Forecourt Controller Rohé BOX69

Programming Manual – UDC POS Interface

Version 1.6 / 10.07.2014

TABLE OF CONTENTS

1. PURPOSE

The following document describes the supported by BOX69 commands and the right approach for controlling pumps connected to Box69. The document is addressed to all POS/CD developers which want to use BOX69 as forecourt controller and protocol converter connected to their system.

2. COMMUNICATION DATA FORMAT

The BOX69 – UDC communication channel which is dedicated to POS connection supports the following parameters:

Data transfer	Asynchronous
Bit Rate	9600 bits per second
Data bits	8
Stop bits	1
Parity	None

Table 1

For detailed information about BOX69 hardware and connectors set-up see "Installation Rohé BOX69 (x.x).pdf

3. 'DOUBLETALK' FORMAT OF THE PROTOCOL

Data transmission between the controller device (CD or POS) and Box69 uses so called "doubletalk" format of the protocol, in which each 8-bit byte is immediately followed by its binary complement.
For example, if we have 90 hex (10010000) transmitted, the following 'doubletalk' byte (**identified as ~90h in this document**) would be 6F hex (01101111). The 'doubletalk' format of the protocol provides bit-for-bit error detection.

Remark: For simplification the description of all binary complement bytes in this document is done using the symbol "~" and the leading "straight" byte.

4. SUPPORTED COMMANDS

The table below describes all the commands Box69 supports from the Tokheim UDC protocol:

Command		Description
Fx	A0	Request fuelling point ID
Fx	A1	Request fuelling point display data
Fx	A2	Request fuelling point status
Fx	A3	Halt sale and turn valves off
Fx	A5	Authorize fuelling point
Fx	A6	Send data for fuelling point main display
Fx	A9	Request single product stand-alone fuelling point totals
Cx	A1	Request activated nozzle and MOP
Cx	A2	Acknowledge de-activated nozzle
Cx	A3	Send CASH / CREDIT prices for fuelling point auxiliary displays

Table 2 Where "x" is the pump address from 1 to 16 (i.e. 0h to Fh)

3.1. Fx A0 - Request fuelling point ID (FP ID)

3.1.1. Controller command

Fx	~Fx	A0	~A0

Where:
Fx = Fuelling point address
A0 = Function code

3.1.2. Box69 response

A	~A

Where:
A = Fuelling point ID (hex)

3.1.3. Example controller command

F6	~F6	A0	~A0

Request fuelling point 7 ID (address 06 Hex).

3.1.4. Example Box69 response

99	~99

Fuelling point 7 ID = 99

3.2. Fx A1 - Request fuelling point display data

3.2.1. Controller command

Fx	~Fx	A1	~A1

Where:
Fx = Fuelling point address
A1 = Function code

3.2.2. Box69 response

A	~A	B	~B	C	~C	D	~D	E	~E	F	~F	G	~G	H	~H	I	~I

A	Price/Unit Vol. (1sb)
B	Price/Unit Vol. (msb)
C	Money (1sb)
D	Money (2sb)
E	Money (msb)
F	Volume (1sb)
G	Volume (2sb)
H	Volume (msb)
I	Fuelling point status byte

3.2.3. Example controller command

F2	~F2	A1	~A1

Request fuelling point 3 (address 02 Hex) display data.
For example - fuelling point 3 is Kienzle-ER3 dispenser.

3.2.4. Example Box69 response

29	~29	11	~11	19	~19	16	~16	00	~00	37	~37	43	~43	01	~01	20	~20

Currently displayed sale data is:

Price	1.129
Money	0016.19
Volume	014.337

The dispenser id idle.

3.3. Fx A2 - Request fuelling point status

3.3.1. Controller command

Fx	~Fx	A2	~A2

Fx	Fuelling point address
A2	Function code

3.3.2. Box69 response

A	~A

Where:
A = Fuelling point status byte

3.3.3. Example controller command

F3	~F3	A2	~A2

Request fuelling point 4 (address 03 Hex) status.

3.3.4. Example Box69 response

20	~20

Fuelling point 4 is idle. (See table 3 for all the statuses used by Box69).

