell-by date is formed from the packaged date + shelf-life days 1
 Print after "sell by" text.

HBA1: **Shelf life days 1**
 0..999

ECO1: EAN code [0...13]:
eco1[0] describes the code type:
0 = Instore code
1 = manufacture code WAN 13

Typ 0 - Instore code

eco1[1..2]: Code structure (see table COST)
eco1[3..6]: Code content
if the code content == XXXX is available:
eco1[3..6]:Value is XXXX
otherwise:
eco1[3..12]:no meaning
eco1[7..12]: are only filled with zeros.
eco1[13]: Zero (string-end character)

Example: ECO1 = 00112340000000
means:

Instore code with code structure 01 and code content 1234.
The code structure can include:

Code digit 24, code content XXXX (freely assignable), with check digit (C),
price coding (PPPPP).

In this case the following code is produced:

241234CPPPPPC

Type 1 - manufacturer code EAN 13:

eco1[1..12]: Code content without check digit.
All 12 digits must be entered, including any possible
zeros.

Example: ECO1 = 1901234567890

produces the code:

901234567890C

ALT1: General text:
The general text is a text in addition to the product text, e.g. for recipes,
ingredients, etc. This additional text can be assigned to different products
by entering the text no.

. = 0: without additional text
.
> 0: Reference to ATST table

ALT4:

PREU: Price overwrite permitted
0=no, 1 = yes

KLAR:	Product type: G/H/M		
	G	Weighed	=0
	H	Manual	=1
	M	Minus	= 2
	H	Manual	=3
	G	Weighed	=4
	H	Manual	=5
	H	Manual	=6
	G	Weighed	=7
	G	Weighed	=8
	G	Weighed	=9
	G	Weighed	=10
	H	Manual	=11
	H	Manual	=12
	H	Manual	=13
	H	Manual	=14

KLGE:	Weight class
	0 = 1 kg
	1 = 100 g
	2 = lib
	3 = ¼ lib
	4 = ½ lib not supported (=> error message)
	5 = x lb x in STPA not supported (=> error message)
	6 = 500g

Attention:Supported weight classes depend on country versions.
The scale supports one or two scale classes:
Examples: 1 kg and 100g
or kg and 500g, or only kg !

HKDR:	Data of origin
	0 = none
	1 = with data of origin
	2 = with data of origin

TFZU: Text field allocation:

Allocation of text numbers to the related text fields.

If freely-programmable labels are an option, a maximum of 10 general text numbers can be assigned to the "data" structure component in TYvarDataTFZU.

**Important: gaps between the numbers may not occur
, i.e. fields must be consecutively filled with
data.**

If no freely programmable labels are possible, the TYvarDataTFZU structure must be preassigned with 0.

```
typedef struct
{
    short    len;
    short    varTyp; /* 4 = VARTFZU */
    long     daten [10];
} TYvarDataTFZU;
```

LFZU: Logo field assignment

Allocation of logo numbers to the related logo fields.

If freely-programmable labels are an option, maximum of 3 logo numbers can be assigned to the "data" structure component in TYvarDataLFZU.

**Important: gaps between the numbers may not occur
, i.e. fields must be consecutively filled with
data.**

If no freely programmable labels are possible, the TYvarDataLFZU structure must be preassigned with 0.

```
typedef struct
{
    short    len;
    short    varTyp; /* 5 = VARLFZU */
    short     daten [3];
} TYvarDataLFZU;
```


PLTE: Product text

The product text can constitute several paragraphs and lines.

Line:

Character string that can be printed in a print row.

Paragraph:

Character string from one or multiple print rows, which are limited by the formatting character \n (= 0A Hexadecimal or LF).

A text field can contain a maximum of:

500 bytes of data memory,

texts are UTF8 coded, the memory requirement in the bytes of a character is a result of converting the unicode of the character into the UTF8 format. Unicode characters < 0x80 (ASCII Code) only need 1 byte of memory

List of supported formatting sequences:

^ number ; font size,
Value range: ^1; ... ^110;

Non-defined formatting sequences are output as text.

5.10 FUGL operator key arrangement

Table : FUGL	global table	Number of attributes: 4
--------------	--------------	-------------------------

Description : Global keyboard function assignment (only operator keys)
--

<i>Description</i>					
Desc.	KEY	OPT	MinValue	MaxValue	Data type
<i>delete flag</i>					
WALO	KEY	FIX	0	1	INTEGER
<i>Keys-function code</i>					
TSFC	KEY	FIX	-32768	32767	INTEGER
<i>Key data</i>					
TSDA	-	FIX	0	32	INTBFHX

Explanations:

This table is primarily used to assign numerical fixed keys that are used across all scales.

