INNOVATIVE TECHNOLOGY LTD

CC2

Communications Protocol Manual

GA863

Issue version 1.2.4

INTELIGENCE IN VALIDATION

A.u.S. Spielgeräte GesmbH
Scheydgasse 48 A-1210 Wien
Tel. +43-1-271 66 00 Fax.+43-1-271 66 00 75
E-mail: verkauf@aus.at
http://www.aus.at
Öffnungszeiten: Mo-Fr. 9-18 Uhr

Contents

Version History

Version	Issue date	Notes
1.1.2	14 Jul 2010	• First issue release. • Comms revision to 1.1.2 • Added payout by denomination commands.
1.1.3	27 Oct 2010	• Added bezel mode command.
1.2.0	25 May 2011	• Added multi-currency commands. • Corrected fraud response format.
1.2.1	23 Apr 2012	• Added coin feeder support commands.
1.2.2	12 Jul 2012	• Added SMART Empty command and events. • Added coin inhibits channel commands.
1.2.3	05 Oct 2012	• First new format issue. • Corrected Set Inhibit Peripheral Device Value command. • Added Get Inhibit Peripheral Device Value command. • Added supported cctalk commands. • Added status event Peripheral device disabled. • Deprecation of PIN commands 219,218.
1.2.4	23 Oct 2012	• Added SMART Payout support to Set Payout Options and Get Payout Options. • New status event added in response to Request Status and Request Status (with currency support) - Note in Bezel Hold. • Modification to Set Bezel Mode command. • Corrected response sizes in Request Status command.

Introduction

The CC2 protocol provides commands to pay-out notes/coins from ITL payout devices by multiple possible methods over a cctalk packet system.

This specification documents the commands used by a device to drive SMART Hopper, SMART Payout and SMART Feeder products using ccTalk commands.

This removes the security provided by the eSSP encryption layer and suitable physical security should be employed to protect the ccTalk bus.

This document should be used in conjunction with the generic cctalk specification documents available for download at http://www.cctalk.org

Representations

Values are represented as 32 bit unsigned integer (4 bytes) and in the lowest value of currency. For example 125.65 would be 0x00003115.

When sending or receiving a value the Least significant byte is sent first. So in this example [0x15] [0x31] [0x00] [0x00] will be sent. When using commands with multi-currency support all values will be followed by a 3 byte ASCII Currency code (e.g. 069 085 082 for EUR)

Packet examples are show unencrypted using an 8 bit checksum with host address of 1 and device address of 3.

Pay-out Encryption

The pay-out devices support various levels of encryption. These options are configured on an individual device off-line by PC tools. Options:

- Level 0 No encryption. All packets are plain with 8-bit checksum. No payout commands are encrypted. THIS IS THE LOWEST LEVEL OF SECURITY AND THE USER WOULD NEED TO CONSIDER THE POSSIBILITY OF BUS HI-JACK FRAUDS WITH THIS OPTION.
- Level 1 Type 1 security bytes for payout commands.
- Level 2 Type 2 security bytes for payout commands.
- DES Encryption for payout commands.
  Packet Encryption Options:
- BNV packet encryption.
- All packets are encrypted using Money Controls BNV security algorithm
- Packet checksum
- Option of using 8bit addition or 16bit CRC checksum.

Command Index

Simple Poll	254	Address Poll	253
Address Clash	252	Address Change	251
Address Random	250	Request Polling Priority	249
Request Manufacturer ID	246	Request Equipment Category ID	245
Request Product Code	244	Request Serial Number	242
Request Software Revision	241	Set Note Inhibit Channels	231
Request Note Channel inhibits	230	Set Master Inhibit Status	228
Get Master Inhibit Status	227	Request Data Storage Capability	216
Request Option Flags	213	Request Last Mod Date	195
Request Build Code	192	Request Address Mode	169
Pump RNG	161	Request Cipher Key	160
Read Buffered Bill Events	159	Request Bill id	157
Request Country Scaling Factor	156	Request Bill Position	155
Route Bill	154	Modify Bill Operating Mode	153
Request Bill Operating Mode	152	Request Currency Revision	145
Switch Encryption Code	137	Store encryption code	136
Read Barcode Data	129	Request Encryption Support	111
Switch DES key	110	Request Encrypted Status	109
Get Inhibit Peripheral Device Value	053	Get Cashbox Operation Data	052
Smart Empty	051	Set Inhibit Peripheral Device Value	050
Get Peripheral Device Master Inhibit	049	Set Peripheral Device Master Inhibit	048
Request Status (with currency support)	047	Get Device Setup (with currency support)	046
Float By Denomination (with currency support)	045	Payout by denomination (with currency support)	044
Set Denomination Amount (with currency support)	043	Get Denomination Amount (with currency support)	042
Get Minimum Payout (with currency support)	041	Float Amount (with currency support)	040
Payout Amount (with currency support)	039	Get Routing (with currency support)	038
Set Route (with currency support)	037	Set Bezel Mode	035
Run Unit Calibration	034	Float By Denomination	033
Payout By Denomination	032	Get Payout Options	031
Set Payout Options	030	Request Status	029
Get Device Setup	028	Set Denomination Amount	027
Get Denomination Amount	026	Get Minimum Payout	025
Empty	024	Float Amount	023
Payout Amount	022	Get Routing	021
Set Routing	020	Request Comms Revision	004
Reset Device	001		

SMART HOPPER implementation list

Reset Device	001	Request Comms Revision	004
Set Routing	020	Get Routing	021
Payout Amount	022	Float Amount	023
Empty	024	Get Minimum Payout	025
Get Denomination Amount	026	Set Denomination Amount	027
Get Device Setup	028	Request Status	029
Set Payout Options	030	Get Payout Options	031
Payout By Denomination	032	Float By Denomination	033
Run Unit Calibration	034	Set Route (with currency support)	037
Get Routing (with currency support)	038	Payout Amount (with currency support)	039
Float Amount (with currency support)	040	Get Minimum Payout (with currency support)	041
Get Denomination Amount (with currency support)	042	Set Denomination Amount (with currency support)	043
Payout by denomination (with currency support)	044	Float By Denomination (with currency support)	045
Get Device Setup (with currency support)	046	Request Status (with currency support)	047
Set Peripheral Device Master Inhibit	048	Get Peripheral Device Master Inhibit	049
Set Inhibit Peripheral Device Value	050	Smart Empty	051
Get Cashbox Operation Data	052	Get Inhibit Peripheral Device Value	053
Request Encrypted Status	109	Switch DES key	110
Request Encryption Support	111	Store encryption code	136
Switch Encryption Code	137	Request Cipher Key	160
Pump RNG	161	Request Address Mode	169
Request Build Code	192	Request Last Mod Date	195
Request Data Storage Capability	216	Get Master Inhibit Status	227
Set Master Inhibit Status	228	Request Software Revision	241
Request Serial Number	242	Request Product Code	244
Request Equipment Category ID	245	Request Manufacturer ID	246
Address Poll	253	Simple Poll	254

SMART PAYOUT implementation list

Reset Device	001	Request Comms Revision	004
Set Routing	020	Get Routing	021
Payout Amount	022	Float Amount	023
Empty	024	Get Minimum Payout	025
Get Denomination Amount	026	Get Device Setup	028
Request Status	029	Set Payout Options	030
Get Payout Options	031	Payout By Denomination	032
Float By Denomination	033	Run Unit Calibration	034
Set Bezel Mode	035	Set Route (with currency support)	037
Get Routing (with currency support)	038	Payout Amount (with currency support)	039
Float Amount (with currency support)	040	Get Minimum Payout (with currency support)	041
Get Denomination Amount (with currency support)	042	Payout by denomination (with currency support)	044
Float By Denomination (with currency support)	045	Get Device Setup (with currency support)	046
Request Status (with currency support)	047	Request Encrypted Status	109
Switch DES key	110	Request Encryption Support	111
Read Barcode Data	129	Store encryption code	136
Switch Encryption Code	137	Request Currency Revision	145
Request Bill Operating Mode	152	Modify Bill Operating Mode	153
Route Bill	154	Request Bill Position	155
Request Country Scaling Factor	156	Request Bill id	157
Read Buffered Bill Events	159	Request Cipher Key	160
Pump RNG	161	Request Address Mode	169
Request Build Code	192	Request Last Mod Date	195
Request Option Flags	213	Request Data Storage Capability	216
Get Master Inhibit Status	227	Set Master Inhibit Status	228
Request Note Channel inhibits	230	Set Note Inhibit Channels	231
Request Software Revision	241	Request Serial Number	242
Request Equipment Category ID	245	Address Random	250
Address Change	251	Address Clash	252
Address Poll	253	Simple Poll	254

SMART SYSTEM implementation list

Reset Device	001	Request Comms Revision	004
Set Routing	020	Get Routing	021
Payout Amount	022	Float Amount	023
Empty	024	Get Minimum Payout	025
Get Denomination Amount	026	Set Denomination Amount	027
Get Device Setup	028	Request Status	029
Set Payout Options	030	Get Payout Options	031
Payout By Denomination	032	Float By Denomination	033
Run Unit Calibration	034	Set Route (with currency support)	037
Get Routing (with currency support)	038	Payout Amount (with currency support)	039
Float Amount (with currency support)	040	Get Minimum Payout (with currency support)	041
Get Denomination Amount (with currency support)	042	Set Denomination Amount (with currency support)	043
Payout by denomination (with currency support)	044	Float By Denomination (with currency support)	045
Get Device Setup (with currency support)	046	Request Status (with currency support)	047
Set Peripheral Device Master Inhibit	048	Get Peripheral Device Master Inhibit	049
Set Inhibit Peripheral Device Value	050	Smart Empty	051
Get Cashbox Operation Data	052	Get Inhibit Peripheral Device Value	053
Request Encrypted Status	109	Switch DES key	110
Request Encryption Support	111	Store encryption code	136
Switch Encryption Code	137	Request Cipher Key	160
Pump RNG	161	Request Address Mode	169
Request Build Code	192	Request Last Mod Date	195
Request Data Storage Capability	216	Get Master Inhibit Status	227
Set Master Inhibit Status	228	Request Software Revision	241
Request Serial Number	242	Request Product Code	244
Request Equipment Category ID	245	Request Manufacturer ID	246
Address Poll	253	Simple Poll	254

CC2 Command List

Complete listing of all CC2 commands

back to index

Command name	Code dec	Code hex
Simple Poll	254	0xFE

Supported on devices:

SMART Hopper	SMART Payout	SMART System

Description

Command to check the correct operation of communication and to confirm the presence in the bus of a device. If no reply is received to the request sent (Reception timeout in Machine), it will indicate that the device is faulty or not connected. All the cctalk peripherals must respond to a Simple Poll, regardless of the cctalk communication protocol level that has been implemented.

