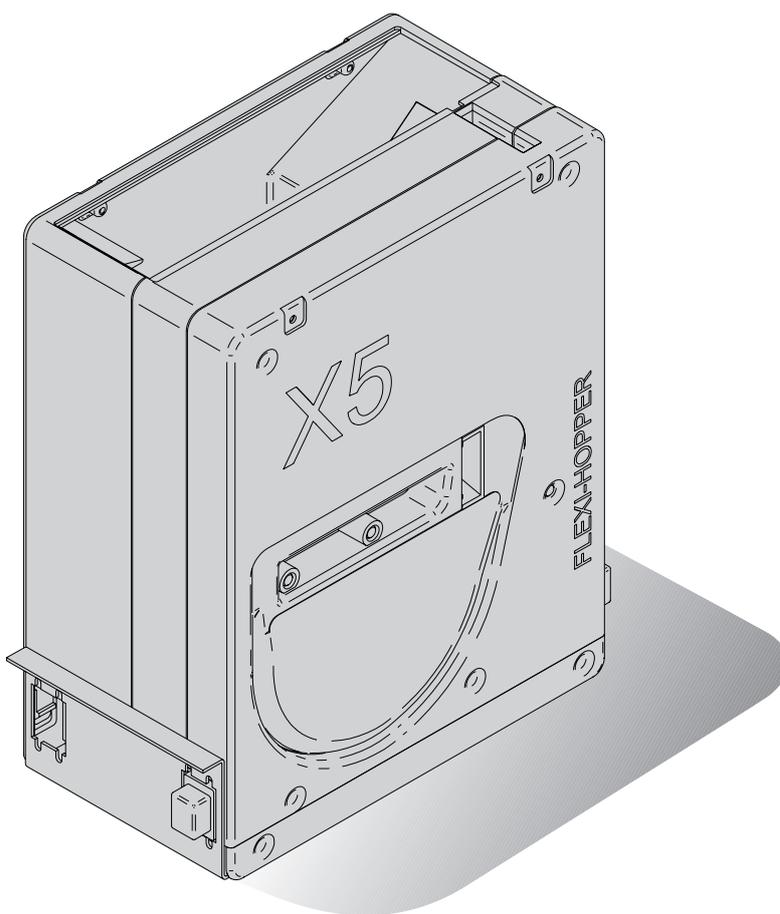


MICROHARD
VENDING PROJECTS

EROGATORE DI MONETE E/O GETTONI

X5 Family

ccTalk protocol/AES



rev. 7-10-02-16

TECHNICAL MANUAL

Documento riservato ai termini di legge con divieto di riproduzione o di trasmissione a terzi senza esplicita autorizzazione della ditta **MICROHARD s.r.l.**

I componenti possono subire aggiornamenti e quindi presentare particolari diversi da quelli raffigurati, senza per questo costituire pregiudizio per i testi contenuti in queste istruzioni.

MICROHARD s.r.l. non si riterrà responsabile di inconvenienti, rotture, incidenti ecc. dovuti alla non conoscenza o alla non applicazione delle norme contenute nelle presenti Istruzioni. Lo stesso dicasi per l'esecuzione di modifiche, varianti e/o per l'uso di parti non autorizzate.

INDICE

1 INFORMAZIONI GENERALI

- 1.1 DESCRIZIONE
- 1.2 CARATTERISTICHE PRINCIPALI
 - 1.2.1 USCITA MONETE
 - 1.2.2 CODICE DI ERRORE
 - 1.2.3 POSIZIONE SCHEDA PCB
 - 1.2.4 STANDARD ccTALK
 - 1.2.5 POSIZIONE CONNETTORI
 - 1.2.6 SISTEMA ANTI INCEPPAMENTO
- 1.3 SICUREZZA
- 1.4 DIMENSIONI
- 1.5 DATI TECNICI
- 1.6 ALIMENTAZIONE MOTORE

2 INSTALLAZIONE

3 INFORMAZIONI ELETTRICHE

- 3.1 DESCRIZIONE GENERALE
- 3.2 ALIMENTAZIONE
- 3.3 MODI DI FUNZIONAMENTO
- 3.4 SENSORI OTTICI
- 3.5 INDICATORI LED
- 3.6 PLACCHE LIVELLO MONETE

4 SPECIFICHE ELETTRICHE

5 RICAMBI

1 INFORMAZIONI GENERALI

1.1 DESCRIZIONE

X5 è un nuovo erogatore di monete e gettoni e può essere utilizzato in diverse applicazioni come kiosk di pagamento, casse automatiche, macchine da gioco e dispositivi cambia-monete e cambia cambia.