At the current time only the following applications are available:

Operator keys

WALO: delete flag :
0 = Data record is being modified or recreated
1 = Data record is deleted

TSFC: Keyboard codes:

Direct operator keys KEYB_DBT_x (x := operator number)

KEYB_DBT_1	0x8100	33024
KEYB_DBT_2	0x8101	33025
KEYB_DBT_3	0x8102	33026
.	.	.
.	.	.
KEYB_DBT_16	0x8102	33039

TSDA: Key value.
here: operator no.:

6 Configuration data

6.1 B2AG general operational procedure

Table : B2AG	Local table	Number of attributes: 14
--------------	-------------	--------------------------

Description : Settings general operational procedure settings

Description					
Desc.	KEY	OPT	MinValue	MaxValue	Data type
<i>delete flag</i>					
WALO	KEY	FIX	0	1	INTEGER
<i>Select paper type</i>					
AAPT	-	OPT	0	1	INTEGER
<i>Select main application</i>					
MAID	-	OPT	1	6	INTEGER
<i>Main application layout (ticket or label)</i>					
MULL	-	OPT	0	1	INTBFHX
<i>Main application - with/without addition of customer sum</i>					
MACT		OPT	0	1	INTBFHX
<i>Select alternative (toggle) application</i>					
TAID	-	OPT	1	6	INTEGER
<i>Alternative (toggle) application layout (ticket or label)</i>					
TULL	-	OPT	0	1	INTBFHX
<i>Alternative (toggle) application - with/without addition of the customer total.</i>					
TACT		OPT	0	1	INTBFHX
<i>Automatic reset</i>					
ANST		OPT	0	2	INTEGER
<i>Open tickets /labels after scale on</i>					
OSSM		OPT	0	2	INTEGER
<i>Beef labeling</i>					
BFTM		OPT	0	2	INTEGER
<i>Automatic abbreviated traceability number</i>					
BTAN		OPT	0	99	INTEGER
<i>PLU display time until input of abbreviated number</i>					
BTDT		OPT	1	99	INTEGER
<i>Negative totals are permitted</i>					
NTIA	-	OPT	0	1	INTBFHX
<i>Call mode quick service</i>					
RPRG	-	OPT	0	1	INTBFHX

Explanations:

WALO: delete flag :

- 0 = Data record is being modified or recreated
- 1 = Data record is initialised with standard values

AAPT: Select paper type

- 0 Receipt paper
 - 1 label paper
- It is not possible to select label paper in the ticket printer

MAID: Select main operating mode

- 1 multiple operator mode. Several operators can register at the scales at the same time.
- 2 self-service sales SB operation
- 3 Quick service sales
- 4 Price labeling with manual triggering
- 5 price labeling with automatic start
- 6 Multiple label printing

MULL: Main operating mode: Select layout

- 0 Ticket layout
- 1 Label layout

If label paper (see AAPT) is set, a TICKET layout is not possible. The ticket or label layout is not available for all operation modes.

MACT: Main operating mode: With addition of customer total

- 0 without addition of customer total
- 1 with addition of customer total

You can define two operating modes in the device: a main mode and an alternative (toggle) operating mode. You can switch between the operating modes via pressing the **Shift-ESC** keys. Therefore, it is possible to have multiple operators in the main operating mode, for example and during breaks in selling, change quickly to price labeling in order to handle prepackages.

TAID: Select alternative (toggle) operating mode

- 1 multiple operator mode. Several operators can register at the scales at the same time.
- 2 self-service sales SB operation
- 3 Quick service sales
- 4 Price labeling with manual triggering
- 5 price labeling with automatic start
- 6 Multiple label printing

- TULL:** **Alternative (toggle) operating mode: Select layout**
0 Ticket layout
1 Label layout
- If label paper (see AAPT) is set, a TICKET layout is not possible. The ticket or label layout is not available for all operation modes.
- TACT:** **Alternative (toggle) operating mode: With addition of customer total**
0 without addition of customer total
1 with addition of customer total
- ANST:** **Automatic reset**
0 The scales are reset if within the limit for automatic resetting,
1 Reset error 115 is displayed
2 Scale is not reset
- OSSM:** **Open tickets /labels after scale on/off**
0 Open tickets are printed out after scale off/on
1 Open tickets are not printed out and deleted
2 open tickets are not printed but are deleted
- BFTM:** **Beef labeling**
0 Identification is off
1 Input traceability no. manually
2 Select automatic traceability no.
When input is made manually, the operator has to enter a traceability number for traceability plus
.
In addition to the product text, a traceability text is output from the PLU 9900 + traceability number.
- BTAN:** **Automatic traceability no.**
00..99 rigidly defined traceability no.
In the case of PLUs for which traceability data is scheduled, a traceability text from PLU
PLU 9900 + traceability number is output in addition to the product text.
- BTDT:** **PLU display time until input of traceability no**
1..99 (1 short time, 99 long time)
Only active with manual input of traceability number.
- NTIA:** **Negative total**
0 Negative total is not permitted
1 negative total is permitted
- RPRG:** **Call mode quick service**
0 price call without registration
1 price call with registration

6.2 B2MS Multiple operator sales

Table : B2MS	Local table	Number of attributes: 12
--------------	-------------	--------------------------

Description : Settings for multiple operator sales
--

Description					
Desc.	KEY	OPT	MinValue	MaxValue	Data type
<i>delete flag</i>					
WALO	KEY	FIX	0	1	INTEGER
<i>Constant key function</i>					
COKM		OPT	0	2	INTEGER
<i>Print product and total label</i>					
PTLM		OPT	0	2	INTEGER
<i>Item operator display mode</i>					
SPOP		OPT	0	2	INTBFHX
<i>Till function</i>					
RGFN		OPT	0	4	INTEGER
<i>Payment in/out</i>					
CIOM		OPT	0	2	INTEGER
<i>Talon printout delay time</i>					
TADT		OPT	0	99	INTEGER
<i>Repeat ticket</i>					
SPIE		OPT	0	1	INTBFHX
<i>Re-open ticket</i>					
SRIE		OPT	0	1	INTBFHX

Explanations:

WALO: **delete flag :**
 0 = Data record is being modified or recreated
 1 = Data record is initialised with standard values