Parameters

This command has no parameters

Command packet example:

dec

003	000	001	254	254

hex

03	00	01	FE	FE

Response

Returns ACK

dec

001	000	003	000	252

hex

01	00	03	00	FC

back to index

Command name	Code dec	Code hex
Address Poll	**253**	**0xFD**

Supported on devices:

SMART Hopper	SMART Payout	SMART System

Description

The host sends this command as a broadcast address packet (0 destination address). The device responses with a single byte containing its address with a series of delays: Disable rx port Delay (4 * addr) ms Send [addr] Delay 1200 - (4 * addr) ms Enable rx port

Parameters

This command has no parameters

Command packet example:

dec

003	000	001	253	255

hex

03	00	01	FD	FF

Command name	Code dec	Code hex
Address Clash	**252**	**0xFC**

back to index

Supported on devices:

SMART Payout

Description

This command is transmitted to a specified address. It attempts to determine if one or more devices share the same address. The device returns a single byte of address data after a random delay: Slave Response Algorithm r = rand(256) // random value in the range 0 to 255 Disable rx port Delay (4 * r) ms Send [addr] Delay 1200 - (4 * r) ms Enable rx port

Parameters

This command has no parameters

Command packet example:

dec

003	000	001	252	000

hex

03	00	01	FC	00

back to index

Command name	Code dec	Code hex
Address Change	**251**	**0xFB**

Supported on devices:

SMART Payout

Description

This command allows the addressed device to have its address changed for subsequent commands. The host sends 1 data byte, the value of which is the new address. It is a good idea to make sure that 2 devices do not share the same address before sending this command. A full ACK message is returned. Note the ACK is sent back from the original address, not the changed address. In other words, the change to the ccTalk address field is done after the ACK is returned rather than before.

This example is a request to change address of device address to address 4.

dec

003	001	001	251	004	252

hex

03	01	01	FB	04	FC

Response

The device responds with its original address.

dec

001	000	003	000	252

hex

01	00	03	00	FC

Command name	Code dec	Code hex
Address Random	**250**	**0xFA**

back to index

Supported on devices:

SMART Payout

Description

This command allows the addressed device to have its address changed to a random value. This is the escape route when you find out that one or more devices share the same address. A full ACK message is returned.

Parameters

This command has no parameters

Command packet example:

dec

003	000	001	250	002

hex

03	00	01	FA	02

Command name	Code dec	Code hex
Request Polling Priority	**249**	**0xF9**

back to index

Supported on devices:

.

Description

This is an indication by a device of the recommended polling interval for buffered credit information. Polling a device at an interval longer than this may result in lost credits. [units] 0 - special case, see below 1 - ms 2 - x10 ms 3 - seconds 4 - minutes 5 - hours 6 - days 7 - weeks 8 - months 9 - years

Parameters

This command has no parameters

Command packet example:

dec

003	000	001	249	003

hex

03	00	01	F9	03

Response

The response is 2 bytes. See the description above for explanation. In this example, the device has a polling priority of 200ms.

dec

001	002	003	000	002	020	228

hex

01	02	03	00	02	14	E4

back to index

Command name	Code dec	Code hex
Request Manufacturer ID	**246**	**0xF6**

Supported on devices:

SMART Hopper	SMART System

Description

This command returns the ASCII code for the manufacturer of the device. In this case ITL

Parameters

This command has no parameters

Command packet example:

dec

003	000	001	246	006

hex

03	00	01	F6	06

Response

Example response showing ITL code

dec

001	003	003	000	073	084	076	016

hex

01	03	03	00	49	54	4C	10

ascii

.	.	.	.	I	T	L	.

Command name	Code dec	Code hex
Request Equipment Category ID	**245**	**0xF5**

back to index

Supported on devices:

SMART Hopper	SMART Payout	SMART System

Description

Returns the type of connected device as ascii array.

Parameters

This command has no parameters

Command packet example:

dec

003	000	001	245	007

hex

03	00	01	F5	07

Response

example showing data response for SMART_HOPPER.

dec

001	012	003	000	083	077	065	082	084	095	072	079	080	080	069	082	060

hex

01	0C	03	00	53	4D	41	52	54	5F	48	4F	50	50	45	52	3C

ascii

.	.	.	.	S	M	A	R	T	_	H	O	P	P	E	R	

Command name	Code dec	Code hex
Request Product Code	**244**	**0xF4**

back to index

Supported on devices:

SMART Hopper	SMART System

Description

This command returns the device product code. The complete identification of the product can be determined by the use of the [Request product code] command followed by the [Request build code] command.

Parameters

This command has no parameters

Command packet example:

dec

003	000	001	244	008

hex

03	00	01	F4	08

Response

Example response showing SMART Hopper code SH3.

dec

001	003	003	000	083	072	051	043

hex

01	03	03	00	53	48	33	2B

ascii

.	.	.	.	S	H	3	.

Command name	Code dec	Code hex
Request Serial Number	**242**	**0xF2**

back to index

Supported on devices:

SMART Hopper	SMART Payout	SMART System

Description

In reply to this command, the Device sends the serial number of the device in a 3-byte code.

Parameters

This command has no parameters

Command packet example:

dec

003	000	001	242	010

hex

03	00	01	F2	0A

Response

Example showing serial number return from a device with code 4321432

dec

001	003	003	000	152	240	065	048

hex

01	03	03	00	98	F0	41	30

Command name	Code dec	Code hex
Request Software Revision	**241**	**0xF1**

back to index

Supported on devices:

SMART Hopper	SMART Payout	SMART System

Description

This command will return the full software revision of the device as an ASCII string.

Parameters

This command has no parameters

Command packet example:

dec

003	000	001	241	011

hex

03	00	01	F1	0B

Response

Example showing software revision of NV00093333453000

dec

001	016	003	000	078	086	048	048	048	057	051	051	051	051	052	053	051	048	048	048	135

hex

01	10	03	00	4E	56	30	30	30	39	33	33	33	33	34	35	33	30	30	30	87

ascii

.	.	.	.	N	V	0	0	0	9	3	3	3	3	4	5	3	0	0	0	.

Command name	Code dec	Code hex
Set Note Inhibit Channels	**231**	**0xE7**

back to index

Supported on devices:

SMART Payout

Description

This command sets the inhibit status of each of the 16 available channels on a bill acceptor device.

Command has two byte bit field. Each bit represents a bill channel. bit 0 = channel1 to bit 15 = channel 16. Set to 0 to inhibt channel, 1 to enable channel. This example shows a command set to enable channels 1,2 and 3 only.

dec

003	002	001	231	007	000	012

hex

03	02	01	E7	07	00	0C

Response

ACK response.

dec

001	000	003	000	252

hex

01	00	03	00	FC

Command name	Code dec	Code hex
Request Note Channel inhibits	230	0xE6

back to index

Supported on devices:

SMART Payout

Description

This command returns the status of the two byte inhibit register for the bill channels of a note accepting device.

Parameters

This command has no parameters

Command packet example:

dec

003	000	001	230	022

hex

03	00	01	E6	16

Response

Response returns the two byte inihibit register. bit 0 = channel 1 to bit 15 = channel 16. This example shows a register setup of channel 2,3 and 4 enabled and all others inhibited.

dec

001	002	003	000	014	000	236

hex

01	02	03	00	0E	00	EC

Command name	Code dec	Code hex
Set Master Inhibit Status	**228**	**0xE4**

back to index

Supported on devices:

SMART Hopper	SMART Payout	SMART System

Description

A command to globally enable or disable the payout device or bill validator for payout/paying in operations. This value is stored in volatile ram and will be set to disabled state after a reset.

Command has 1 data byte which is a bit field. Bit 0 controls the Master inhibit state (0 = disable, 1 = enable) Bits 1-7 are not used. The example shows a command to set the master inhibit state to enable.

dec

003	001	001	228	001	022

hex

03	01	01	E4	01	16

Response

ACK response.

dec

001	000	003	000	252

hex

01	00	03	00	FC

back to index

Command name	Code dec	Code hex
Get Master Inhibit Status	**227**	**0xE3**

Supported on devices:

SMART Hopper	SMART Payout	SMART System

Description

This command returns the current status of the Master inhibit value from the device.

Parameters

This command has no parameters

Command packet example:

dec

003	000	001	227	025

hex

03	00	01	E3	19

Response

Response contains the master inhibit status register byte. Bit 0 gives the status: 0 is disabled, 1 is enabled. Bits 1 to 7 are not used. This response shows the master inhibit set disabled.

dec

001	001	003	000	000	251

hex

01	01	03	00	00	FB

Command name	Code dec	Code hex
Request Data Storage Capability	**216**	**0xD8**

back to index

Supported on devices:

SMART Hopper	SMART Payout	SMART System

Description

A command to return the data storage capability of the device. This command is included for system compatibility. ITL device products return all 0 for this.

Parameters

This command has no parameters

Command packet example:

dec

003	000	001	216	036

hex

03	00	01	D8	24

Response

This command always returns 5 zero bytes for ITL products

dec

001	005	003	000	000	000	000	000	000	247

hex

01	05	03	00	00	00	00	00	00	F7

Command name	Code dec	Code hex
Request Option Flags	213	0xD5

back to index

Supported on devices:

SMART Payout

Description

This command returns a one byte bit field register formatted as: 0 stacker, 1 escrow, 2 individual bill accept counters, 3 individual error counters, 4 non-volatile counters, 5 bill teach facility, 6 bill security tuning, 7 remote bill programming. If the bit is set (1) the option is supported.