1.2 CARATTERISTICHE PRINCIPALI

5 configurazioni di prodotto e funzionamento

1.2.1 USCITA MONETE

Utilizzando diversi accessori a corredo per X5 è possibile determinare 5 diverse posizioni di uscita delle monete o gettoni, deviando il flusso dalla posizione impostata.

1.2.2 CODICE DI ERRORE

L'accensione del LED giallo del X5 segnala: che il dispositivo è alimentato; attraverso un serie di differenti lampeggi il codice dell'errore, consentendo la veloce individuazione della causa del malfunzionamento oppure il conteggio delle monete pagate in funzionamento ordinario

1.2.3 POSIZIONE SCHEDA PCB

La scheda PCB consente tutte le funzionalità del X5 –Pay-out è aggiornabile dall' esterno senza dovere smontare parti dell'erogatore,

1.2.4 STANDARD ccTALK

Nelle versioni X5 Pay-out l'unità funziona utilizzando lo standard ccTalk. È possibile scegliere il dispositivo con connettore Cinch 12 vie oppure con connettore a vaschetta 10 pins standard CC-Talk. La versione parallela è disponibile con il solo connettore cinch.

1.2.5 POSIZIONE CONNETTORI

il connettore per il funzionamento ccTalk o il connettore a 12 pin (Cinch) possono essere alloggiati in due diverse posizioni:

- lato opposto finestra di uscita monete
- lato finestra di uscita monete. (reverse)

1.2.6 SISTEMA ANTI INCEPPAMENTO

Se si dovesse verificare un inceppamento del motore o della cinghia avviene un arresto del motore con successiva ripartenza con moto contrario; poi un ulteriore arresto e ripartenza nella corretta direzione. Se ciò non dovesse avvenire

l'operazione verrà ripetuta con 3 tentativi.