COKM: **Constant key function**
 0 Fix unit price
 1 Set tare value
 2 set unit price and tare value

PTLM: **Print product and total label**
 0 Print product and total label
 1 Only print total label
 2 Only print product label

SPOP: Item operator display mode

Bit field

Bit no.:

2 1 0

| | 1----- Display for total/subtotal

| 1----- Display after registration

1----- Display for sheet cancellation

RGFN: Cash register functions

0 Cash drawer does not open

1 Cash register opens **with/without sale** with total and operator key
e.g * **u. 1**Registration **possible** with open cash drawer.2 Cash register only opens **with sale** with total and operator key
e.g * **u.1**Registration **possible with open cash drawer.**3 Cash register opens **without** sale with total and operator key
e.g * **u. 1**Registration **not possible** with open cash drawer.4 Cash register only opens **with sale** with total and operator key
e.g * **u. 1** Registration **not possible** with open cash drawer.**CIOM: Payment in/out**

0 = Without payment in/out

1 = Payment in/out with a currency

2 = Combined payment in/out in local currency and foreign currency

TADT: Talon printout delay time

0 Without talon

1..99 With talon the input number corresponds to the delay time in seconds

SPIE: Repeat ticket

0 Without repeat ticket

1 With repeat ticket

SRIE: Re-open ticket

0 without ticket re-opening

1 With ticket re-opening

6.3 B2PL Setting for price labeling

Table : B2PL	Local table	Number of attributes: 3
--------------	-------------	-------------------------

Description : Settings for price labeling

<i>Description</i>					
Desc.	KEY	OPT	MinValue	MaxValue	Data type
<i>delete flag</i>					
WALO	KEY	FIX	0	1	INTEGER
<i>Price labeling mode</i>					
LBMO		OPT	0	2	INTEGER
<i>Multiple label print delay</i>					
MULA		OPT	0	99	INTEGER

Explanations:

WALO: **delete flag :**

0 = Data record is being modified or recreated
1 = Data record is initialised with standard values

LBMO: **Price labeling modes**

- 0 Price labeling. Item printouts with selling price "0.00" are not possible
- 1 Weight labeling. Item printouts with selling price "0.00" are possible, unit and selling price are also printed
- 2 Weight labeling only. Item printouts with selling price "0.00" are possible. If the unit price is "0.00", unit and selling price are not printed.

MULA: **Multiple label print delay**

0 labels are printed consecutively without delay
1..99 delay time when printing the labels

6.4 B2LC Label configuration

Table : B2AP	Local table	Number of attributes: 19
--------------	-------------	--------------------------

Description : Label configuration

Description					
Desc.	KEY	OPT	MinValue	MaxValue	Data type
<i>delete flag</i>					
WALO	KEY	FIX	0	1	INTEGER
<i>Predefined label layout</i>					
ALID	-	OPT	0	99	INTEGER
<i>BLD label layout</i>					
BLDI	-	OPT	0	99	INTEGER
<i>Label height</i>					
HIGH	-	OPT	28	148	INTEGER
<i>Label distance</i>					
PGAP	-	OPT	8	40	INTEGER
<i>Text sell-by</i>					
PSBD	-	OPT	0	2	INTBFHX
<i>EAN code on label</i>					
EANC	-	OPT	0	2	INTBFHX
<i>EAN code height on label</i>					
EHIG	-	OPT	5	22	INTEGER
<i>Tare printout on label</i>					
TARE	-	OPT	0	1	INTBFHX
<i>Header text print-out on label</i>					
HEAD	-	OPT	0	99	INTEGER
<i>Ttext for sell-by date on label</i>					
SBDT	-	OPT	0	9999999999	INTEGER
<i>Header logo on label</i>					
LOGO	-	OPT	0	9999	INTEGER
<i>Text field start on label</i>					
TFBE	-	OPT	0	84	INTEGER
<i>Data field start on label</i>					
DFBE	-	OPT	0	84	INTEGER
<i>Data field monitoring on/off on label</i>					
DFIC	-	OPT	0	1	INTBFHX
<i>Text format for texts on label</i>					
TLAY	-	OPT	0	110	INTEGER
<i>Price range and texts on label</i>					
FRAM	-	OPT	0	7	INTBFHX
<i>Operator, device- or digi code on label</i>					
SDDI	-	OPT	0	3	INTEGER
<i>Print foreign currency on label</i>					
FCSE	-	OPT	0	4	INTEGER
<i>label 180° rotated</i>					
P180	-	OPT	0	1	INTBFHX

<i>Text packed on</i>					
ATPO	-	OPT	0	999999999	INTEGER
<i>Text use by</i>					
ATUB	-	OPT	0	999999999	INTEGER

Explanations:**WALO: delete flag :**

0 = Data record is being modified or recreated
1 = Data record is initialised with standard values

ALID: Label type

0 Label from BLD. See BLDI

1.. 99 predefined label.

There are pre-defined label layouts in the device. If you are working with BLD (Bizerba Label Designer) you can select freely-definable. Pre-defined labels have a text and a data area. The text area contains text, e.g. the product text. The data area contains data on the product, weight, sales price sum EAN code, etc. Text and data field areas can be adjusted. No difference is made between text and data field range on BLD labels.