Parameters

This command has no parameters

Command packet example:

dec

003	000	001	213	039

hex

03	00	01	D5	27

Response

This example shows a response of a device supporting stacker and escrow functionality.

dec

001	001	003	000	003	248

hex

01	01	03	00	03	F8

Command name	Code dec	Code hex
Request Last Mod Date	**195**	**0xC3**

back to index

Supported on devices:

SMART Hopper	SMART Payout	SMART System

Description

Returns an encoded date array showing the last modification of this device.

Parameters

This command has no parameters

Command packet example:

dec

003	000	001	195	057

hex

03	00	01	C3	39

Response

Example showing date 1st Jan 1 year after base year. Base year for ITL SMART products is 2009

dec

001	002	003	000	017	001	232

hex

01	02	03	00	11	01	E8

Command name	Code dec	Code hex
Request Build Code	**192**	**0xC0**

back to index

Supported on devices:

SMART Hopper	SMART Payout	SMART System

Description

This command will return an ASCII array describing the build version of the device.

Parameters

This command has no parameters

Command packet example:

dec

003	000	001	192	060

hex

03	00	01	C0	3C

Response

Example response showing build code: standard

dec

001	008	003	000	115	116	097	110	100	097	114	100	163

hex

01	08	03	00	73	74	61	6E	64	61	72	64	A3

ascii

.	.	.	.	s	t	a	n	d	a	r	d	.

Command name	Code dec	Code hex
Request Address Mode	**169**	**0xA9**

back to index

Supported on devices:

SMART Hopper	SMART Payout	SMART System

Description

This command returns the mode in which the cctalk address is stored and if it can be changed by serial command. Returns a bit register configured as: B0 - Address is stored in ROM B1 - Address is stored in RAM B2 - Address is stored in EEPROM or NV memory B3 - Address selection via interface connector B4 - Address selection via PCB links B5 - Address selection via switch B6 - Address may be changed with serial commands (volatile) B7 - Address may be changed with serial commands (non-volatile)

Parameters

This command has no parameters

Command packet example:

dec

003	000	001	169	083

hex

03	00	01	A9	53

Response

Example response showing address is stored in EEPROM but not changeable by serial command.

dec

001	001	003	000	004	247

hex

01	01	03	00	04	F7

Command name	Code dec	Code hex
Pump RNG	**161**	**0xA1**

back to index

Supported on devices:

SMART Hopper	SMART Payout	SMART System

Description

This command is part of the security level payouts. No further details are given here.

Command name	Code dec	Code hex
Request Cipher Key	**160**	**0xA0**

back to index

Supported on devices:

SMART Hopper	SMART Payout	SMART System

Description

This command is used in security level payouts. No further details are given here.

Command name	Code dec	Code hex
Read Buffered Bill Events	**159**	**0x9F**

back to index

Supported on devices:

SMART Payout

Description

This command gives of the last 5 bill events on a bill accepting device. The return data is formatted as 11 bytes... byte 0 is an event counter. This is incremented on each event. The event codes are 5 two byte pairs, Byte A and Byte B. The codes are explained in the table below.

Parameters

This command has no parameters

Command packet example:

dec

003	000	001	159	093

hex

03	00	01	9F	5D

Response

This example shows 2 events returned. 2,1 is a valid bill in channel 2 held in escrow, 0,3 is note rejected due to transport problem.

dec

001	011	003	000	002	000	003	002	001	000	000	000	000	000	000	233

hex

01	0B	03	00	02	00	03	02	01	00	00	00	00	00	00	E9

For CC2 protocol SMART Payout, this command always returns zeros.
This is done for compatibility with some host systems. The host should use the Get Status (029) command for CC2 events.

dec

001	011	003	000	000	000	000	000	000	000	000	000	000	000	000	241

hex

01	0B	03	00	00	00	00	00	00	00	00	00	00	00	00	F1

Byte A	Byte B	Event	Type
1 to 255	0	Bill type 1 to 255 validated correctly and	Credit
1 to 255	1	Bill type 1 to 255 validated correctly and	Pending Credit
0	0	Master inhibit active	Status
0	1	Bill returned from escrow	Status
0	2	Invalid bill (due to validation fail)	Reject
0	3	Invalid bill (due to transport problem)	Reject
0	4	Inhibited bill (on serial)	Status
0	5	Inhibited bill (on DIP switches)	Status
0	6	Bill jammed in transport (unsafe mode)	Fatal Error
0	7	Bill jammed in stacker	Fatal Error
0	8	Bill pulled backwards	Fraud Attempt
0	9	Bill tamper	Fraud Attempt
0	10	Stacker OK	Status
0	11	Stacker removed	Status
0	12	Stacker inserted	Status
0	13	Stacker faulty	Fatal Error
0	14	Stacker full	Status
0	15	Stacker jammed	Fatal Error
0	16	Bill jammed in transport (safe mode)	Fatal Error
0	17	Opto fraud detected	Fraud Attempt
0	18	String fraud detected	Fraud Attempt
0	19	Anti-string mechanism faulty	Fatal Error
0	20	Barcode detected	Status

Command name	Code dec	Code hex
Request Bill id	**157**	**0x9D**

back to index

Supported on devices:

SMART Payout

Description

This command will return the ascii bill data for a requested channel. The return data is formatted as a 7 character identification codeâ ¦ [C] [C] [V] [V] [V] [V] [I] CC = Standard 2 letter country code e.g. EU for the euro. VVVV = Bill value in terms of the country scaling factor I = Issue code. Starts at A and progresses B, C, D, Eâ ¦

The command takes 1 byte data which represents the bill channel of the ID required. In this example we require the id data of channel 2.

dec

003	001	001	157	001	093

hex

03	01	01	9D	01	5D

Response

7 acsii bytes are returned, formatted as described above. In this example channel 2 contains EUR 10 bill. The ascii string returned is EU0010A

dec

001	007	003	000	069	085	048	048	049	048	065	089

hex

01	07	03	00	45	55	30	30	31	30	41	59

ascii

.	.	.	.	E	U	0	0	1	0	A	Y

If the channel is not supported, all zeros are returned.

dec

001	007	003	000	048	048	048	048	048	048	048	165

hex

01	07	03	00	30	30	30	30	30	30	30	A5

ascii

.	.	.	.	0	0	0	0	0	0	0	.

Command name	Code dec	Code hex
Request Country Scaling Factor	**156**	**0x9C**

back to index

Supported on devices:

SMART Payout

Description

A request to return the scaling factor and decimal place position for the given country code.

The command data contains two byte ascii country code. In this example EU is 085,069.

dec

003	001	001	156	085	010

hex

03	01	01	9C	55	0A

Response

For a supported country code then 3 value bytes are returned. bytes 0 and 1 are the value multiplier (scaling factor). Byte 2 is the number of decimal places. In this example scaling factor = 100, decimal places = 2. So a channel value of 100 would be a real value of 100 * 100/100 = 100.00

dec

001	003	003	000	100	000	002	147

hex

01	03	03	00	64	00	02	93

If the country code is not supported, then all zeros are returned.

dec

001	003	003	000	000	000	000	249

hex

01	03	03	00	00	00	00	F9

Command name	Code dec	Code hex
Request Bill Position	**155**	**0x9B**

back to index

Supported on devices:

SMART Payout

Description

Use this command for obtaining the inhibit mask position of a given currency code. Two data bytes are returned.

This command data is 2 byte ascii code of the country required. In this example EU (euro).

dec

003	002	001	155	069	085	197

hex

03	02	01	9B	45	55	C5

Response

The inhibit mask based on the command currency code is returned. In this example, a validator has euro currency on channel1 1,2 and 3 only.

dec

001	002	003	000	007	000	243

hex

01	02	03	00	07	00	F3

Command name	Code dec	Code hex
Route Bill	**154**	**0x9A**

back to index

Supported on devices:

SMART Payout

Description

The host command to decide a destination for the bill held in escrow.

Command has 1 data byte containing a route code. 0 = Return escrow bill, 1= send to stack, 255 = extend escrow hold time. This example is a command to return an escrow bill.

dec

003	001	001	154	000	097

hex

03	01	01	9A	00	61

Response

ACK response.

dec

001	000	003	000	252

hex

01	00	03	00	FC

Command name	Code dec	Code hex
Modify Bill Operating Mode	**153**	**0x99**

back to index

Supported on devices:

SMART Payout

Description

Host option to enable/disable escrow mode in bill validators.

Command data byte is bit field. Bit 0 is not used. Bit 1 is 0 to disable escrow, 1 to enable. This example shows command to set escrow on the device.

dec

003	001	001	153	002	096

hex

03	01	01	99	02	60

Response

ACK response for success.

dec

001	000	003	000	252

hex

01	00	03	00	FC

Command name	Code dec	Code hex
Request Bill Operating Mode	**152**	**0x98**

back to index

Supported on devices:

SMART Payout

Description

This command returns the status of the bill operating mode register. It is configured as bit field. bit 0 is not used, bit 1 is the escrow function. If the bit is set, the function is used, 0 = not used.

Parameters

This command has no parameters

Command packet example:

dec

003	000	001	152	100

hex

03	00	01	98	64

Response

In this example escrow mode is set on the device.

dec

001	001	003	000	002	249

hex

01	01	03	00	02	F9

back to index

Command name	Code dec	Code hex
Request Currency Revision	**145**	**0x91**

Supported on devices:

SMART Payout

Description

This command returns an 8 byte ascii array identifying the dataset version on the device.

Parameters

This command has no parameters

Command packet example:

dec

003	000	001	145	107

hex

03	00	01	91	6B

Response

In this example the ascii for EUR02604 dataset on an NV200 is returned.

dec

001	008	003	000	069	085	082	048	050	054	048	052	012

hex

01	08	03	00	45	55	52	30	32	36	30	34	0C

ascii

.	.	.	.	E	U	R	0	2	6	0	4	.