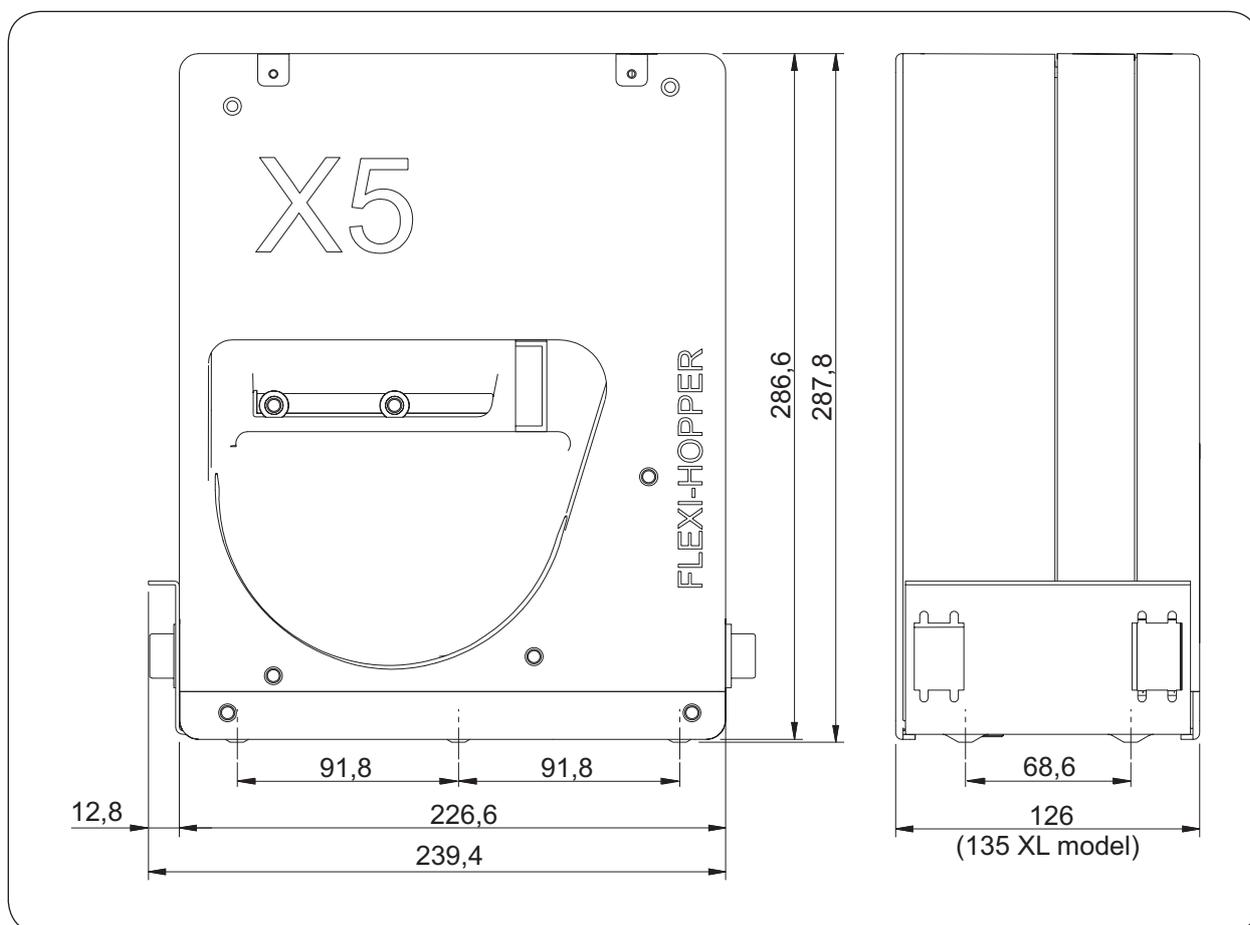
1.3 SICUREZZA



X5 non deve essere collegato/scollegato dalla slitta base con alimentazione inserita.

Non introdurre le mani all'interno del X5 durante il suo funzionamento poiché sono presenti parti meccaniche in movimento.

1.1 DIMENSIONI



1.5 DATI TECNICI

Velocità di erogazione	4 coins/sec
Capacità monete	1500 pezzi del valore di 1,00 •
Peso (a vuoto)	kg 2
Diametri monete erogate	da 16 a 39 mm (da 20 a 31,5 mm con catena standard)
Spessori monete erogate	da 1,25 a 4,5 mm

1.6 ALIMENTAZIONE MOTORE

	standby	empty	max load	forced stop
MOTORE 24Vdc \pm 10%	0 mA	80 mA	500 mA	(transient) 500 mA
LOGICA 12Vdc \pm 10%	80 mA	80 mA	80 mA	-

standby:	X5 fermo ma alimentato
empty:	normale funzionamento
max load:	funzionamento con serbatoio monete pieno
forced stop:	referito al valore di corrente assorbita dal motore oltre la quale si intende inceppato, si avvia la procedura di anti inceppamento.

2 INSTALLAZIONE



NON DARE ALIMENTAZIONE ELETTRICA AL X5 FINO AL COMPLETAMENTO, E DOPO LE OPPORTUNE VERIFICHE, DEL MONTAGGIO

- Fissare la slitta del X5 alla macchina.
- Verificare il corretto inserimento tra spina e presa in caso di utilizzo del connettore standard 12 pin.
- Collegare il connettore flat o 12 pin seguendo gli indirizzi di ogni singolo pin come riportato nel par. 4.1, utilizzando un cavo adatto a sopportare correnti e tensioni massime.
- Innestare X5 sulla slitta fino al suo totale inserimento
- Dare alimentazione elettrica.

3 INFORMAZIONI ELETTRICHE

3.1 DESCRIZIONE GENERALE

I modi di funzionamento del X5 sono gestiti da un microprocessore:
protocollo ccTalk/AES
protocollo parallelo
protocollo multi moneta
contadividi moneta

3.2 ALIMENTAZIONE

X5 è equipaggiato con un motore alimentato in tensione continua a 24 V.

3.3 MODI DI FUNZIONAMENTO

MOD. X5-cc-Talk/AES

Funziona attraverso il protocollo ccTalk/AES.

3.4 SENSORI OTTICI

Sono presenti una coppia di sensori ottici per il rilevamento delle monete pagate (anche con foro centrale) e una coppia di sensori induttivi per i modelli multicoins.

MOD. X5-ccTalk/AES

Attraverso la linea dati il protocollo ccTalk/AES monitorizza tutte le funzioni dei sensori.

3.5 INDICATORI LED

MOD. X5-ccTalk/AES

Possiede un solo LED di colore verde lampeggiante:

lampeggio veloce indica che X5 stà erogando una moneta per lampeggio

lampeggio con intervallo più lungo indica che X5 è in funzione

led fisso acceso X5 è in errore: può comprendere errore fotocellule oppure dati corrotti in EEPROM oppure alimentazione insufficiente.

3.6 PLACCHE LIVELLO MONETE

All'interno del X5 sono presenti delle piastre di ottone per il rilevamento del livello delle monete.

MOD. X5-cc-Talk/AES

I segnali delle piastre sono gestiti internamente dal protocollo ccTalk.