BLDI: BLD label layout :

1..99 Select BLD label layout (ALID = 0)

HIGH: Label height :

28..148 mm 54 and 55 mm are not permitted

PGAP: Label distance :

8 .. 40 Dots (8 Dots = 1mm)

PSBD: Packaged-on and sell-by date:

Bit field

Bit no.:

1 0

|

|

1 - Sell-by date is printed

|

1 packaged date is printed

EANC: EAN code print-out on label:

Bit field

Bit no.:

1 0

|

|

1 - Total EAN code is printed

|

1 product EAN code is printed

EHIG: **EAN code height :**
5..22 mm

TARE: **Tare printout:**

Bit field
Bit no.:
0
|
1 -Tare is printed

HEAD: **Header text printout :**
0 without header text
1..99 Number of the header text to be printed

SBDT: **Text for sell-by :**
0 Standard text
1..999999999 Number of additional text to be printed

LOGO: **Logo print-out in header row :**
0 without header logo
1..9999 Number of the header logo to be printed

TFBE: **Start of text field :**
0..84
Predefined labels are separated into a text and data field area. The text field contains the product text. The text field can be adjusted upwards or downwards within the label:
0: The text field begins right after the label header.
1..84: The text field is adjusted downwards in mm.
(0 position starts after the label header)
In BLD labels there is no subdivision of this type.

DFBE: **Start of data field :**
0..84
Predefined labels are separated into a text and data field area. The data field contains product data including unit price, weight, sales price, total, EAN code, etc. The data field can be adjusted upwards or downwards within the label.
0: The data field is moved to the end of the label.
1..84: The data field is always adjusted downwards in mm.
(0 position starts after the label header)
In BLD labels there is no subdivision of this type.

DFIC: Data field monitoring on/off:

Bit field
 Bit no.:
0
 |
 1 - data field monitoring is on

When data field monitoring is activated, this prevents texts from the text field being written to the data field.

Text is cut-off on the label.

In the case of labels on tickets, the print-out is simply extended.

If monitoring is deactivated, texts from the text field might continue into the data field.

TLAY: Text format for product texts:

0 The texts are represented according to the formatting in the PLU text or in line with the text sizes provided in the layout.

1..110 Select a defined font for text.

FRAM: Price range and texts

Bit field
 Bit no.:
 5 4 3 2 1 0
 | | | | | |
 | | | | | 1- Print price range
 | | | | 1----- Print "price"
 | | | 1----- Print "net" or "item"
 | | 1----- Print "sell by"
 | 1----- Print. "packed"
 1----- Print "tare"

Bit no.:
 6
 |
 1-Print. Additional text

SDDI: Operator no., machine parameter check digit on label

0 without output

1 with operator no.

2 with key machine figure

3 with digit code

The label contains a field intended for output. The selected number is inserted into the field.

FCSE: Foreign currency

Bit Nr.:

3 2 1 0

| | | |

| | | 1- Only print total

| | 1----- Print selling price and total

| 1----- Print unit price, selling price and
total

1----- Print conversion rate

P180: rotate label 180°

0: no rotation

1: rotation

ATPO: Text packed on

0 Standard Text

1..999999999 number of text to be printed

ATUB: Text use by

0 Standard text

1..999999999 number of text to be printed

6.5 B2SC Ticket layout configuration

Table : B2SC	Local table	Number of attributes: 14
--------------	-------------	--------------------------

Description : Ticket layout configuration

Description					
Desc.	KEY	OPT	MinValue	MaxValue	Data type
<i>delete flag</i>					
WALO	KEY	FIX	0	1	INTEGER
<i>EAN code on ticket</i>					
EANC	-	OPT	0	3	INTBFHX
<i>EAN code height on ticket</i>					
EHIG	-	OPT	5	22	INTEGER
<i>Tare printout on ticket</i>					
TARE	-	OPT	0	2	INTEGER
<i>Header text printout on ticket</i>					
HAED	-	OPT	0	99	INTEGER
<i>Footer text printout on ticket</i>					
FOOT	-	OPT	0	99	INTEGER
<i>Additional text for foreign currency on ticket</i>					
FCAD	-	OPT	0	999999999	INTEGER
<i>Additional text for rounding on ticket</i>					
RDAD	-	OPT	0	999999999	INTEGER
<i>Font change on ticket</i>					
TLAY	-	OPT	0	110	INTEGER
<i>Number of PLU text lines to be printed on ticket</i>					
NRLI	-	OPT	0	9	INTEGER
<i>Print ticket counter</i>					
PRSC	-	OPT	0	1	INTBFHX
<i>Print currency on ticket</i>					
FCSE	-	OPT	0	4	INTBFHX
<i>Print amount paid in/due out</i>					
PRPY	-	OPT	0	1	INTBFHX
<i>Print additional texts</i>					
PRGT	-	OPT	0	1	INTBFHX

Explanations:**WALO: delete flag :**

0 = Data record is being modified or recreated
1 = Data record is initialised with standard values

EANC: EAN code print-out on ticket:

Bit field

Bit no.:

2 1 0

| | |

| | 1- Talon EAN code is printed

| 1 - Total EAN code is printed

1 product EAN code is printed

EHIG: EAN code height:

5.22 mm

TARE: Tare printout on ticket:

0 = no tare value on ticket

1 Tare value net weight is identified by N
(country-dependent character)

2 Tare value on ticket net weight is not identified with N

HEAD: Header text printout :

0 without header text

1..99 Number of the header text to be printed
(Text from KFST)

FOOT: Footer text printout :

0 without footer text printout

1..99 Number of the footer text to be printed
(Text from KFST)

FCAD: Additional text for foreign currency on ticket

0 without text

1..999999999 Number of the text to be printed
(Text from ATST)

RDAD: Additional text for rounding on ticket

0 without text

1..999999999 Number of the text to be printed
(Text from ATST)

TLAY: Text format for product texts:

0 The texts are represented according to the formatting in the PLU text or
in line with the text sizes provided in the layout.