Command name	Code dec	Code hex
Switch Encryption Code	**137**	**0x89**

back to index

Supported on devices:

SMART Hopper	SMART Payout	SMART System

Description

The host can change the current BNV encryption key to a new value using this command. The command is encrypted with the old key and the response by the device encrypted with the old key. The new key will then take effect but will only persist in memory until it is either changed again or the device is reset. Use the Store Encryption Code command to persistently store this new code.

Example to set new key to 987654. Data bytes are packed so byte 2 = 98h (84 dec), byte 1 = 76h (118 dec), byte 0 = 54h (154 dec)

dec

003	003	001	137	084	118	152	014

hex

03	03	01	89	54	76	98	0E

Response

ACK for successful change.

dec

001	000	003	000	252

hex

01	00	03	00	FC

back to index

Command name	Code dec	Code hex
Store encryption code	**136**	**0x88**

Supported on devices:

SMART Hopper	SMART Payout	SMART System

Description

This command stores the current BNV encryption key into non-volatile memory.

Parameters

This command has no parameters

Command packet example:

dec

003	000	001	136	116

hex

03	00	01	88	74

Response

ACK response for success.

dec

001	000	003	000	252

hex

01	00	03	00	FC

Command name	Code dec	Code hex
Read Barcode Data	**129**	**0x81**

back to index

Supported on devices:

SMART Payout

Description

The host command to read data about a barcode that has been in escrow or stacked. If no barcode has been read, the response will be an ACK without any data bytes. Barcode Data is valid until a new note insertion is detected.

Parameters

This command has no parameters

Command packet example:

dec

003	000	001	129	123

hex

03	00	01	81	7B

Response

ACK with no data. No valid barcode data is stored.

dec

001	000	003	000	252

hex

01	00	03	00	FC

An example response of 16 digit barcode data 1234567812345678.

dec

001	016	003	000	049	050	051	052	053	054	055	056	049	050	051	052	053	054	055	056	164

hex

01	10	03	00	31	32	33	34	35	36	37	38	31	32	33	34	35	36	37	38	A4

ascii

.	.	.	.	1	2	3	4	5	6	7	8	1	2	3	4	5	6	7	8	.

back to index

Command name	Code dec	Code hex
Request Encryption Support	**111**	**0x6F**

Supported on devices:

SMART Hopper	SMART Payout	SMART System

Description

This command will return data about the device encryption configuration to allow host machine installation and configuration. This command will respond even if the device has BNV encryption set and the command is sent unencrypted. The commands data bytes act as a validation signature. Response Data: [Protocol level] [Command level] [Protocol key size] [Command key size] [Command block size] [Trusted mode] [BNV2 | BNV1] [BNV4 | BNV3] [BNV6 | BNV5] [DES1] [DES2] [DES3] [DES4] [DES5] [DES6] [DES7] [DES8]

The validation data (0xAA 0x55 0x00 0x00 0x55 0xAA) bytes are sent to ensure that the command can not be confused if sent as an encrypted packet from another command.

dec

003	006	001	111	170	085	000	000	085	170	137

hex

03	06	01	6F	AA	55	00	00	55	AA	89

Command name	Code dec	Code hex
Switch DES key	**110**	**0x6E**

back to index

Supported on devices:

SMART Hopper	SMART Payout	SMART System

Description

This command allows the host to change the current DES key. The old key and new key are interleaved and the two blocks of 8 byte data are encrypted with the old key. The device then decrypts the data and checks that the old keys match. If so then the swap is made an ACK is sent to the host and the new key stored in persistent memory. If the keys do not match then no reply is sent. This command can also be sent for key verification. If the host sets the new and old keys to the same value then the device will reply with an ACK if the key is correct or no reply if the key sent is not correct.

Command data format

Byte 0	Byte 1	Byte 2	Byte 3	Byte 4	Byte 5	Byte 6	Byte 7	Byte 8	Byte 9	Byte 10	Byte 11	Byte 12	Byte 13	Byte 14	Byte 15
old 1	new 1	old 2	new 2	old 3	new 3	old 4	new 4	old 5	new 5	old 6	new 6	old 7	new 7	old 8	new 8

Response

ACK response for successfull change

dec

001	000	003	000	252

hex

01	00	03	00	FC

Command name	Code dec	Code hex
Request Encrypted Status	**109**	**0x6D**

back to index

Supported on devices:

SMART Hopper	SMART Payout	SMART System

Description

To guard against ACK/NACK hijack frauds â ¨ this command may be used to determine the status of the device. The command may be sent after a payout or if a NACK was received to a payout command to verify the status of the device. The host send 3 challenge bytes, which are then embedded in the reply from the device to verify correct peripheral response.

Commad data contains 3 bytes of host generated data.

dec

003	003	001	109	023	074	212	087

hex

03	03	01	6D	17	4A	D4	57

Response

Host Response data 16 bytes (DES encrypted): [CRC Low][Challenge 1][Event count] [Last payout Request 0][Last payout Request 1][Last payout Request 2] [Last payout Request 3] [Last payout Amount 0][Last Amount Request 1][Last Amount Request 2] [Last payout Amount 3] [Challenge 2][Random 1][Random 2] [Challenge 3][CRC High]

back to index

Command name	Code dec	Code hex
Get Inhibit Peripheral Device Value	**053**	**0x35**

Supported on devices:

SMART Hopper	SMART System

Description

This command returns the current status of the Master Inhibit value from the device indicated by the peripheral code.

Command has 8 data bytes. Byte 0 = periferal code, 0 = coin mech, 1 = coin feeder. Bytes 1-4 are the coin value, bytes 5-7 are the acsii country code. This example is a command to return the state of a coin feeder coin channel value 0.20 EUR.

dec

003	008	001	053	001	020	000	000	000	069	085	082	190

hex

03	08	01	35	01	14	00	00	00	45	55	52	BE

Response

ACK response with 1 data byte for success. Data byte shows state of coin inhibit 0 = disabled, 1 = enabled.

dec

001	000	003	000	252

hex

01	00	03	00	FC

NACK with fail code for unsuccessful command. This example shows failure for currency mis-match.

dec

001	001	003	005	002	244

hex

01	01	03	05	02	F4

fail code	reason
no return	Command not implemented

0	No device detected
1	Device out of service
2	Device currency miss-match

back to index

Command name	Code dec	Code hex
Get Cashbox Operation Data	**052**	**0x34**

Supported on devices:

SMART Hopper	SMART System

Description

This command allows the host to obtain individual levels of each coin denomination emptied to the cashbox after a Smart empty operation.

Parameters

This command has no parameters

Command packet example:

dec

003	000	001	052	200

hex

03	00	01	34	C8

Response

The device responds with an ACK and an array of data: byte 0 - number of denominations, then 9 bytes for each of the denominations giving 2 bytes for the level of coins emptied, 4 bytes for the coin value and 3 bytes for the ascii country code. There the follows 4 bytes giving a count of unknown coins emptied. This example shows that a hopper with 3 denominations of EUR coins was emptied. There were 10 x 0.20 cent coins, 50 x 1.00 coins and 20 x 2.00 coins. 5 unknown coins were also emptied.

dec

001	028	003	000	003	010	000	020	000	000	000	069	085	082	050	000	100	000	000	000	069	085
082	020	000	200	000	000	000	069	085	082	137											

hex

01	1C	03	00	03	0A	00	14	00	00	00	45	55	52	32	00	64	00	00	00	45	55
52	14	00	C8	00	00	00	45	55	52	89											

back to index

Command name	Code dec	Code hex
Smart Empty	**051**	**0x33**

Supported on devices:

SMART Hopper	SMART System

Description

This command will generate events during the emptying process and give values of the current amount emptied in response to Get Status commands. This is encrypted level command and the example is shown as at a unencrypted level. Details of the encryption levels are not shown here.

Parameters

This command has no parameters

Command packet example:

dec

003	000	001	051	201

hex

03	00	01	33	C9

Response

An ACK response is given for a successful request to start SMART emptying.

dec

001	000	003	000	252

hex

01	00	03	00	FC

Command name	Code dec	Code hex
Set Inhibit Peripheral Device Value	**050**	**0x32**

back to index

Supported on devices:

SMART Hopper	SMART System

Description

A command to enable or disable a coin value for the attached coin paying-in mechanism or coin feeder. The peripheral device type is address by the device code byte. If the command is not successfully executed, then a NACK will be returned with one data byte giving the reason failure. This value is stored in volatile ram and will be set to disabled state after a reset.

Command has 9 data bytes: Byte 0 - Peripheral code, 0 = coin mech, 1= coin feeder. Byte 1 enable state, 1 = enable, 0 = disable. Bytes 2-5 - 4 byte coin value. Bytes 6-8 the ascii country code. This example is to enable coin value EUR 0.10 on a coin feeder unit.

dec

003	009	001	050	001	001	010	000	000	000	069	085	082	201

hex

03	09	01	32	01	01	0A	00	00	00	45	55	52	C9

Response

ACK for success.

dec

001	000	003	000	252

hex

01	00	03	00	FC

NACK with fail code for failure. Example shows fail for device not detected

dec

001	001	003	005	000	246

hex

01	01	03	05	00	F6

fail code	reason

no return	Command not implemented
0	No device detected
1	Device out of service
2	Device currency miss-match

Command name	Code dec	Code hex
Get Peripheral Device Master Inhibit	**049**	**0x31**

back to index

Supported on devices:

SMART Hopper	SMART System

Description

This command returns the master inhibit status of a connected peripheral to the Smart Hopper.