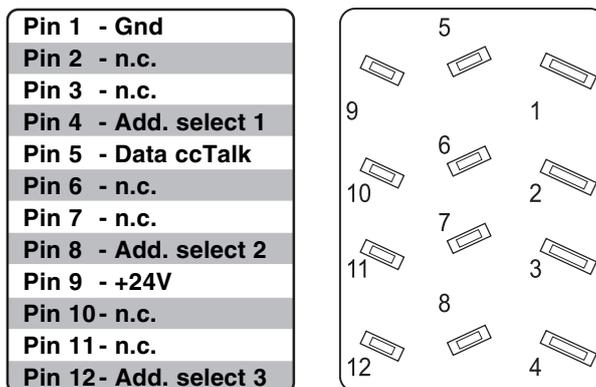
4- SPECIFICHE ELETTRICHE



N.B. PER IL COLLEGAMENTO DEL X5 UTILIZZARE UN CAVO 22AWG.

4.1 CONNETTORE MOD. X5-cc-Talk/AES

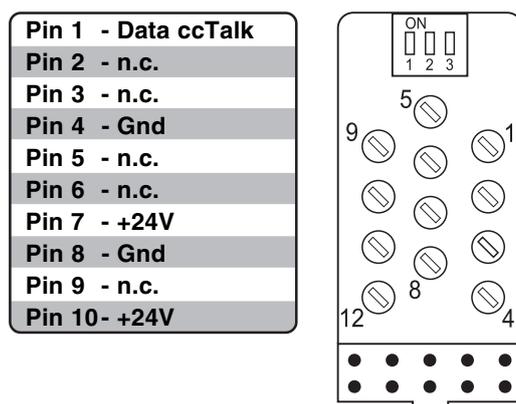
Connettore 12 Pin (versioni 12V ccTalk, 24V ccTalk)



Pin 1 e Pin 9 alimentazione
Pin 4, Pin 8 e Pin 12 indirizzo X5 (come da modalità del par. 4.2)
Pin 5 linea dati

I restanti Pin non sono collegati.

Connettore 10 Pin optional montato su scheda optional MH455a (versioni 12V ccTalk, 24V ccTalk)

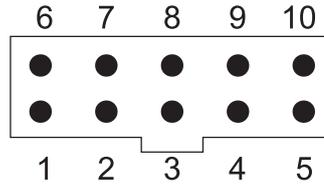


Pin 1 linea dati
Pin 4 e Pin 8 GND
Pin 7 e Pin 10 +24

I restanti Pin non sono collegati.

Connettore Flat 10 Pin (versioni 12V ccTalk, 24V ccTalk)

Pin 1-	Data ccTalk
Pin 2-	n.c.
Pin 3-	n.c.
Pin 4-	Gnd
Pin 5-	n.c.
Pin 6-	n.c.
Pin 7-	+24V
Pin 8-	Gnd
Pin 9-	n.c.
Pin 10-	+24V

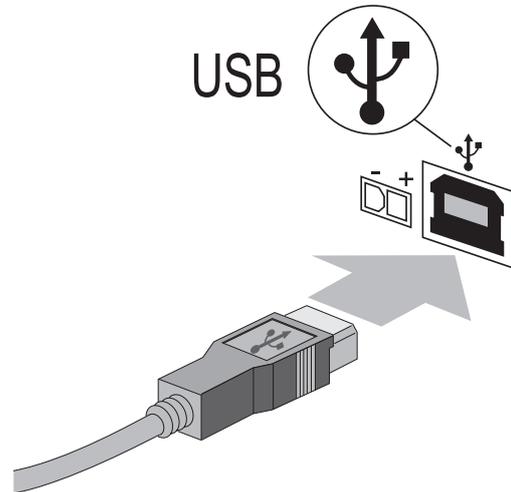


Pin 1 linea dati
Pin 4 e Pin 8 GND
Pin 7 e Pin 10 +24

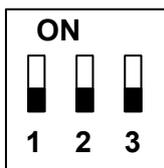
I restanti Pin non sono collegati.

Connettore USB (versione 24V ccTalk)

Alimentazione separata 24VDC - 12VDC (a richiesta)
 su connettore "mini fit" 2 poli.



L'indirizzo seriale dell' hopper può essere settato utilizzando il banco di Dip-Switch
 L'indirizzo cambia a seconda della combinazione di Dip-Switch, secondo la tabella :



Dip-Switch 1	Dip-Switch 2	Dip-Switch 3	N° hopper	Indirizzo seriale
			1	3
		X	2	4
	X		3	5
	X	X	4	6
X			5	7
X		X	6	8
X	X		7	9
X	X	X	8	10

Header 254 “Simple poll”

byte inviati: nessuno
 byte ricevuti: nessuno
 l'hopper risponde ACK se comando riconosciuto e indirizzo corretto

Header 253 “Address poll”

byte inviati: nessuno
 byte ricevuti: 1 solo byte (formato non standard) uguale all'indirizzo in uso ed inviato con un ritardo pari a 4*address mS.