Status Byte	Fuelling Point State
2F	Uninitialized fuelling point (FP); fuelling point requires main display data sent with function code A6
20	Idle fuelling point – nozzle down.
A0	Idle fuelling point – nozzle up – fuelling point is not authorized. This status is a request for fuelling point authorization (calling).
90	1) Nozzle up – fuelling point authorized – no valves open. Returned after an A5 Authorize command is accepted, but before the valves open, during the segment check and display blanking cycles. 2) Sale in progress, only slow flow valve open. Pulser check active, waiting for 10th pulse.
D0	Sale in progress: 1) Only slow flow valve open. Pulser check active, waiting for 10^{th} pulse. 2) The (only) valve is open
F0	Sale in progress, both valves open.
91	Sale terminated (due to lowered nozzle or max delivery reached).
91	Fuel delivery stopped by an island preset.
99	Nozzle lowered during halted sale; sale terminated.

Table 3
All statuses used in the Box69 state-machine

3.4. Fx A3 - Halt sale and turn valves off (emergency stop command)

3.4.1. Controller command

Fx	~Fx	A3	~A3

Where:
Fx = Fuelling point address
A3 = Function code

3.4.2. Box69 response

A	~A

Where:
A = Fuelling point status byte (see table statuses)

3.4.3. Example controller command

F7	~F7	A3	~A3

Halt fuelling point 8 (address 07 Hex).

3.4.4. Example Box69 response

98	~98

Halting fuelling point 8.

3.5. Fx A5 - Authorize fuelling point

3.5.1. Controller command

Fx	~Fx	A5	~A5	A	~A	B	~B	C	~C	D	~D	E	~E	F	~F	G	~G	H	~H	I	~I

Fx	Fuelling point address
A5	Function code
A	Slow Flow Offset
B	Price/Unit Vol. (1sb)
C	Price/Unit Vol. (msb)
D	Maximum delivery – Money (1sb)
E	Maximum delivery – Money (2sb)
F	Maximum delivery – Money (msb)
G	Maximum delivery – Volume (1sb)
H	Maximum delivery – Volume (2sb)
I	Maximum delivery – Volume (msb)

3.5.2. Box69 response

A	~A

Where:
A = Fuelling point status byte

3.5.3. Example controller command

F3	~F3	A5	~A5	05	~05	19	~19	10	~10	97	~97	64	~64	05	~05

33	~33	44	~44	55	~55

Authorize fuelling point 4 (address 03 Hex) for a sale using the following values:

Slow flow offset	05
Fuel price	1.019
Maximum money limit	564.97
Maximum volume limit	554.433

3.5.4. Example Box69 response

90	~90

3.6. Fx A6 - Send data for fuelling point main display

3.6.1. Controller command

Fx	~Fx	A6	~A6	A	~A	B	~B	C	~C	D	~D	E	~E	F	~F	G	~G	H	~H

Fx	Fuelling point address	
A6	Function code	
A	Price/Unit Vol. (1sb)	
B	Price/Unit Vol. (msb)	
C	Money (1sb)	
D	Money (2sb)	
E	Money (msb)	
F	Volume (1sb)	
G	Volume (2sb)	
H	Volume (msb)	

3.6.2. Box69 response

A	~A

Where:
A = Fuelling point status byte (see table statuses)

3.6.3. Example controller command

F3	~F3	A6	~A6	19	~19	10	~10	77	~77	88	~88	09	~09

34	~34	03	~03	97	~97

Send initial data for main price displays on fuelling point 4(address 03 Hex) as follows:

fuel price	1.019
money	0988.77
volume	970.334

3.6.4. Example response

20	~20

Fuelling point 4 initialized.

3.7. Fx A9 - Request multi – product stand alone fuelling point nozzle totals

3.7.1. Controller command

Fx	~Fx	A9	~A9	A	~A

Fx	Fuelling point address	
A9	Function code	
A	Type of totals requested	

TOTALS REQ. CONTROL BYTE

Bit	Name	Value
0	run/shift	0: running totals / 1: shift totals (only running are supported)
1 2 3	Nozzle number (0-7)	0 - nozzle1 1 - nozzle2,… 7 - nozzle8
4	cash/credit	0: cash / 1: credit (only cash is supported)
5	no use	
6 7		0 0 - all nozzles totals 0 1 - single nozzle totals

3.7.2. Example controller command

F6	~F6	A9	~A9	01	~01

Request fuelling point 7 (address 06 Hex) cash shift totals.