The host sends 1 data parameter. 0 = coin mech, 1 = coin feeder. This example shows a request for the master inhibit status of the coin feeder.

dec

003	001	001	049	001	201

hex

03	01	01	31	01	C9

Response

The device response ACK and a data byte giving the status of the master inhibit. In this case 1 for enabled.

dec

001	001	003	000	001	250

hex

01	01	03	00	01	FA

For an error response, the device gives a error data byte with NAK. I this case, the currency of the coin mech and hopper do not match.

dec

001	001	003	005	002	244

hex

01	01	03	05	02	F4

fail code	reason
no return	Command not implemented

0	No device detected
1	Device out of service
2	Device currency miss-match

back to index

Command name	Code dec	Code hex
Set Peripheral Device Master Inhibit	**048**	**0x30**

Supported on devices:

SMART Hopper	SMART System

Description

A command to globally enable or disable the attached coin paying-in mechanism or coin feeder operations. The peripheral device type is address by the device code byte. If the command is not successfully executed, then a NACK will be returned with one data byte giving the reason failure. This value is stored in volatile ram and will be set to disabled state after a reset.

Command has two data bytes. Data 0 is the peripheral code, 0 = coin mech, 1 = coin feeder. Data 1 is the inhibit command. It is a bit field, bit 0 - inhibit status, 0 = device inhibited, 1 = device enabled. Bits 1 - 7 are not used. Example shows master inhibit set to enable device on a coin feeder unit.

dec

003	002	001	048	001	001	200

hex

03	02	01	30	01	01	C8

Response

ACK response for success.

dec

001	000	003	000	252

hex

01	00	03	00	FC

NAK response with error code for failure. Example shows master inhibit set fail on coin feeder due to device out of service.

dec

001	001	003	005	001	245

hex

01	01	03	05	01	F5

fail code	reason
no return	Command not implemented
0	No device detected
1	Device out of service
2	Device currency miss-match

Command name	Code dec	Code hex
Request Status (with currency support)	**047**	**0x2F**

back to index

Supported on devices:

SMART Hopper	SMART Payout	SMART System

Description

This command will return the status of the device and the progress of the current requested operation. After issuing any action commands this command should be used to track the status. The possible status codes are shown in the table.

Parameters

This command has no parameters

Command packet example:

dec

003	000	001	047	205

hex

03	00	01	2F	CD

Response

Example response data for Dispensing event value EUR 5.30. The full event list is shown in the table below.

dec

001	009	003	000	001	001	018	002	000	000	069	085	082	241

hex

01	09	03	00	01	01	12	02	00	00	45	55	52	F1

event	code (hex)	format	size (bytes)
Idle	00		1
Dispensing	01	currency data (see table)	variable
Dispensed	02	currency data (see table)	variable
Coins Low	03		1

Empty	04		1
Jammed	05	currency data (see table)	variable
Halted	06	currency data (see table)	variable
Floating	07	currency data (see table)	variable
Floated	08	currency data (see table)	variable
Timeout	09	currency data (see table)	variable
Incomplete payout	0A	incomplete payout data (see table)	variable
Incomplete float	0B	incomplete payout data (see table)	variable
Cashbox paid	0C	currency data (see table)	variable
Coin credit	0D	4 bytes giving value + 3 bytes country code of coin added by attached mech.	8
Emptying	0E		1
Emptied	0F		1
Fraud attempt	10	currency data (see table)	variable
Disabled	11		1
Note stored	12		1
Slave reset	13		1
Note read	14	4 bytes giving the value of the note + 3 bytes country code	8
Note credit	15	4 byte value giving the value of the note credited + 3 bytes country code	8
Note rejecting	16		1
Note rejected	17		1
Note stacking	18		1
Note stacked	19		1
Note path jam	1A		1
Note stack jam	1B		1
Note from front at start	1C	4 byte value of note rejected + 3 byte country code (0 if unknown)	8
Note stacked at start	1D	4 byte value of note stacked + 3 byte country code (0 if unknown)	8
Cashbox full	1E		1
Cashbox removed	1F		1

Cashbox replaced	20		1
Lid open	21		1
Lid closed	22		1
Calibration fault	24	1 byte fault code	2
Attached mech jam	25		1
Attached mech open	26		1
Smart empying	27	currency data (see table)	variable
Smart emptied	28	currency data (see table)	variable
Barcode escrow	34		1
Barcode stacked	35		1
Multiple coins added	36	currency data (see table)	variable
Peripheral error	37	2 bytes dev code,error code	3
Peripheral device disabled	38	1 byte device code	2
Note held in bezel	39	4 bytes giving the value of the note + 3 bytes country code	8

data byte	**currency data description**
0	Event code (e.g. 01 for dispensing)
1	Number of currencies in event
2-5	Value
6-8	ASCII country code
...	Repeat value and country code for each denomination in the event.

data byte	**incomplete data description**
0	Event code (e.g. 0A for incomplete payout)
1	Number of currencies in event
2-5	Value paid
10-12	ASCII country code
...	Repeat value and country code for each denomination in the event.

6-9	Value requested

Command name	Code dec	Code hex
Get Device Setup (with currency support)	**046**	**0x2E**

back to index

Supported on devices:

SMART Hopper	SMART Payout	SMART System

Description

This command will return the setup of the device, the number of different types of coin/note and the value and currency of each coin/note that the device can handle. The length of the returned data will be 1+(n*7) bytes long, where n is the number of notes/coins that can be used. The response is formatted as a variable number of bytes, byte 0 being the number of denominations in the system and then 7 bytes for each of the denominations in the system, 4 bytes for the value and 3 bytes ascii country code.

Parameters

This command has no parameters

Command packet example:

dec

003	000	001	046	206

hex

03	00	01	2E	CE

Response

This example shows a response for a system with 3 denominations. EUR 0.20, EUR 0.50 and GBP 1.00

dec

001	022	003	000	003	020	000	000	000	069	085	082	050	000	000	000	069	085	082	100	000	000
000	047	042	050	214																	

hex

01	16	03	00	03	14	00	00	00	45	55	52	32	00	00	00	45	55	52	64	00	00
00	2F	2A	32	D6																	

Command name	Code dec	Code hex
Float By Denomination (with currency support)	**045**	**0x2D**

back to index

Supported on devices:

SMART Hopper	SMART Payout	SMART System

Description

This command allows the host to request a selected amount of denominations to be left in the device. All other notes/ coins will be routed to the cashbox. The format of the command depends on the security level setting of the device. This may be obtained using the Request Encryption Support (header 111) command, which returns as part of its data, the [Command level] byte. Successful commands sent with the security levels set return an event count byte. This byte is value 0 at reset and then wraps from 255 to 1. It increments on every successful payout, float or empty command. Non-security levels just return an ACK. The example show here is for no security level set.

The command is formatted in a variable number of data bytes. Byte 0 is the number of denominations in the request, the rest are formatted in blocks of 2 bytes - float level of denomination,4 bytes - value of denomination and 3 bytes ascii value of country code for denomination. This block is then repeated for the required float denominations. Command example shows request to leave levels of 3 x EUR 0.10, 4 x EUR 1.00 and 6 x EUR 0.05 with no security levels set.

dec

003	028	001	045	003	003	000	010	000	000	000	069	085	082	004	000	100	000	000	000	069	085
082	006	000	005	000	000	000	068	085	082	109											

hex

03	1C	01	2D	03	03	00	0A	00	00	00	45	55	52	04	00	64	00	00	00	45	55
52	06	00	05	00	00	00	44	55	52	6D											

Command format

data byte	function
0	Number of denominations in request
1 to 2	Number of coins to leave
3 to 6	Value of denomination
7 to 9	ASCII country code
...	Repeat for each float denomination
...	

Response

ACK response only for none security level requests

dec

001	000	003	000	252

hex

01	00	03	00	FC

NAK response with data byte for failed requests. Example shows request fail for inability to float to the exact amount requested.

dec

001	001	003	005	002	244

hex

01	01	03	05	02	F4

ACK response for commands with security bytes. 1 byte event counter is given. Value is 0 at reset then increments on each successful command. After value 255 it wraps to 1.

dec

001	001	003	000	004	247

hex

01	01	03	00	04	F7

error reason	error code (hex)
Not enough value in device	01
Cannot pay this exact amount	02
Device busy	03
Device disabled	04
Device lid/path open	05
Device jam	06
Calibration error	07
Fraud detected	08

Command name	Code dec	Code hex
Payout by denomination (with currency support)	044	0x2C

back to index

Supported on devices:

SMART Hopper	SMART Payout	SMART System

Description

This command allows the host to request a value amount to be paid from the device specifying the quantity of denominations required. The format of the command depends on the security level setting of the device. This may be obtained using the Request Encryption Support (header 111) command, which returns as part of its data, the [Command level] byte. Successful commands sent with the security levels set return an event count byte. This byte is value 0 at reset and then wraps from 255 to 1. It increments on every successful payout, float or empty command. Non-security levels just return an ACK. The example show here is for no security level set.

Command has a variable number of data bytes. A command array is made up as follows. Byte 0 - the number of denomination requests. Then each block of bytes makes up a denomination request. Bytes 1-2 the number of denominations to pay, Bytes 3-6 the value of the denomination, Bytes 7-9 the ascii code of the country of denomination. This is then repeated for each denomination. This example is a unsecure request to pay 3 x 0.10 EUR, 7 x 1.00 EUR and 12 x 0.02 EUR.

dec

003	028	001	044	003	003	000	010	000	000	000	069	085	083	007	000	100	000	000	000	069	085
082	012	000	002	000	000	000	069	085	082	102											

hex

03	1C	01	2C	03	03	00	0A	00	00	00	45	55	53	07	00	64	00	00	00	45	55
52	0C	00	02	00	00	00	45	55	52	66											

Response

ACK with no data for non-security requests.

dec

001	000	003	000	252

hex

01	00	03	00	FC

ACK with event count data byte for secure command requests.

dec

001	001	003	000	023	228

hex

01	01	03	00	17	E4

NACK with error code byte for failures. This example shows payout request fail for device busy.

dec

001	001	003	005	003	243

hex

01	01	03	05	03	F3

error reason	error code (hex)
Not enough value in device	01
Cannot pay this exact amount	02
Device busy	03
Device disabled	04
Device lid/path open	05
Device jam	06
Calibration error	07
Fraud detected	08

Command name	Code dec	Code hex
Set Denomination Amount (with currency support)	**043**	**0x2B**

back to index

Supported on devices:

SMART Hopper	SMART System

Description

This command will add the number of coins specified in Count to the internal coin counter for the value specified in Value. If the count specified is Zero then the counter will be reset. It is not possible to set the absolute value to anything except zero in a single command. This command should be used each time the coin acceptor routes a coin into the hopper, or when the hopper has coins manually added. This command is invalid for the Smart Payout as the notes are automatically added to the counter by the note validator.