Header 246 “Request manufacturer ID”

byte inviati: nessuno
 byte ricevuti: stringa che indica il costruttore. In questo caso “MicroHard.Srl.”

Header 245 “Request equipment category ID”

byte inviati: nessuno
 byte ricevuti: stringa che indica il tipo di periferica. In questo caso “Coin Inject System”

Header 244 “Request product code”

byte inviati: nessuno
 byte ricevuti: stringa che indica il tipo il codice prodotto. In questo caso “X5 DPO”

Header 242 “Request serial number”

byte inviati: nessuno
 byte ricevuti: 3 byte [byte1],[byte2],[byte3] che indicano il numero seriale della periferica. Il primo byte è sempre l'LSB quindi il numero seriale è $[byte1]+256*[byte2]+65536*[byte3]$

Header 241 “Request software revision”

byte inviati: nessuno
 byte ricevuti: stringa con le versioni soft e hardware. Per es “1.0 1.0” indica software rev 1.0 e hardware rev 1.0

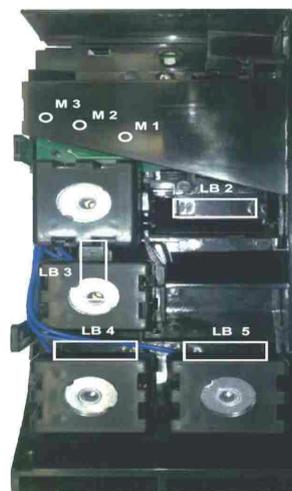
Header 217 “Request Payout Hi-Lo status”

byte inviati: nessuno
 byte ricevuti: 1 byte indicante lo stato e la presenza dei sensori di minimo e massimo
 Bit0=1 le monete NON superano il sensore di minimo (se=0 superano)

Bit1=1 le monete NON superano il sensore di massimo (se=0 superano)
 Bit2=1 sensore di minimo presente (se=0 non presente)
 Bit3=1 sensore di massimo presente (se=0 non presente)

Header 236 “Read Opto States”

byte inviati: nessuno
 byte ricevuti: 1 Byte di stato dei sensori ottici
 dove un bit=0 indica percorso libero
 Bit 0: M1
 Bit 1: M2
 Bit 2: M3 return into hopper
 Bit 3: LB5 right storage position
 Bit 4: LB4 left storage position
 Bit 5: LB2 uscita erogazione moneta
 Bit 6: LB3 riciclo moneta
 Bit 7: NU



Header 231 “Modify inhibit mask”

byte inviati: 2 byte. Ogni bit dei 2 byte abilitano o disabilitano il canale corrispondente. Il bit 0 del primo byte controlla il canale 1, il bit7 del secondo byte controlla il canale 16

byte ricevuti: Nessuno

L'hopper risponde con ACK

All'accensione ed al reset i canali sono tutti attivi. Prestare molta attenzione: se si disabilitano alcune monete erogare alcuni importi può risultare impossibile (ci sarà un timeout di pagamento)

Header 230 “Request inhibit mask”

byte inviati: Nessuno

byte ricevuti: 2. Il significato è lo stesso per i 2 byte inviati con il comando 231

Header 197 “Calculate ROM checksum”

byte inviati: nessuno

byte ricevuti: 3 byte [byte1],[byte2],[byte3] che indicano il checksum della periferica. Il primo byte è sempre l'LSB quindi il checksum è $[byte1]+256*[byte2]+65536*[byte3]$

Header 169 “Request address mode”

byte inviati: nessuno.

byte ricevuti: 1 con valore =32

Il valore 32 indica che l'indirizzo della periferica è settato via deep-sw

Header 164 “Enable Hopper”

byte inviati: 1 byte. Se byte=165 l'hopper è abilitato ai pagamenti.

byte ricevuti: Nessuno

L'hopper risponde con ACK

Header 163 “Test Hopper”

byte inviati: nessuno

byte ricevuti: 2 byte che indicano lo stato dell' hopper

Byte 1 (WARNING: l'hopper è comunque in servizio)

Bit0=1 Hopper fermato per superamento limite corrente e ripartito dopo sblocco

Bit1=1 Timeout durante erogazione (mancanza monete o motore bloccato)

Bit2=1 Motore ha effettuato manovra sblocco durante ultima erogazione

Bit3=1 Bit settato=1 al PowerOn e resettato al primo comando valido

Bit4=1 Hopper disabilitato (abilitarlo con comando 165)

Bit5=1 Hopper fermato da comando 132 (“Emergency stop value”)

Bit6= 1 Hopper in funzione / bit=0 hopper in IDLE (in attesa comandi)

Bit7= 1 Problema durante recupero moneta. Moneta non recuperata correttamente.