1..110 Select a defined font for text.

6.6 B2PS PLU settings

Table : B2PS	Local table	Number of attributes: 8
--------------	-------------	-------------------------

Description : PLU settings 1

<i>Description</i>					
Desc.	KEY	OPT	MinValue	MaxValue	Data type
<i>delete flag</i>					
WALO	KEY	FIX	0	1	INTEGER
<i>10-key keypad for unit price or PLU input mod.</i>					
KM10	-	OPT	0	2	INTEGER
<i>PLU input of number of digits via decimal keypad</i>					
NDRK	-	OPT	2	6	INTEGER
<i>Direct PLU product key mode</i>					
DPKM	-	OPT	0	2	INTEGER
<i>PLU product range limit min. value</i>					
PMIN	-	OPT	1	9999	INTEGER
<i>PLU product range limit max. value</i>					
PMAX	-	OPT	1	9999	INTEGER
<i>PLU product range limit min. value (6 digit PLU number)</i>					
PMI6	-	OPT	1	999999	INTEGER
<i>PLU product range limit max. value (6 digit PLU number)</i>					
PMA6	-	OPT	1	999999	INTEGER

Explanations:

WALO: **delete flag :**

0 = Data record is being modified or recreated
1 = Data record is initialised with standard values

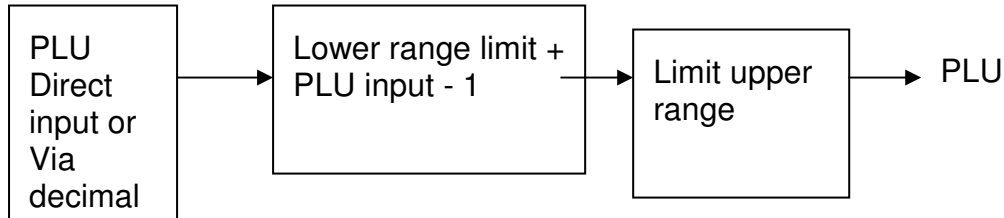
KM10: **10-key keypad for unit price or PLU input**

0 enter unit price via keyboard switch to PLU input via shift key
1 PLU number input via decimal keypad. Switching to unit price input with shift key
2 PLU no. input only via 10-key keypad, unit price input not possible

NDRK: **PLU input of number of digits via decimal keypad**

2..4 Input 2..4 digits

DPKM: Direct PLU product key mode
 0 PLU product key no. corresponds to the PLU no.
 1 Direct PLU keypad is disabled
 2 PLU results from PLU product key assignment, see B2BK
 (In this case the range limit does not apply, see below)



The PLU number is limited by the specified range.

The lower range limit is assigned to PLU number 1

Example:

501-600 limitation

PLU input 1 produces PLU 501, this is within the 500..600 range

100 results in PLU 600 and is within the 500..600 range

101 results in PLU 601 this is not within the range and is rejected

PMIN: Lower range limit
 1..9999

PMAX: Upper range limit
 1..9999

PMI6: PLU product range limit min. value (6 digit PLU number)
 1..999999

PMA6: PLU product range limit max. value (6 digit PLU number)
 1..999999

Notice: PMIN and PMAX are used to be compatible with old backups.
 Old backups can be restored, but new backups deliver
 PMIN = 0 and PMAX = 0.

6.7 FUCO FUCN local key arrangement

Table : FUCO	Local table	Number of attributes: 4
--------------	-------------	-------------------------

Description : Keyboard function assignment
--

Description					
Desc.	KEY	OPT	MinValue	MaxValue	Data type
<i>delete flag</i>					
WALO	KEY	FIX	0	1	INTEGER
<i>Keys-function code</i>					
TSFC	KEY	FIX	-32768	32767	INTEGER
<i>Key data</i>					
TSDA	-	FIX	0	32	INTBFHX

Tabelle : FUCN	Local table	Number of attributes: 4
----------------	-------------	-------------------------

Description : Keyboard function assignment
--

Beschreibung					
Bez.	KEY	OPT	MinWert	MaxWert	Datentyp
<i>delete flag</i>					
WALO	KEY	FIX	0	1	INTEGER
<i>Keys-function code</i>					
TSFC	KEY	FIX	-32768	32767	INTEGER
<i>Key data</i>					
TSDA	-	FIX	0	9999999999999	INTEGER

This tables are primarily used to assign scale-specific numerical fixed keys (except for operator keys).

Scheduled applications are, for example:

SB PLU keys
Fixed tare keys

The FUCO table contains only key data with 32 bit content as a subset of the FUCN table. The FUCN table on the other hand, encompasses all key data, including SB keys with 13-digit EAN code. (only for SC, BCII ignores 13 digit values)

WALO: delete flag :
 0 = Data record is being modified or recreated
 1 = Data record is deleted

TSFC: Keyboard codes:

**Direct PLU or SB-PLU keys KEYB_PLU_x
 (x:= PLU number)**

KEYB_PLU_1	0x8000	32768
KEYB_PLU_2	0x8001	32769
KEYB_PLU_3	0x8002	32770
.	.	.
.	.	.
.	.	.
KEYB_PLU_254	0x80FD	33021
KEYB_PLU_255	0x80FE	33022
KEYB_PLU_256	0x80FF	33023

**Tare fixed value memory keys KEYB_TARA_x
 (x := number of tare fixed value)**

KEYB_TARA_1	0x8200	33280
KEYB_TARA_2	0x8201	33281
KEYB_TARA_3	0x8202	33282
KEYB_TARA_4	0x8203	33283

TSDA: Key value.
 e.g. PLU number, TARE, etc.