Command has 9 data bytes. Bytes 0-3 denomination value, Bytes 4-5 the level to add, Bytes 6-8 the country code. The example adds 10 EUR 0.50c coins to the system.

dec

003	009	001	043	050	000	000	000	010	000	069	085	082	160

hex

03	09	01	2B	32	00	00	00	0A	00	45	55	52	A0

Response

ACK for success.

dec

001	000	003	000	252

hex

01	00	03	00	FC

NACK for fail due to denomination/country not existing on system.

dec

001	000	003	005	247

hex

01	00	03	05	F7

Command name	Code dec	Code hex
Get Denomination Amount (with currency support)	**042**	**0x2A**

back to index

Supported on devices:

SMART Hopper	SMART Payout	SMART System

Description

This command will return the count of the number of coins/notes of the value specified in the command data.

Command has 7 data bytes. Bytes 0-3 is the value, Bytes 4-7 give the ascii country code. This example asks for the number of 5.00 EUR notes stored in the system.

dec

003	007	001	042	244	001	000	000	069	085	082	234

hex

03	07	01	2A	F4	01	00	00	45	55	52	EA

Response

ACK response with 2 data bytes giving the level of that denomination. This example shows a level of 10.

dec

001	001	003	000	010	241

hex

01	01	03	00	0A	F1

NAK for failure due to denomination/country not in system.

dec

001	000	003	005	247

hex

01	00	03	05	F7

Command name	Code dec	Code hex
Get Minimum Payout (with currency support)	**041**	**0x29**

back to index

Supported on devices:

SMART Hopper	SMART Payout	SMART System

Description

This command will return the value of the minimum payout that is possible with the coins/notes that are currently in the device. This is effectively the value of the lowest coin/note in the device.

Command data is 3 data bytes. This contains the currency code for the minimum request. This example asks for the min EUR payout.

dec

003	003	001	041	069	085	082	228

hex

03	03	01	29	45	55	52	E4

Response

ACK with 4 bytes of data showing the min payout value. This example shows a minimum payout of 1.00 EUR.

dec

001	004	003	000	100	000	000	000	148

hex

01	04	03	00	64	00	00	00	94

Command name	Code dec	Code hex
Float Amount (with currency support)	**040**	**0x28**

back to index

Supported on devices:

SMART Hopper	SMART Payout	SMART System

Description

This command allows the host to request a value amount to be left in the device, paying out all excess monies into the device cashbox. The format of the command depends on the security level setting of the device. This may be obtained using the Request Encryption Support (header 111) command, which returns as part of its data, the [Command level] byte. Successful commands sent with the security levels set return an event count byte. This byte is value 0 at reset and then wraps from 255 to 1. It increments on every successful payout, float or empty command. Non-security levels just return an ACK. The example show here is for no security level set.

Command has 11 data bytes. Bytes 0-3 is the minimum payout remaining amount. Bytes 4-7 are the value to float to. Bytes 8-10 is the country code. This example is a request to float to 20.00 EUR with a min payout of 5.00 EUR

dec

003	011	001	040	244	001	000	000	208	007	000	000	069	085	082	017

hex

03	0B	01	28	F4	01	00	00	D0	07	00	00	45	55	52	11

Response

ACK with no data for non-security requests.

dec

001	000	003	000	252

hex

01	00	03	00	FC

ACK with event count data byte for secure command requests.

dec

001	001	003	000	023	228

hex

01	01	03	00	17	E4

NACK with error code byte for failures. This example shows payout request fail for device busy.

dec

001	001	003	005	003	243

hex

01	01	03	05	03	F3

error reason	error code (hex)
Not enough value in device	01
Cannot pay this exact amount	02
Device busy	03
Device disabled	04
Device lid/path open	05
Device jam	06
Calibration error	07
Fraud detected	08

Command name	Code dec	Code hex
Payout Amount (with currency support)	**039**	**0x27**

back to index

Supported on devices:

SMART Hopper	SMART Payout	SMART System

Description

This command allows the host to request a value amount to be paid from the device. The format of the command depends on the security level setting of the device. This may be obtained using the Request Encryption Support (header 111) command, which returns as part of its data, the [Command level] byte. Successful commands sent with the security levels set return an event count byte. This byte is value 0 at reset and then wraps from 255 to 1. It increments on every successful payout, float or empty command. Non-security levels just return an ACK. The example show here is for no security level set.

Non security command has 7 data bytes. Bytes 0-3 is the value to be paid out. Bytes 4-7 is the ascii country code. This example is a request for a payout of 10.32 EUR

dec

003	007	001	039	008	004	000	000	069	085	082	214

hex

03	07	01	27	08	04	00	00	45	55	52	D6

Response

ACK with no data for non-security requests.

dec

001	000	003	000	252

hex

01	00	03	00	FC

ACK with event count data byte for secure command requests.

dec

001	001	003	000	023	228

hex

01	01	03	00	17	E4

NACK with error code byte for failures. This example shows payout request fail for device busy.

dec

001	001	003	005	003	243

hex

01	01	03	05	03	F3

error reason	error code (hex)
Not enough value in device	01
Cannot pay this exact amount	02
Device busy	03
Device disabled	04
Device lid/path open	05
Device jam	06
Calibration error	07
Fraud detected	08

Command name	Code dec	Code hex
Get Routing (with currency support)	**038**	**0x26**

back to index

Supported on devices:

SMART Hopper	SMART Payout	SMART System

Description

This command returns the route setting for a particular value denomination. 0 for values routed to be stored for payout, 1 for values to be routed to the external cashbox.

Command has 7 bytes, Bytes 0-3 is the value, byte 4-6 is the country code. This example is request for the route of 0.50 EUR coin.

dec

003	007	001	038	050	000	000	000	069	085	082	177

hex

03	07	01	26	32	00	00	00	45	55	52	B1

Response

An ACK with data byte, 0 = route payout, 1 = route cashbox. Example shows route on device has been set to cashbox

dec

001	001	003	000	001	250

hex

01	01	03	00	01	FA

Command name	Code dec	Code hex
Set Route (with currency support)	**037**	**0x25**

back to index

Supported on devices:

SMART Hopper	SMART Payout	SMART System

Description

A command to control the route of a denomination entered in to the device. Money can either be stored in the device available for payout or sent to an external cashbox. For routes to cashbox - In the case of the Smart Payout, notes entered will be routed straight to the cashbox; with the Smart Hopper/Smart System coins will be routed to the cashbox as they are detected by the hopper discrimination system.

This command has 8 data bytes. Byte 0 is the desired route, 0 for payout, 1 for cashbox. Bytes 1-4 is the value, Bytes 5-7 is the ascii country code. This example shows a command to set 5.00 EUR note to route to cashbox.

dec

003	008	001	037	001	244	001	000	000	069	085	082	237

hex

03	08	01	25	01	F4	01	00	00	45	55	52	ED

Response

ACK for success.

dec

001	000	003	000	252

hex

01	00	03	00	FC

Command name	Code dec	Code hex
Set Bezel Mode	**035**	**0x23**

back to index

Supported on devices:

SMART Payout

Description

A command to set the colour mix mode of the Smart Payout Bezel.

This example sets the bezel color to RED to be stored in the EEPROM.

dec

003	009	001	035	000	255	000	000	001	000	000	000	000	208

hex

03	09	01	23	00	FF	00	00	01	00	00	00	00	D0

Bezel mode command format

data byte	function
0	Bezel type (see table)
1	RED pwm colour (0-255)
2	GREEN pwm colour (0-255)
3	BLUE pwm colour (0-255)
4	Store mode (0 = RAM, 1 = EEPROM)
5 - 8	not used - set to 0

Bezel types

Type	Description
0	Standard (steady colour when enabled)
1	Flashing bezel when enabled

2	Bezel colour when disabled (all PWM values to 255 will turn bezel off when disabled)

Response

ACK response.

dec

001	000	003	000	252

hex

01	00	03	00	FC

Command name	Code dec	Code hex
Run Unit Calibration	**034**	**0x22**

back to index

Supported on devices:

SMART Hopper	SMART Payout	SMART System

Description

A command to set the host to run its self-calibration function. This is done in response to a calibration fault event type 7 and is used so that the host can optimise its power spread by determining when the device motors will activate. This command will only function if the Host Calibration option is enabled.

Parameters

This command has no parameters

Command packet example:

dec

003	000	001	034	218

hex

03	00	01	22	DA

Response

ACK response

dec

001	000	003	000	252

hex

01	00	03	00	FC

Command name	Code dec	Code hex
Float By Denomination	**033**	**0x21**

back to index

Supported on devices:

SMART Hopper	SMART Payout	SMART System

Description

This command allows the host to request a selected amount of denominations to be left in the device. The format of the command depends on the security level setting of the device. This may be obtained using the Request Encryption Support (header 111) command, which returns as part of its data, the [Command level] byte. Successful commands sent with the security levels set return an event count byte. This byte is value 0 at reset and then wraps from 255 to 1. It increments on every successful payout, float or empty command. Non-security levels just return an ACK. The example show here is for no security level set.

The command is formatted in a variable number of data bytes. Byte 0 is the number of denominations in the request, the rest are formatted in blocks of 2 bytes - float level of denomination,4 bytes - value of denomination. This block is then repeated for the required float denominations. Command example shows request to leave levels of 3 x 0.10, 4 x 1.00 and 6 x 0.05 with no security levels set.

dec

003	019	001	033	003	003	000	010	000	000	000	004	000	100	000	000	000	006	000	005	000	000
000	069																				

hex

03	13	01	21	03	03	00	0A	00	00	00	04	00	64	00	00	00	06	00	05	00	00
00	45																				

Command data format

data byte	function
0	Number of denominations in request
1 to 2	Number of coins to pay
3 to 6	Value of coin
...	Repeat for each qty and denomination
...	