I bit si resettano (tranne il bit 3 e 6) ad ogni nuova abilitazione (comando 164) o al reset (comando 1)

Byte 2 (ERROR: se un solo bit=1 l'hopper è bloccato in “FUORI SERVIZIO”)

Bit0=1 Sensori induttivi monete difettosi .

Bit1=1 Checksum dati monete errati.

Bit2=1 Sensore hall difettoso.

Bit3=1 Hopper bloccato per fotocellule LB2, LB3, M1, M2, M3 ostruite.

Bit4=1 Hopper bloccato per superamento limite corrente dopo 3 tentativi.

Bit5=1 Hopper bloccato per polling-timeout durante “purge hopper”

Bit6,7= NU

Il bit di errore si resettano (e l'hopper ridiventa operativo) se le condizioni dell'errore non persistono più. I soli bit5,6 si resettano al “power on” (errori più gravi che richiedono la presenza dell'operatore).

Header 134 “Dispense hopper value”

Per erogare un importo l'hopper necessita di un parametro di 10 byte:

I primi 8 byte [byte1],...,[byte8] devono essere tutti=0

Gli ultimi 2 byte inviati (il primo è l'LSB) indicano il valore in cent da erogare che quindi vale [byte4]+256*[byte5].

INVIANDO ZERO COME VALORE SI HA IL REFILL DELLE TASCHE. L'HOPPER SI COMPORTA COME PER UNA NORMALE EROGAZIONE QUINDI OCCORRE SEGUIRE IN POLLING LO STATO TRAMITE IL COMANDO 133 IL CUI BYTE 6 INDICA QUANDO RITORNA A 0 CHE LE TASCHE SONO PIENE O CHE LA RICERCA SI E' FERMATA PER TIMEOUT MANCANZA MONETE ADATTE

byte ricevuti: nessuno. L'hopper risponde con ACK se comando corretto e precedentemente abilitato con il comando 164.

Ad ogni pagamento l'hopper viene disabilitato. Ad una richiesta di pagamento senza abilitazione l'hopper risponde con NAK.

DURANTE IL PAGAMENTO SONO AMMESSI SOLO I COMANDI 133 e 132 e 1 (gli altri ritornano un NAK). Se inviamo il reset (header=1) invece che un “Emergency stop value” (header 132) l'operazione si interrompe ugualmente ma chiaramente saranno resettati i contatori delle monete erogate

Header 133 “Request hopper polling value”

byte inviati: nessuno

byte ricevuti: 6 byte.

Il primo byte [data1] è un “Event Counter” ovvero un contatore incrementato per ogni moneta pagata (una moneta qualsiasi ma deve essere pagata e non riciclata). Viene posto=0 al reset ed al power on e riparte da 1 quando supera 255. Se si sta eseguendo un comando 134 i Byte 2 e 3 indicano il valore in cent dell'importo da erogare valore=[data2]+256*[data3].

Se invece si sta eseguendo il comando 121 “purge hopper” indicano il numero di monete non riconosciute durante lo svuotamento (ma solo durante il “purge hopper”)

Byte 4 e 5 indicano il valore in cent dell'importo erogato valore=[data4]+256*[data5]

Il byte6 indica lo status dell'hopper riguardo il pagamento.

Se=7-9 pagamento in corso (7->motore fermo 8->motore avanti 9->motore indietro)

Se=3-5 “purge hopper” in corso (3->motore fermo 4->motore avanti 5->motore indietro)

Se=0 hopper in idle e ultimo azionamento motore è stato per comando 134

Se=255 hopper in idle e ultimo azionamento motore è stato per comando “purge hopper”

Tutti i byte tranne “l'event counter” vengono resettati ad ogni nuovo comando 134.

Al power off i contatori vengono salvati, l'evento counter invece riparte da 0. Ad un comando di reset invece vengono tutti azzerati.

Header 132 “Emergency stop value”

byte inviati: nessuno

byte ricevuti: 2 byte (il primo è l'LSB) che indicano il valore in cent dell'importo residuo non pagato dopo lo stop che quindi vale $[data1]+256*[data2]$

L'hopper dopo questo comando viene anche disabilitato.

Ripetendo il comando il valore non pagato è sempre ripetuto e conservato fino al reset, allo spegnimento o ad un nuovo comando di pagamento (header 134).