6.8 B2FC currency settings

Table : B2FC	Local table	Number of attributes: 10
--------------	-------------	--------------------------

Description : Foreign currency settings

Description					
Desc.	KEY	OPT	MinValue	MaxValue	Data type
<i>delete flag</i>					
WALO	KEY	FIX	0	1	INTEGER
<i>Select foreign currency function</i>					
FCFN	-	OPT	0	2	INTEGER
<i>Select foreign currency display</i>					
FCDS	-	OPT	0	1	INTBFHX
<i>Foreign currency calculation method</i>					
FCTM	-	OPT	0	1	INTEGER
<i>Foreign currency characters for ticket and label layout</i>					
FCSG	-	OPT	0	12	VARSTR
<i>Foreign currency characters on label for price field</i>					
FCSP	-	OPT	0	12	VARSTR
<i>Decimal places in foreign currency</i>					
FCPC	-	OPT	0	3	INTEGER
<i>Foreign currency factor</i>					
FCFM	-	OPT	0	3	INTEGER
<i>Currency exchange rate (calculation factor)</i>					
FCCA	-	OPT	0	999999	INTEGER
<i>Rounding of foreign currency</i>					
FCRD	-	OPT	0	2	INTEGER

- WALO:** **delete flag :**
0 = Data record is being modified or recreated
1 = Data record is initialised with standard values
- FCFN:** **Select foreign currency function**
0 Euro/Foreign currency function from
1 Foreign currency function on
2 Euro function on (only possible for non-Euro scales)
- FCDS:** **Select foreign currency display**
0: Foreign currency display off
 (Euro or foreign currency key non-functional)

1: Foreign currency display on
 Display unit and selling price, total and/or subtotal and total in payment
 in/out mode
- FCTM:** **Foreign currency calculation method**
0 Total calculated through conversion
1 Total calculated through addition of selling price
- FCSG:** **Foreign currency characters for ticket and label layout**
Foreign currency characters as UTF8 coded text. Conclude with '\0' end
character.
- FCSP:** **Foreign currency character for label for price field**
Foreign currency characters as UTF8 coded text. Conclude with '\0' end
character.
- FCPC:** **Number of decimal places in foreign currency**
0 .. 3
- FCFM:** **Foreign currency factor**
0 .. 3 = factor 1..1000
Example
FW = Foreign currency
NW = National currency
Input FCCA = 101
FCFM = 3
=> 1000 FW = 1.01 NW
- FCCA:** **Currency exchange rate (calculation factor)**
e.g. 101
results in 0001.01 in the national currency, e.g. €
- FCRD:** **Rounding of foreign currency**
0 no rounding
1 mathematical rounding to 2
2 mathematical rounding to 5

6.9 B2NP Non-PLU settings

Table : B2NP	Local table	Number of attributes: 9
--------------	-------------	-------------------------

Description : Non-PLU settings

<i>Description</i>					
Desc.	KEY	OPT	MinValue	MaxValue	Data type
<i>delete flag</i>					
WALO	KEY	FIX	0	1	INTEGER
<i>Shelf-life days for non-programmed products</i>					
SBDN	-	OPT	0	32	INTBFHX
<i>EAN code for non-PLU weighed products</i>					
NIWE	-	OPT	0	14	VARSTR
<i>EAN code for non-PLU manual (item) products</i>					
NIMA	-	OPT	0	14	VARSTR
<i>EAN code for non-PLU minus products</i>					
NIMI	-	OPT	0	14	VARSTR
<i>EAN code for total</i>					
NITO	-	OPT	0	14	VARSTR
<i>Product group for non-PLU weighed products</i>					
NWWE	-	OPT	0	9999	INTEGER
<i>Product group for non-PLU manual (item) products</i>					
NWMA	-	OPT	0	9999	INTEGER
<i>Product group for non-PLU minus products</i>					
NWMI	-	OPT	0	9999	INTEGER

Explanations:

WALO: **delete flag :**
0 = Data record is being modified or recreated
1 = Data record is initialised with standard values

SBDN: **Sell-by date for non- PLU**
The sell-by date is printed similar to non-PLU. The product (PLU) only if the 00 sell-by days are allocated to the PLU.
Details of data as system time ([sec] since 1.1.1970):1970

NIWE: **EAN code for NON-PLU weighed products**
Description of EAN code. See PLST (ECO1)

NIMA: **EAN code for NON-PLU manual (item) products**
Description of EAN code. See PLST (ECO1)

NIMI:	EAN code for NON-PLU minus products Description of EAN code. See PLST (ECO1)
NITO:	EAN code for total Description of EAN code. See PLST (ECO1)
NWWE:	Product group for non-PLU weighed products 0 .. 9999
NWMA:	Product group for non-PLU manual (item) products 0 .. 9999
NWMI:	Product group for non-PLU minus products 0 .. 9999