Response

ACK response only for none security level requests

dec

001	000	003	000	252

hex

01	00	03	00	FC

NAK response with data byte for failed requests. Example shows request fail for inability to float to the exact amount requested.

dec

001	001	003	005	002	244

hex

01	01	03	05	02	F4

ACK response for commands with security bytes. 1 byte event counter is given. Value is 0 at reset then increments on each successful command. after value 255 it wraps to 1.

dec

001	001	003	000	004	247

hex

01	01	03	00	04	F7

error reason	error code (hex)
Not enough value in device	01
Cannot pay this exact amount	02
Device busy	03
Device disabled	04
Device lid/path open	05
Device jam	06
Calibration error	07
Fraud detected	08

back to index

Command name	Code dec	Code hex
Payout By Denomination	**032**	**0x20**

Supported on devices:

SMART Hopper	SMART Payout	SMART System

Description

This command allows the host to request a value amount to be paid from the device specifying the quantity of denominations required. The format of the command depends on the security level setting of the device. This may be obtained using the Request Encryption Support (header 111) command, which returns as part of its data, the [Command level] byte. Successful commands sent with the security levels set return an event count byte. This byte is value 0 at reset and then wraps from 255 to 1. It increments on every successful payout, float or empty command. Non-security levels just return an ACK. The example show here is for no security level set.

Command example shows payout request of 3 x 0.10, 4 x 1.00 and 6 x 0.05 with no security levels set.

dec

003	019	001	032	003	003	000	010	000	000	000	004	000	100	000	000	000	006	000	005	000	000
000	070																				

hex

03	13	01	20	03	03	00	0A	00	00	00	04	00	64	00	00	00	06	00	05	00	00
00	46																				

Command data format

data byte	function
0	Number of denominations in request
1 to 2	Number of coins to pay
3 to 6	Value of coin
...	Repeat for each qty and denomination
...	

Response

ACK response only for none security level requests

dec

001	000	003	000	252

hex

01	00	03	00	FC

NAK response with data byte for failed requests. Example shows request fail for inability to pay the exact amount requested.

dec

001	001	003	005	002	244

hex

01	01	03	05	02	F4

ACK response for commands with security bytes. 1 byte event counter is given. Value is 0 at reset then increments on each successful command. after value 255 it wraps to 1.

dec

001	001	003	000	004	247

hex

01	01	03	00	04	F7

error reason	**error code (hex)**
Not enough value in device	01
Cannot pay this exact amount	02
Device busy	03
Device disabled	04
Device lid/path open	05
Device jam	06
Calibration error	07
Fraud detected	08

Command name	Code dec	Code hex
Get Payout Options	**031**	**0x1F**

back to index

Supported on devices:

SMART Hopper	SMART Payout	SMART System

Description

This command will return two bytes giving the current status of the device options. REG 0 is transmitted first.

Parameters

This command has no parameters

Command packet example:

dec

003	000	001	031	221

hex

03	00	01	1F	DD

Response

Response showing two option bytes.

dec

001	002	003	000	007	000	243

hex

01	02	03	00	07	00	F3

function	action	default	device
REG 0			
bit 0	Pay mode 0 = free pay, 1 = High value split	1	SH, SS
bit 1	Level check 0 = disabled, 1 = enabled	1	SH, SS
bit 2	Motor speed 0 = low, 1 = high	1	SH, SS
bit 3	Not used set to 0	0	

bit 4	Not used set to 0	0	
bit 5	Not used set to 0	0	
bit 6	Not used set to 0	0	
bit 7	Not used set to 0	0	
REG 1			
bit 0	Enable Note In Bezel Hold message. 1 = enabled, 0 = disabled.	0	SP
bit 1	Not used set to 0	0	
bit 2	Not used set to 0	0	
bit 3	Not used set to 0	0	
bit 4	Not used set to 0	0	
bit 5	Not used set to 0	0	
bit 6	Not used set to 0	0	
bit 7	Not used set to 0	0	

Command name	Code dec	Code hex
Set Payout Options	**030**	**0x1E**

back to index

Supported on devices:

SMART Hopper	SMART Payout	SMART System

Description

The host can set run-time options for the SMART Hopper, SMART Payout or SMART System device.

Example setup for default setup - high value split,Level check on and high speed motor.

dec

003	001	001	030	007	214

hex

03	01	01	1E	07	D6

Option data format

function	action	default	device
REG 0			
bit 0	Pay mode 0 = free pay, 1 = High value split	1	SH, SS
bit 1	Level check 0 = disabled, 1 = enabled	1	SH, SS
bit 2	Motor speed 0 = low, 1 = high	1	SH, SS
bit 3	Not used set to 0	0	
bit 4	Not used set to 0	0	
bit 5	Not used set to 0	0	
bit 6	Not used set to 0	0	
bit 7	Not used set to 0	0	
REG 1			
bit 0	Enable Note In Bezel Hold message. 1 = enabled, 0 = disabled.	0	SP

bit 1	Not used set to 0	0	
bit 2	Not used set to 0	0	
bit 3	Not used set to 0	0	
bit 4	Not used set to 0	0	
bit 5	Not used set to 0	0	
bit 6	Not used set to 0	0	
bit 7	Not used set to 0	0	

Response

Responds with ACK for supported devices

dec

001	000	003	000	252

hex

01	00	03	00	FC

Command name	Code dec	Code hex
Request Status	**029**	**0x1D**

back to index

Supported on devices:

SMART Hopper	SMART Payout	SMART System

Description

This command will return the status of the device and the progress of the current requested operation. After issuing any action commands this command should be used to track the status. The possible status codes are shown in the table below. Please note that more than one event and associated data may be returned in one request.

Parameters

This command has no parameters

Command packet example:

dec

003	000	001	029	223

hex

03	00	01	1D	DF

Response

Example response data for Dispensing event value 5.30. The full event list is shown in the table below.

dec

001	005	003	000	001	018	002	000	000	226

hex

01	05	03	00	01	12	02	00	00	E2

event	code (hex)	format	size (bytes)	notes
Idle	00		1	Device is enabled and ready for function.
Dispensing	01	4 bytes giving current value dispensed	5	A requested payout is in process.

Dispensed	02	4 bytes giving total value dispensed	5	A requested payout has finished
Coins Low	03		1	
Empty	04		1	
Jammed	05	4 bytes giving value dispensed at jam	5	A payout has jammed - the value at the jam point is given.
Halted	06	4 byte giving value dispensed at halt	5	A payout has been halted by the host. The value at halt is given.
Floating	07	4 bytes giving value to cashbox	5	A requested float is in process
Floated	08	4 bytes giving value to cashbox	5	A requested float has finished.
Timeout	09	4 bytes giving value dispensed at timeout	5	A requested payout function has timed-out. The value paid at the time-out point is given.
Incomplete payout	0A	4 bytes value dispensed, 4 bytes value requested	9	At start-up a discrepancy between the last paid amount and last requested amount was detected.
Incomplete float	0B	4 bytes value dispensed, 4 bytes value requested	9	At start-up a discrepancy between the last floated amount and last requested amount was detected.
Cashbox paid	0C	4 bytes giving value paid into the cashbox	5	Coins to this value have been passed to the external cashbox.
Coin credit	0D	4 bytes giving value of coin added by attached mech.	5	Coins have been detected as paid into the cashbox by an attached coin mech.
Emptying	0E		1	The payout unit is in the process of emptying its contents to the system cashbox
Emptied	0F		1	The payout unit has emptied its contents to the system cashbox
Fraud attempt	10	4 bytes giving value dispensed at fraud	5	A tamper attempt has been detected.
Disabled	11		1	The unit device is disabled and not available for operation.
Note stored	12		1	A banknote has been passed to the payout store.

Slave reset	13		1	The device has undergone a power-up reset.
Note read	14	4 bytes giving the value of the note	5	A banknote is in the process of being scanned in the device. If the value is not known the data value is 0. When the note value is known, its channel number is the data value.
Note credit	15	4 byte value giving the value of the note credited	5	A banknote has been passed to a secure storage point in the system and the host can safely issue its credit amount.
Note rejecting	16		1	A banknote is in the process of being rejected from the device.
Note rejected	17		1	A banknote has been rejected from the device.
Note stacking	18		1	A banknote is in the process of being moved from the escrow position to the system stack position.
Note stacked	19		1	A banknote has been placed in the system stack position.
Note path jam	1A		1	A banknote has been detected as stuck in the device note path. This is classed as an unsafe jam as the note may be retrievable by the user.
Note stack jam	1B		1	A banknote is detected as being stuck in the device stack mehanism. This is classed as a safe jam as it is not usually retrievable by the user.
Note from front at start	1C	4 byte value of note rejected (0 if unknown)	5	A banknote has been detected as being rejected from the device front during a pwoer-up process.
Note stacked at start	1D	4 byte value of note stacked (0 if unknown)	5	A banknote has been detected as being moved to the device stacker mechanism during a power-up process.
Cashbox full	1E		1	The banknote stacker store has been detected as being full.
Cashbox removed	1F		1	The device removable cashbox has been detected as being removed.
Cashbox replaced	20		1	The device removable cashbox has been detected as being replaced.
Lid open	21		1	
Lid closed	22		1	
Calibration fault	24	1 byte fault code	2	
Attached mech jam	25		1	The coin mech attached to the payout device has been detected as having a jam condition.
Attached mech open	26		1	The coin mech attached to the payout device has been detected as having an open deck.

Smart empying	27	4 byte emptied value so far.	5	The payout device is in a smart emptying process.
Smart emptied	28	4 byte value emptied	5	The payout device has completed its smart emptying process.
Barcode escrow	34		1	A bar code ticket has been detected and scanned by the device and is held in escrow.
Barcode stacked	35		1	A bar code ticket has been accepted by the device and has passed to the stacker mechanism
Multiple coins added	36	4 byte value of coins added since last request	5	The coin feeder has passed this value to the hopper unit since the last status request.
Peripheral error	37	2 bytes dev code,error code	3	The attached Peripheral to the payout device has generated an error.
Peripheral device disabled	38	1 byte device code	2	The attached Peripheral to the payout device is in a disabled state.
Note held in bezel	39	4 byte value of note held	5	A dispensed banknote is currently held in the bezel. This event is only given if the option is enabled in Set Payout Options command.