3.7.3. Example response

	Volume										Money									
Product 1	90	~90	78	~78	56	~56	34	~34	12	~12	21	~21	43	~43	65	~65	87	~87	09	~09
Product 2	55	~55	44	~44	33	~33	22	~22	11	~11	00	~00	99	~99	88	~88	77	~77	06	~06
Product 3	66	~66	77	~77	88	~88	99	~99	00	~00	11	~11	22	~22	33	~33	14	~14	00	~00
Product 4	00	~00	50	~50	00	~00	00	~00	00	~00	00	~00	05	~05	00	~00	00	~00	00	~00
Product 5	90	~90	78	~78	56	~56	34	~34	12	~12	21	~21	43	~43	65	~65	87	~87	09	~09
Product 6	55	~55	44	~44	33	~33	22	~22	11	~11	00	~00	99	~99	88	~88	77	~77	06	~06
Product 7	66	~66	77	~77	88	~88	99	~99	00	~00	11	~11	22	~22	33	~33	14	~14	00	~00
Product 8	00	~00	50	~50	00	~00	00	~00	00	~00	00	~00	05	~05	00	~00	00	~00	00	~00
Status	20	~20																		

Fuelling point 7 cash shift totals are:

Nozzle 1 volume	1234567.890
Nozzle 1 money	09876543.21
Nozzle 2 volume	1122334.455
Nozzle 2 money	06778899.00
Nozzle 3 Volume	0099887.766
Nozzle 3 money	00143322.11
Nozzle 4 volume	0000005.000
Nozzle 4 money	00000005.00
Nozzle 5 volume	1234567.890
Nozzle 5 money	09876543.21
Nozzle 6 volume	1122334.455
Nozzle 6 money	06778899.00
Nozzle 7 Volume	0099887.766
Nozzle 7 money	00143322.11
Nozzle 8 volume	0000005.000
Nozzle 8 money	00000005.00
The fuelling point is idle	20

3.7.4. Fx A9 - Request single – product standalone fuelling point totals

3.7.5. Example controller command

F6	~F6	A9	~A9	40	~40

Request single nozzle, cash total, nozzle-1, running total, fuelling point 7 (address 06 Hex).

3.7.6. Example response

	Volume										Money									
Product 1	90	~90	78	~78	56	~56	34	~34	12	~12	21	~21	43	~43	65	~65	87	~87	09	~09
Status	20	~20																		

Fuelling point 7 single nozzle, cash total, nozzle-1, running total is:

Nozzle 1 volume	1234567.890
Nozzle 1 money	09876543.21
The fuelling point is idle	20

3.8. Cx A1 - Request activated nozzle and mop

3.8.1. Controller command

Cx	~Cx	A1	~A1

Where:
Cx = Fuelling point address for Group III dispensers
A1 = Function code

3.8.2. Box69 response

A	~A

Where A is a bit-mapped number:
bit 7 = method of payment (0 = cash, 1 = credit)
bits 6-0 = nozzle number (0-8, 0 = no nozzle selected)

3.9. Cx A2 - Acknowledge deactivated nozzle

3.9.1. Controller command

Cx	~Cx	~A2	A2

Where:
Cx = Fuelling point address
A2 = Function code

3.9.2. Box69 response

B0	~B0

Where B0 = Acknowledge nozzle deactivation.

3.9.3. Example controller command

C6	~C6	A2	~A2

Acknowledge deactivated nozzle on fuelling point 7 (address 06 Hex)