6.10 B2O1 Further settings 1

Table : B2O1	Local table	Number of attributes: 4
--------------	-------------	-------------------------

Description : Further settings 1

<i>Description</i>					
Desc.	KEY	OPT	MinValue	MaxValue	Data type
<i>delete flag</i>					
WALO	KEY	FIX	0	1	INTEGER
<i>Year number format</i>					
YFOR	-	OPT	0	1	INTEGER
<i>Date format</i>					
DFOR	-	OPT	0	9	INTEGER
<i>Scale sensitivity</i>					
SSEN	-	OPT	0	9	INTEGER
<i>Keyboard selection</i>					
KSEL	-	OPT	0	3	INTEGER

Explanations:

WALO: **delete flag :**
 0 = Data record is being modified or recreated
 1 = Data record is initialised with standard values

YFOR: **Year number format**
 0 representation of year number with 2 digits
 1 representation of year number with 4 digits

DFOR: **Date format**

00 = Day-Month-Year numeric	e.g. 21.07.06
01 = Day-Month-Year alphanumeric	e.g. 21July06
02 = Day-Month alphanumeric	e.g. 21July
03 = Month Year alphanumeric	e.g. July03
04 = Month Day Year American numeric	e.g. 07.21.06
05 = Month Day Year American alphanumeric	e.g. Juli21.06
06 = Year Month Day Hungarian numeric	e.g. 06.07.21
07 = Year Month Day Hungarian alphanumeric	e.g. 06Juli21
08 = Day Month Year Numeric	e.g. 21-07-06
09 = Day Month Year Alphanumeric	e.g. 21-July-06

SSEN: **scale sensitivity**
0 scale sensitive (Standard)
...
9 scale less sensitive

KSEL: **Keyboard selection**
0 Standard 84-keyborad
1 84-Keyboard 24er self service frame
2 84- Keyboard mit 30er self service frame
3 84- Keyboard reprogrammed (see KFTF)

6.11 KFTF Keyboard layout

Table : KFTF	Local table	Number of attributes: 19
--------------	-------------	--------------------------

Description : Keyboard layout

Description					
Desc.	KEY	OPT	MinValue	MaxValue	Data type
<i>delete flag</i>					
WALO	KEY	FIX	0	1	INTEGER
<i>Keyboard number</i>					
TANR	KEY	FIX	1	131	INTEGER
<i>Line number</i>					
ZENR	KEY	FIX	0	16	INTEGER
<i>Keys-function code in column</i>					
SP01	-	FIX	0	16	INTBFHX
<i>Keys-function code in column</i>					
SP02	-	FIX	0	16	INTBFHX
<i>Keys-function code in column</i>					
SP03	-	FIX	0	16	INTBFHX
<i>Keys-function code in column</i>					
SP04	-	FIX	0	16	INTBFHX
<i>Keys-function code in column</i>					
SP05	-	FIX	0	16	INTBFHX
<i>Keys-function code in column</i>					
SP06	-	FIX	0	16	INTBFHX
<i>Keys-function code in column</i>					
SP07	-	FIX	0	16	INTBFHX
<i>Keys-function code in column</i>					
SP08	-	FIX	0	16	INTBFHX
<i>Keys-function code in column</i>					
SP09	-	FIX	0	16	INTBFHX
<i>Keys-function code in column</i>					
SP10	-	FIX	0	16	INTBFHX
<i>Keys-function code in column</i>					
SP11	-	FIX	0	16	INTBFHX
<i>Keys-function code in column</i>					
SP12	-	FIX	0	16	INTBFHX
<i>Keys-function code in column</i>					
SP13	-	FIX	0	16	INTBFHX
<i>Keys-function code in column</i>					
SP14	-	FIX	0	16	INTBFHX
<i>Keys-function code in column</i>					
SP15	-	FIX	0	16	INTBFHX
<i>Keys-function code in column</i>					
SP16	-	FIX	0	16	INTBFHX

Explanations:

- WALO:** **delete flag :**
0 = Data record is being modified or recreated
1 = Data record is deleted
- TANR:** **Keyboard number:**
Number of stored keyboard
- ZENR:** **Line number:**
Row 1 identifies the row of keys on the bottom edge of the scales' housing.
- SPXX:** **Column number:**
Column 1 describes the row of keys on the left edge of the scales' housing.

The keyboard is separated into four levels.

Level	Keynumber	Codes
The funktion level describes the function of the key	0x10 (decimal 17)	Function codes see table
capital characters level active in text editor mode (Shift - key and key must be pressed)	0x01	Unicode (UC16)
small characters level active in text editor mode	0x02	Unicode (UC16)
Control characters level active in text editor mode (Ctrl - key and key must be pressed)	0x03	Unicode (UC16)

In text level the codes are coded in **unicode**.

To activate the settings in the KFTF set menu 631 to value 3 (B2O1=> KSEL = 3)

The following function codes can be used in the function level:

Function code	Function
0x0c	Paper feed
0x8000..0x80FF	Direct PLU key 1.. 256
0x8100..0x800F	Direct operator keys 1..16
0x8200..0x8203	Tare fixed value key 1..4
0x8302	Tare
0x8303	Subtotal
0x8304	Total
0x8305	Constant
0x8306	Manual
0x8307	Minus
0x8308	Reversal
0x830A	Multiplication
0x830C	Switch PLU price
0x832C	Setting the scales to ZERO
0x8405	Foreign currency key
0x8406	Key for price per part
0x8512	Date key
0x8515	Print key

The following function codes are fix in and can't be reprogrammed

Function code	Function
0x0d	Adopt return input
0x1b	ESC
0x30 .. 0x39	Digits 0 to 9
0x1000	Up arrow
0x1001	Down arrow
0x8301	Delete (C key)
0x839B	Mode menu key
0x83F0	Shift
0x83F1	Control

In the text levels 0x2190 (arrow left) und 0x2192 (arrow right) can not be reprogrammed.