Command name	Code dec	Code hex
Get Device Setup	**028**	**0x1C**

back to index

Supported on devices:

SMART Hopper	SMART Payout	SMART System

Description

This command will return the setup of the device, containing the currency, the number of different types of coin/note and the value of each coin/note that the device can handle. The length of the returned data will be 4+(n*4) bytes long, where n is the number of notes/coins that can be used. The response data is formatted as follows: bytes 0 to 2 - the country code of the device give as ASCII code, byte 3 - the number of denominations available in the device, byte 4 to byte x - the 4 byte value for each of the denominations

Parameters

This command has no parameters

Command packet example:

dec

003	000	001	028	224

hex

03	00	01	1C	E0

Response

An example response for a device with EUR currency with 3 note values available for storage/payout: 5 EUR,10 EUR, 20 EUR.

dec

001	016	003	000	069	085	082	003	244	001	000	000	232	003	000	000	208	007	000	000	070

hex

01	10	03	00	45	55	52	03	F4	01	00	00	E8	03	00	00	D0	07	00	00	46

Command name	Code dec	Code hex
Set Denomination Amount	**027**	**0x1B**

back to index

Supported on devices:

SMART Hopper	SMART System

Description

This command will add the number of coins specified in Count to the internal coin counter for the value specified in Value. If the count specified is zero, then the counter will be reset. It is not possible to set the absolute value to anything except zero in a single command. This command should be used each time the coin acceptor routes a coin into the hopper, or when the hopper has coins manually added. This command is invalid for the Smart Payout as the notes are automatically added to the counter by the note validator.

The command data consists of 6 bytes. Bytes 0-3 give the value of the coin to set, bytes 4 - 5 are the number of coins to add to the system. This example shows 20 EUR 0.50 coins being added to the system.

dec

003	006	001	027	050	000	000	000	020	000	149

hex

03	06	01	1B	32	00	00	00	14	00	95

Response

ACK response for successful update of coin level.

dec

001	000	003	000	252

hex

01	00	03	00	FC

Example showing that device does not have this value of coin in its system.

dec

001	001	003	005	001	245

hex

01	01	03	05	01	F5

back to index

Command name	Code dec	Code hex
Get Denomination Amount	**026**	**0x1A**

Supported on devices:

SMART Hopper	SMART Payout	SMART System

Description

This command will return the count of the number of coins/notes of the value specified in the command data. The response contains two data bytes representing the amount of notes/coins stored for the command data given. If the value given in the command is not present in the device then a NAK is returned with data byte value 1.

An example command requesting the number of EUR 5.00 notes stored in the device.

dec

003	004	001	026	244	001	000	000	233

hex

03	04	01	1A	F4	01	00	00	E9

Response

Example response showing device has 12 notes stored.

dec

001	002	003	000	012	000	238

hex

01	02	03	00	0C	00	EE

An example showing NAK response to a value not present in the device.

dec

001	001	003	005	001	245

hex

01	01	03	05	01	F5

Command name	Code dec	Code hex
Get Minimum Payout	**025**	**0x19**

back to index

Supported on devices:

SMART Hopper	SMART Payout	SMART System

Description

This command will return the value of the minimum payout that is possible with the coins/notes that are currently in the device. This is effectively the value of the lowest coin/note in the device.

Parameters

This command has no parameters

Command packet example:

dec

003	000	001	025	227

hex

03	00	01	19	E3

Response

Example response showing device has a minimum payout available of EUR 0.20

dec

001	003	003	000	020	000	000	229

hex

01	03	03	00	14	00	00	E5

Command name	Code dec	Code hex
Empty	**024**	**0x18**

back to index

Supported on devices:

SMART Hopper	SMART Payout	SMART System

Description

This command allows the host to request the device to empty all of its store of monies into the external cashbox for collection. No values of cashbox payout are given during the empty procedure and after the device has emptied, the denomination counters will all be set to zero. The format of the command depends on the security level setting of the device. This may be obtained using the Request Encryption Support (header 111) command, which returns as part of its data, the [Command level] byte. Successful commands sent with the security levels set return an event count byte. This byte is value 0 at reset and then wraps from 255 to 1. It increments on every successful payout, float or empty command. Non-security levels just return an ACK.

Parameters

This command has no parameters

Command packet example:

dec

003	000	001	024	228

hex

03	00	01	18	E4

Command name	Code dec	Code hex
Float Amount	**023**	**0x17**

back to index

Supported on devices:

SMART Hopper	SMART Payout	SMART System

Description

This command allows the host to request a value amount to be left in the device, paying out all excess monies into the device cashbox. The format of the command depends on the security level setting of the device. This may be obtained using the Request Encryption Support (header 111) command, which returns as part of its data, the [Command level] byte. Successful commands sent with the security levels set return an event count byte. This byte is value 0 at reset and then wraps from 255 to 1. It increments on every successful payout, float or empty command. Non-security levels just return an ACK. The example show here is for no security level set.

Float amount to EUR 5.30 (no security bytes)

dec

003	004	001	023	018	002	000	000	205

hex

03	04	01	17	12	02	00	00	CD

Response

Returns ACK

dec

001	000	003	000	252

hex

01	00	03	00	FC

Example showing NAK failure for not enough value in the payout.

dec

001	001	003	005	001	245

hex

01	01	03	05	01	F5

error reason	error code (hex)

Not enough value in device	01
Cannot pay this exact amount	02
Device busy	03
Device disabled	04
Device lid/path open	05
Device jam	06
Calibration error	07
Fraud detected	08

Command name	Code dec	Code hex
Payout Amount	**022**	**0x16**

back to index

Supported on devices:

SMART Hopper	SMART Payout	SMART System

Description

This command allows the host to request a value amount to be paid from the device. The format of the command depends on the security level setting of the device. This may be obtained using the Request Encryption Support (header 111) command, which returns as part of its data, the [Command level] byte. Successful commands sent with the security levels set return an event count byte. This byte is value 0 at reset and then wraps from 255 to 1. It increments on every successful payout, float or empty command. Non-security levels just return an ACK. The example show here is for no security level set.

Payout amount EUR 5.30 (no security bytes)

dec

003	004	001	022	018	002	000	000	206

hex

03	04	01	16	12	02	00	00	CE

Response

Returns ACK

dec

001	000	003	000	252

hex

01	00	03	00	FC

Example showing NAK failure for not enough value in the payout.

dec

001	001	003	005	001	245

hex

01	01	03	05	01	F5

error reason	error code (hex)

Not enough value in device	01
Cannot pay this exact amount	02
Device busy	03
Device disabled	04
Device lid/path open	05
Device jam	06
Calibration error	07
Fraud detected	08

Command name	Code dec	Code hex
Get Routing	**021**	**0x15**

back to index

Supported on devices:

SMART Hopper	SMART Payout	SMART System

Description

This command returns the route setting for a particular value denomination. 0 for values routed to be stored for payout, 1 for values to be routed to the external cashbox.

Example command for requesting route of EUR 5.00.

dec

003	004	001	021	244	001	000	000	238

hex

03	04	01	15	F4	01	00	00	EE

Response

Response showing route of request was set to cashbox

dec

001	001	003	000	001	250

hex

01	01	03	00	01	FA

If route cannot be set (value of coin/note not found) a NAK is returned with data byte 1.

dec

001	001	003	005	001	245

hex

01	01	03	05	01	F5

Command name	Code dec	Code hex
Set Routing	**020**	**0x14**

back to index

Supported on devices:

SMART Hopper	SMART Payout	SMART System

Description

A command to control the route of a denomination entered in to the device. Money can either be stored in the device available for payout or sent to an external cashbox. For routes to cashbox - In the case of the Smart Payout, notes entered will be routed straight to the cashbox; with the Smart Hopper/Smart System coins will be routed to the cashbox as they are detected by the hopper discrimination system.

Command has 5 data bytes. Byte 0 sets the desired route, 0 for payout,1 for cashbox. Bytes 1 to 4 are the value of the coin to route. This example command shows EUR 2.00 being set to route to payout.

dec

003	005	001	020	000	200	000	000	000	027

hex

03	05	01	14	00	C8	00	00	00	1B

An example command showing EUR 0.50 coin set to be routed to the cashbox.

dec

003	005	001	020	001	050	000	000	000	176

hex

03	05	01	14	01	32	00	00	00	B0

Response

If successful route change returns ACK

dec

001	000	003	000	252

hex

01	00	03	00	FC

If route cannot be set (value of coin/note not found) a NAK is returned with data byte 1.

dec

001	001	003	005	001	245

hex

01	01	03	05	01	F5

Command name	Code dec	Code hex
Request Comms Revision	**004**	**0x4**

back to index

Supported on devices:

SMART Hopper	SMART Payout	SMART System

Description

As a reply to this command, the Smart Payout or Hopper sends the implementation level of the cctalkÂ® protocol and the communication software version. E.g. Data 1 = 0 Data 2 = 1 Data 3 = 2 Data 4 = 0 Gives a revision of 1.2.0. This version ties in with the version of the specification document and allows updates and changes to be tracked.

Parameters

This command has no parameters

Command packet example:

dec

003	000	001	004	248

hex

03	00	01	04	F8

Response

Example response for coms rev 0.1.1.2

dec

001	004	003	000	000	001	001	002	244

hex

01	04	03	00	00	01	01	02	F4

Command name	Code dec	Code hex
Reset Device	**001**	**0x1**

back to index

Supported on devices:

SMART Hopper	SMART Payout	SMART System

Description

This command causes the device to carry out a full reset. The Device sends a positive acknowledgement immediately before making the reset.

Parameters

This command has no parameters

Command packet example:

dec

003	000	001	001	251

hex

03	00	01	01	FB

Response

Returns ACK

dec

001	000	003	000	252

hex

01	00	03	00	FC