NB: la risposta a seconda dello stato del percorso delle monete può essere molto ritardata quindi prevedere un timeout di almeno 500mS

Header 131 “Request hopper coin value”

byte inviati: 1 byte che indica il canale (1-16) corrispondente alla moneta di cui si vuole sapere il codice e valore.

byte ricevuti: 8 byte. I primi 6 indicano il codice moneta standard ccTalk per es “EU200” per i 2€. Gli ultimi 2 byte (il primo è l'LSB) indicano il valore in cent che quindi vale $[data7]+256*[data8]$

Header 130 “Request indexed hopper dispense count”

byte inviati: 1 byte che indica il canale (1-16) corrispondente alla moneta di cui si vuole sapere il numero di pezzi erogati.

byte ricevuti: 2 byte (il primo è l'LSB) che indicano il numero delle monete di quel canale erogate a partire dal reset. Il valore quindi vale $[data1]+256*[data2]$

Il valore dei primi 8 contatori (canali 1-8) è resettato conservato allo spegnimento mentre al comando reset sono tutti azzerati

Header 121 “Purge hopper”

L'hopper viene completamente svuotato. Anche le monete non riconosciute vengono erogate.

Questo è un comando di servizio ovvero da eseguirsi con operatore presente e potenzialmente molto pericoloso. Quindi per sicurezza occorre seguire una procedura ben precisa pena la non esecuzione del comando. In aggiunta durante lo svuotamento occorre inviare con un tempo di polling non superiore ad 1 secondo il comando 133. Se tale comando non viene inviato regolarmente l'hopper va in blocco alla prima moneta uscita dopo il timeout occorso. Occorre spegnerlo e riaccenderlo (reset hardware) per riavviarlo

Procedura:

- 1- Inviare un reset (comando 1)
- 2- Richiedere il numero di serie (comando 242)
- 3- Abilitare l'hopper (comando 164)
- 4- **ORA ED ENTRO 1 SECONDO OCCORRE INVIARE IL COMANDO SUCCESSIVO PENA LA NON ESECUZIONE DEL COMANDO**
- 5- Inviare il comando 121 con argomento i numeri di serie, l'hopper risponde con ACK.
- 6- **INVIARE IN POLLING FINO AL TIMEOUT DI MONETE FINITE IL COMANDO 133**

Dal valore del byte 6 si evince se lo svuotamento è terminato. Se tutte le monete sono state riconosciute (byte2_3 del comando 133=0) il valore può essere calcolato dai byte 3_4 del comando 133

DURANTE LO SVUOTAMENTO SONO AMMESSI SOLO I COMANDI 133 e 132 e 1 (gli altri ritornano un NAK). Se inviamo il reset (header=1) invece che un “Emergency stop value” (header 132) l'operazione si interrompe ma chiaramente saranno resettati i contatori delle monete erogate

Header 4 “Request Comms Revision”

byte inviati: nessuno

byte ricevuti: 3 byte [1],[4],[3]

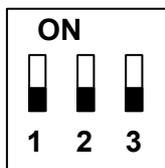
Header 1 “Reset device”

byte inviati: nessuno

byte ricevuti: nessuno

L'hopper risponde con ACK dopo aver eseguito un soft reset.

L'indirizzo seriale dell' hopper può essere settato utilizzando il banco di Dip-Switch
 L'indirizzo cambia a seconda della combinazione di Dip-Switch, secondo la tabella :



Dip-Switch 1	Dip-Switch 2	Dip-Switch 3	N° hopper	Indirizzo seriale
			1	3
		X	2	4
	X		3	5
	X	X	4	6
X			5	7
X		X	6	8
X	X		7	9
X	X	X	8	10

- il comando 134 "Dispense hopper value" va usato per contare le monete. Si invia un valore qualsiasi di pagamento e il conteggio non si ferma finchè si invia un "132-Emergency stop value" o interviene il "timeout mancanza monete".
- Per contare le monete incassate basta usare in polling il comando "133-Request hopper polling value" durante il conteggio o dopo che esso è terminato.
- Al termine del conteggio per sapere il taglio delle monete incassate usare il "130-Request indexed hopper dispense count". (NB: il conteggio dei tagli è progressivo quindi azzerabile solo all'accensione o con un reset).
- Il CCS non ha il comando "purge hopper". Il comando "134-Dispense hopper value" dato dopo un reset (quando le monete sono tutte indirizzate sul sorter 0 scarto) è in grado di "svuotare" l'intero hopper
- Possono essere programmate fino a 16 monete nei 16 canali ma solo 2 (mod ccs2) o 3 (mod ccs3) più uno scarto sono i condotti dove possono essere direzionate le monete. Monete programmate sullo stesso condotto di separazione andranno quindi a mischiarsi.