7 Commissioning

7.1 B2SU Commissioning BCII

Tabelle : B2SU	locale table	Number of attributes: 8
----------------	--------------	-------------------------

Description : Commissioning BCII

Description					
Desc.	KEY	OPT	MinValue	MaxValue	Data type
<i>delete flag</i>					
WALO	KEY	FIX	0	1	INTEGER
<i>Check validity of commissioning</i>					
CHSU	KEY	FIX	1	1	INTEGER
<i>Number PLU</i>					
NRPL	-	OPT	100	6000	INTEGER
<i>Number of extra texts per PLU</i>					
GTPP	-	OPT	0	4	INTEGER
<i>Number of BLD text fields per PLU</i>					
BTPP	-	OPT	0	10	INTEGER
<i>Number of BLD logo fields per PLU</i>					
BLPP	-	OPT	0	3	INTEGER
<i>Number general texts</i>					
NRGT	-	OPT	50	1000	INTEGER
<i>Number of entries (receipt storage)</i>					
NR6B	-	OPT	50	1200	INTEGER

Description:

- WALO:** **delete flag :**
0 = Data record is being modified or recreated
1 = Data record is deleted
- CHSU:** **Check validity of commissioning**
1= check and store iv valid commissioning
- NRPL:** **Number of PLU :**
Number of PLU which can be stored
- GTPP:** **Number of extra texts per PLU**
Number of extra texts which can be added, to the PLU Text of the PLU
- BTPP:** **Number of BLD text fields per PLU:**
Number of BLD text fields, which can be allocated to the PLU
- BLPP:** **Number of BLD logo fields per PLU:**
Number of BLD logo fields, which can be allocated to the PLU
- NRGT:** **Number of general texts:**
Number of extra texts. (Text pool which contains the extra texts)
- NR6B:** **Number of entries (receipt storage):**

Notice: The commissioning will be activated after RAM clear.
The message B2CR clears the RAM and reboots the scale.

8 Messages

This chapter only describes the meaning of individual messages. Details on accessing the program are included in the user manuals on SWCom or SWLSV1.

8.1 PROG Program version

The answer to the PROG message (N PROG) is supplied by the PROGM table, which detects the current program version of the scales connected to the host.

Table : PROG	Local message	Number of attributes: 8
--------------	---------------	-------------------------

Description : Program version BCII

Description					
Desc.	KEY	OPT	MinValue	MaxValue	Data type
<i>Bit field</i>					
FLAG	KEY	FIX	0	8	INTBFHX
<i>Version Sx-Tcp</i>					
VERS	KEY	FIX	0	8	INTBFHX
<i>Revision Sx-Tcp</i>					
REV	KEY	FIX	0	99	INTEGER
<i>Date/time of when program was created Sx-Tcp</i>					
DATE	KEY	FIX	0	32	INTBFHX
<i>Program number BCII</i>					
PROG	KEY	FIX	0	15	STRING
<i>Database version</i>					
LDBV	-	OPT	0	99	INTEGER
<i>Database revision</i>					
LDBR	-	OPT	0	99	INTEGER
<i>Interfaces - code page</i>					
COPG	-	OPT	0	9999	INTEGER

Explanations:

FLAG: **Bit field**

Bit no.:				
7	6	5	4	3-0
				+ - application bits (insignificant)
			1	- License available for data transmission
		1		- LTCP: Block backup at application level
	1			- LTCP: Host switched on at "BLK"
1				- Password active

COPG: **Interfaces - code page**

1001 = UTF8
The BCII only supports UTF8!!

8.2 TIME date/time

The time message (N TIME) sets the system time of the scales.

Table : HOUR	global message	Number of attributes: 1
--------------	----------------	-------------------------

Description : System time

<i>Description</i>					
Desc.	KEY	OPT	MinValue	MaxValue	Data type
<i>System time</i>					
UUHR	KEY	FIX	0	32	INTBFHX

;

Explanations:

UUHR: **System time**
 ([sec] since 1.1.1970):

When the TIME message is transmitted with SWCom, the scales directly adopts the host system time.

8.3 WARE Scales status

The answer to the GOODS message (N GOODS) is supplied by the scales number of the BCII scales.

Table : WARE	global message	Number of attributes: 2
--------------	----------------	-------------------------

Description : Scales reference (reply from BCII)
--

<i>Description</i>					
Desc.	KEY	OPT	MinValue	MaxValue	Data type
<i>Scales reference (target)</i>					
wrso	KEY	FIX	0	32	INTBFHX
<i>Active scales (actual)</i>					
wris	KEY	FIX	0	32	INTBFHX

Explanations:

WRSO:

WRIS: **Bit field: The scales address specifies the position.**

Bit 0: Scale 1

Bit 31: Scale 32

BCII do not constitute a scales network where master data is distributed.
The BCII always supplies wrso) wris whereby the bit position of the custom scales number.

8.4 B2CR clear RAM reboot scale

B2CR clears the RAM and reboots the scale.
During reboot the connection to the host is disconnected.