3.9.4. Example Box69 response

B0	~B0

Nozzle deactivation acknowledged

3.10. Cx A3 - Send CASH / CREDIT prices for fuelling point auxiliary displays

3.9.5. Controller command

Cx	~Cx	A3	~A3	A	~A	B	~B	C	~C	D	~D	E	~E	F	~F	G	~G	H	~H

I	~I	J	~J	K	~K	L	~L	M	~M	N	~N	O	~O	P	~P

Cx	Fuelling point address
A3	Function code
A (4 bytes)	Product 1 Cash & Credit Prices
B (4 bytes)	Product 2 Cash & Credit Prices
C (4 bytes)	Product 3 Cash & Credit Prices
D (4 bytes)	Product 4 Cash & Credit Prices
E (4 bytes)	Product 5 Cash & Credit Prices
F (4 bytes)	Product 6 Cash & Credit Prices
G (4 bytes)	Product 7 Cash & Credit Prices
H (4 bytes)	Product 8 Cash & Credit Prices
I (1 byte)	Product 1 blend percentage
J (1 byte)	Product 2 blend percentage
K (1 byte)	Product 3 blend percentage
L (1 byte)	Product 4 blend percentage
M (1 byte)	Product 5 blend percentage
N (1 byte)	Product 6 blend percentage
O (1 byte)	Product 7 blend percentage
P (1 byte)	Product 8 blend percentage

3.9.6. Box69 response

B0	~B0

Where:
B0 – Prices received

3.9.7. Example controller command

Command	C0	~C0	A3	~A3				
	CASH				CREDIT			
Product 1	12	~12	34	~34	43	~43	21	~21
Product 2	56	~56	78	~78	87	~87	65	~65
Product 3	90	~90	12	~12	21	~21	9	~09
Product 4	12	~12	34	~34	43	~43	21	~21
Product 5	00	~00	00	~00	00	~00	00	~00
Product 6	00	~00	00	~00	00	~00	00	~00
Product 7	00	~00	00	~00	00	~00	00	~00
Product 8	00	~00	00	~00	00	~00	00	~00
	Blend							
Product 1 blend	00	~00						
Product 2 blend	00	~00						
Product 3 blend	00	~00						
Product 4 blend	00	~00						
Product 5 blend	00	~00						
Product 6 blend	00	~00						
Product 7 blend	00	~00						
Product 8 blend	00	~00						

Send cash/cred prices for fuelling point 1 (address 00 Hex) as follows:

	cash	credit
Product 1	1.234	4.321
Product 2	5.678	8.765
Product 3	9.012	2.109
Product 4	1.234	4.321
Product 5	0.000	0.000
Product 6	0.000	0.000
Product 7	0.000	0.000
Product 8	0.000	0.000

Note: The blend values are not taking into consideration.

3.9.8. Example response

B0	~B0

Prices Received.

5. BOX69 FP-CONTROL STATE MACHINE AND TRANSACTION DIAGRAMS

The below table shows how the states of Box69 state-machine change per FP (fuelling point) depending on the connected PUMP/FP events. A particular event is active (cause change) only in the states marked with red. In the other states (left blank in the table) it does not cause state change.

PUMP EVENT / FP STATE	2F	20	A0	90	D0	F0	91	99
Nozzle activated		A0						
Nozzle deactivated			20	20	20/91	20	20	20
Segment Check Complete				D0				
Pulser Check Limit Reached					F0			
Preset Max Reached					91	91		

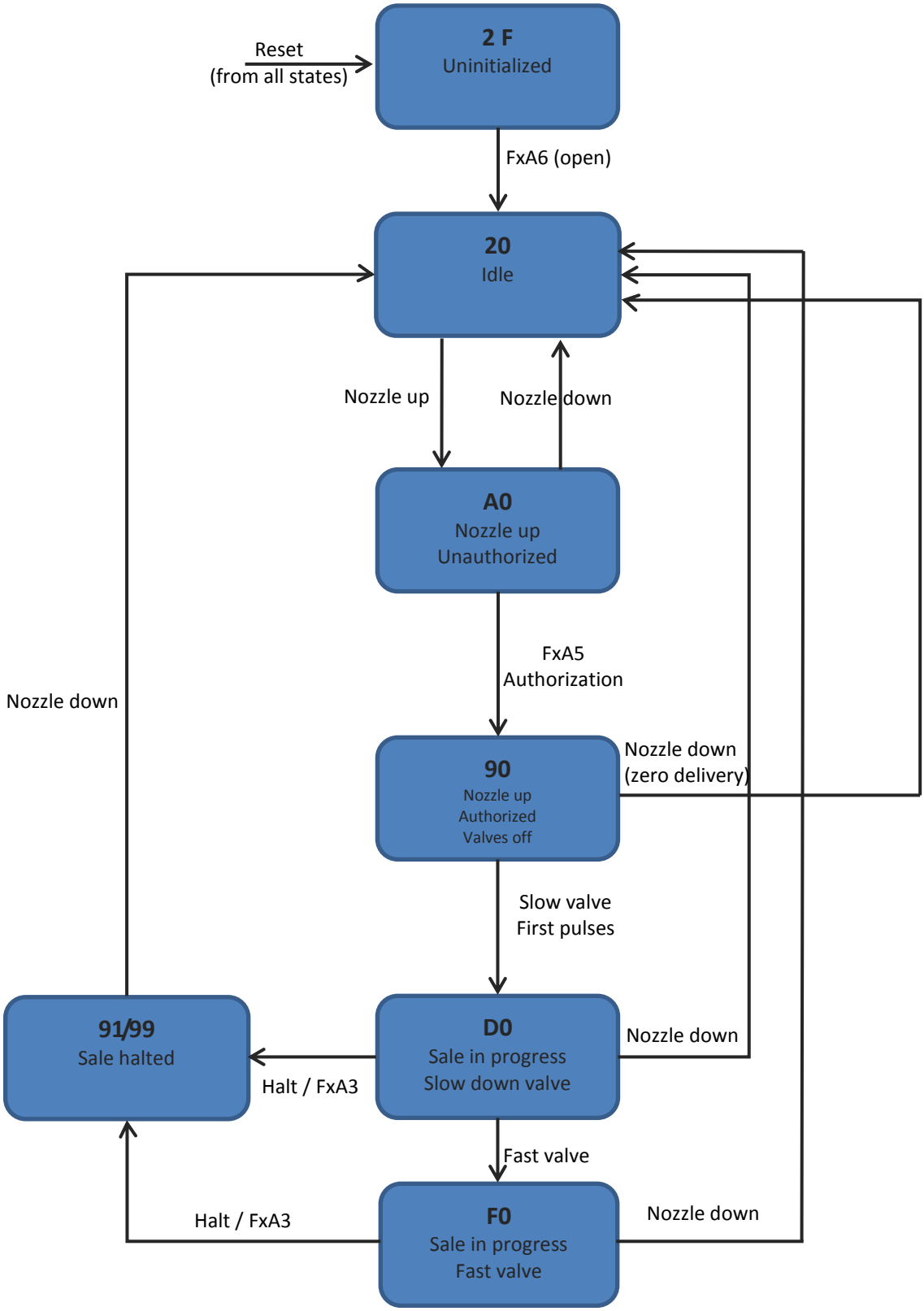
Table 4
FP-states transactions based on the pump events

The below table shows how the Box69 states per FP change depending on the POS/CD commands sent to the Box (respectively to the FP). As shown a particular command causes state change only in the states marked with red. In the states marked in black the command is accepted but doesn't cause state change and in the other states left blank in the table the command is not allowed to be sent for the particular FP.

POS EVENT / FP STATE	2F	20	A0	90	D0	F0	91	99
Req. ID (FxA0)	2F	20	A0	90	D0	F0	91	99
Req. Display Data (FxA1)	2F	20	A0	90	D0	F0	91	99
Req. Status (FxA2)	2F	20	A0	90	D0	F0	91	99
Halt (FxA3)				91/99	91/99	91/99	91/99	
Authorize Sale (FxA5)			90					
Send Main Display Data (FxA6)	20	20	A0					
Request S. A. Ttls (FxA9)	2F	20	A0					
Request Act. Nozzle (CxA1)			A0					
Acknow Deactiv. Nozzle (CxA2)		20						
Send Aux Price Disp Data (CxA3)		20	A0					

Table 5
FP-states transactions based on the commands send by the POS

4.1. BOX69 state-machine per FP



SOME IMPORTANT REMARKS ABOUT THE STATE MACHINE DIAGRAM:

- After power ON the POS-system should start pooling Box69 for status of all of the FP's with command FxA2. It is recommended at least ones in a second each FP to be asked for its state;
- After reset (power-down of Box69 or the real pump power-down) the FP or all of the FPs are in state 2F "uninitialized". If the POS-system retrieves such state from a FP it needs to send first command FxA6 (open dispenser) with the necessary information for the main display for moving to state 20 (idle);
- If the FP reports state A0 (nozzle up) the POS should request first the nozzle (product) number via command CxA1 and then gives authorization for the particular nozzle with command FxA5;
- State transactions from "90 -> 20", "D0 -> 20", "F0 -> 20" or "91/99 -> 20" always mean that the pump generated a transaction which needs to be retrieved by the POS using command FxA1;

6. APPENDIX A – Typical communication log between Box69 and POS system (one FP connected – FP1)

Remark: For simplification and easy reading the binary complement bytes of the 'doubletalk' protocol (see chapter 3) are removed from the below log-listing.

Time Stamp in ms	Direction	UDC Command + Comment

```
/* After power-on the POS is asking frequently for FP's ID – for
detecting which FP(s) are available/connected. For simplifying the
example it pols only FP1.  */

005130 B69<-POS- 0xF0,0xA0,
005746 B69<-POS- 0xF0,0xA0,
.
.
.
010674 B69<-POS- 0xF0,0xA0,
.
.
012524 B69<-POS- 0xF0,0xA0,
012530 B69->POS- 0xE8,  //FP1 answers with its ID thus it is connected;

012934 B69<-POS- 0xF0,0xA2,    // Requesting FP's status
012940 B69->POS- 0x2F,         // Uninitialized (see table 3)

013344 B69<-POS- 0xF0,0xA1,    // Requesting FP's display data (this is
                               //important for making sure that no pending
                               //transaction will be lost)
013367 B69->POS- 0x11,0x11,0x00,0x00,0x00,0x00,0x00,0x00,0x2F,
                               //Amount: 0000.00
                               //Volume: 0000.00
                               //Price:  1.111
                               //FP's Status: 2F

013774 B69<-POS- 0xF0,0xA6,0xAA,0x02,0x02,0x02,0x00,0x00,0x00,0x00,
                               //"Open FP" this a must (see the state
                               // machine diagram).
013777 B69->POS- 0x20,         // The FP turns to IDLE


014164 B69<-POS- 0xF0,0xA9,0x00, // Requesting FP's totals
014355 B69->POS-
0x00,0x54,0x53,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x60,0x48,0x48,0x00,0x0
0,0x00,0x00,0x00,0x00,0x00,0x08,0x33,0x52,0x00,0x00,0x00,0x00,0x00,0x00,0
x00,0x54,0x61,0x54,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x20,0x09,0x45,0x00
,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x
00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,
0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x20,

014861 B69<-POS- 0xC0,0xA2, // Nozzle return acknowledge (not mandatory)
014867 B69->POS- 0xB0,
```

```
015271 B69<-POS- 0xF0,0xA2,    // Requesting FP's status
015277 B69->POS- 0x20,         // The FP is IDLE

024873 B69<-POS-
0xC0,0xA3,0x11,0x11,0x00,0x00,0x22,0x22,0x00,0x00,0x33,0x33,0x00,0x00,0x4
4,0x44,0x00,0x00,0x55,0x55,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0
x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,
// The POS sends the off-line prices (this is used to show the product
// price on each product price display of the dispenser
024877 B69->POS- 0xB0,         // The prices are accepted

029378 B69<-POS- 0xF0,0xA2,    // Requesting FP's status
029384 B69->POS- 0x20,         // The FP is IDLE
029788 B69<-POS- 0xF0,0xA2,
029794 B69->POS- 0x20,
.
.
.
030198 B69<-POS- 0xF0,0xA2,
030204 B69->POS- 0x20,
.
.
030608 B69<-POS- 0xF0,0xA2,    // Requesting FP's status
030614 B69->POS- 0xA0,         // A nozzle is up (pump is calling)

031018 B69<-POS- 0xC0,0xA1,    // Requesting the nozzle/product number
031024 B69->POS- 0x01,         // It's nozzle/product one

031502 B69<-POS- 0xF0,0xA5,0x05,0x11,0x11,0x00,0x99,0x99,0x00,0x90,0x99,
// The POS authorizes FP1 to start delivery with product price 01.111 and
// maximal presets.
031506 B69->POS- 0x90, // The FP is authorized

031531 B69<-POS- 0xF0,0xA1, // Requesting FP's display data
031554 B69->POS- 0x11,0x11,0x00,0x00,0x00,0x00,0x00,0x00,0x90,
031941 B69<-POS- 0xF0,0xA1,
031964 B69->POS- 0x11,0x11,0x00,0x00,0x00,0x00,0x00,0x00,0x90,
.
.
.
039321 B69<-POS- 0xF0,0xA1,
039344 B69->POS- 0x11,0x11,0x20,0x12,0x00,0x10,0x01,0x00,0x90,
039731 B69<-POS- 0xF0,0xA1,
039754 B69->POS- 0x11,0x11,0x20,0x12,0x00,0x10,0x01,0x00,0xD0,
// The fuelling has started
.
040961 B69<-POS- 0xF0,0xA1,
040984 B69->POS- 0x11,0x11,0x20,0x12,0x00,0x10,0x01,0x00,0xD0,
.
041371 B69<-POS- 0xF0,0xA1,
041394 B69->POS- 0x11,0x11,0x60,0x41,0x00,0x74,0x03,0x00,0xD0,
041781 B69<-POS- 0xF0,0xA1,
041804 B69->POS- 0x11,0x11,0x60,0x41,0x00,0x74,0x03,0x00,0xD0,
.
.
.
044651 B69<-POS- 0xF0,0xA1,
```

```
044674 B69->POS- 0x11,0x11,0x20,0x56,0x00,0x06,0x05,0x00,0xD0,
.
.
.
.
045061 B69<-POS- 0xF0,0xA1,
045084 B69->POS- 0x11,0x11,0x20,0x56,0x00,0x06,0x05,0x00,0xD0,
.
.
.
047954 B69->POS- 0x11,0x11,0x20,0x00,0x01,0x02,0x09,0x00,0xD0,
048341 B69<-POS- 0xF0,0xA1,
048364 B69->POS- 0x11,0x11,0x20,0x00,0x01,0x02,0x09,0x00,0xD0,
048751 B69<-POS- 0xF0,0xA1,
048774 B69->POS- 0x11,0x11,0x20,0x00,0x01,0x02,0x09,0x00,0xD0,
049161 B69<-POS- 0xF0,0xA1,
049184 B69->POS- 0x11,0x11,0x20,0x00,0x01,0x02,0x09,0x00,0xD0,
049571 B69<-POS- 0xF0,0xA1,
049594 B69->POS- 0x11,0x11,0x20,0x00,0x01,0x02,0x09,0x00,0xD0,
049981 B69<-POS- 0xF0,0xA1,
050004 B69->POS- 0x11,0x11,0x20,0x00,0x01,0x02,0x09,0x00,0xD0,

050391 B69<-POS- 0xF0,0xA1,
050414 B69->POS- 0x11,0x11,0x20,0x00,0x01,0x02,0x09,0x00,0x91,
// The fuelling is terminated or it is over

050801 B69<-POS- 0xF0,0xA1,
050824 B69->POS- 0x11,0x11,0x90,0x22,0x01,0x06,0x11,0x00,0x91,
.
.
051621 B69<-POS- 0xF0,0xA1,
051644 B69->POS- 0x11,0x11,0x90,0x22,0x01,0x06,0x11,0x00,0x91,

052031 B69<-POS- 0xF0,0xA1,
052054 B69->POS- 0x11,0x11,0x90,0x22,0x01,0x06,0x11,0x00,0x20,
// The FP is back to IDLE – the delivery has finished

052441 B69<-POS- 0xC0,0xA2,
052447 B69->POS- 0xB0,        // Nozzle return acknowledge

052492 B69<-POS- 0xF0,0xA1,   // Requesting FP's display data
                              // This display data is considered as final
                              // transaction data

052515 B69->POS- 0x11,0x11,0x90,0x22,0x01,0x06,0x11,0x00,0x20,
                              // Product price 1.111
                              // Money: 012.290
                              // Volume: 0011.06

052543 B69<-POS- 0xF0,0xA9,0x00,    // Requesting FP's totals (the last
                                    // delivery should be included)
052734 B69->POS-
0x06,0x65,0x53,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x60,0x48,0x48,0x00,0x0
0,0x00,0x00,0x00,0x00,0x00,0x08,0x33,0x52,0x00,0x00,0x00,0x00,0x00,0x00,0
x00,0x54,0x61,0x54,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x20,0x09,0x45,0x00
,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x
```

```
00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,
0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x20,

052736 B69<-POS- 0xF0,0xA2,    // Requesting FP's status
052743 B69->POS- 0x20,         // The FP is IDLE
...
```