Sequenza consigliata per conteggio monete:

1-reset hopper (comando 1)

2-abilitare hopper (comando 164)

3-impostare (comando 210) il sorter 1 per i canali contenenti monete che devono andare a sinistra 2 per i canali con monete che devono andare a destra e 3 (sono per ccs3) per le monete a sinistra

4-inviare comando 134 con valore 0 se si vuole resettare i contatori delle monete o 1 se si vuole un progressivo dei contatori

5-monitorare in polling col comando 133 l'importo erogato finchè non interviene il timeout di mancanza monete (data byte6=0) o se si vuole eseguire un stop inviare il comando 132.

6-usare il comando 130 sui canali programmati per sapere il taglio delle monete incassate

NB: in caso di interruzione di alimentazione i contatori delle prime 8 monete viene salvato insieme al totale erogato

Header 254 "Simple poll"

byte inviati: nessuno

byte ricevuti: nessuno

l'hopper risponde ACK se comando riconosciuto e indirizzo corretto

Header 246 "Request manufacturer ID"

byte inviati: nessuno

byte ricevuti: stringa che indica il costruttore. In questo caso "MicroHard.Srl."

Header 245 "Request equipment category ID"

byte inviati: nessuno

byte ricevuti: stringa che indica il tipo di periferica. In questo caso "Coin Entry System"

Header 244 "Request product code"

byte inviati: nessuno

byte ricevuti: stringa che indica il tipo il codice prodotto. In questo caso "X5 CCS2"

Header 242 "Request serial number"

byte inviati: nessuno

byte ricevuti: 3 byte [byte1],[byte2],[byte3] che indicano il numero seriale della periferica. Il primo byte è sempre l'LSB quindi il numero seriale è [byte1]+256*[byte2]+65536*[byte3]

Header 241 “Request software revision”

byte inviati: nessuno

byte ricevuti: stringa con le versioni soft e hardware. Per es “1.0 1.0” indica software rev 1.0 e hardware rev 1.0

Header 236 “Read Opto States”

byte inviati: nessuno

byte ricevuti: 1 Byte di stato dei sensori ottici

dove un bit=0 indica percorso libero

Bit 0: M1

Bit 1: M2

Bit 2: M3 return into hopper

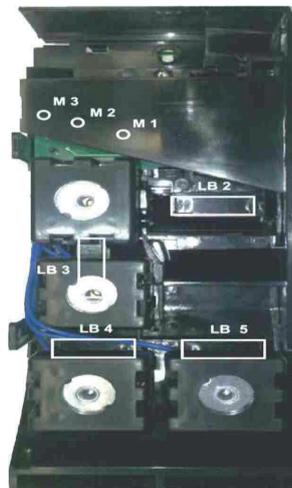
Bit 3: LB5 right storage position

Bit 4: LB4 left storage position

Bit 5: LB2 uscita erogazione moneta

Bit 6: LB3 riciclo moneta

Bit 7: NU



Header 217 “Request Payout Hi-Lo status”

byte inviati: nessuno

byte ricevuti: 1 byte indicante lo stato e la presenza dei sensori di minimo e massimo

Bit0=1 le monete NON superano il sensore di minimo (se=0 superano)

Bit1=1 le monete NON superano il sensore di massimo (se=0 superano)

Bit2=1 sensore di minimo presente (se=0 non presente)

Bit3=1 sensore di massimo presente (se=0 non presente)

Header 210 “Modify sorter path”

byte inviati: 2

byte ricevuti: Nessuno (l’hopper risponde con ACK)

Il byte 1 indica il canale 1-16 e il byte 2 indica il canale di separazione dove viene inviata la moneta. Il canale 0 è il canale di rifiuto, il canale 1 è il sinistro e il canale 2 è il destro.

Se per es vogliamo che il canale1 venga inviato al canale sinistro occorre inviare i byte 1,1.

Al reset ed al power on tutti i canali vengono indirizzati sul rifiuto. Occorre quindi impostarli come si fa per le gettoniere.

Se un canale è indirizzato sul rifiuto durante il conteggio (comando 134) il suo valore non viene conteggiato. In definitiva programmare una moneta sul canale di rifiuto equivale ad inibirla.

Header 209 “Request sorter path”

byte inviati: 1, che indica il canale di cui si vuol conoscere il canale di separazione

byte ricevuti: 1 che indica per il valore richiesto

Header 197 “Calculate ROM checksum”

byte inviati: nessuno

byte ricevuti: 3 byte [byte1],[byte2],[byte3] che indicano il checksum della periferica. Il primo byte è sempre l’LSB quindi il checksum è $[byte1]+256*[byte2]+65536*[byte3]$

Header 169 “Request address mode”

byte inviati: nessuno.

byte ricevuti: 1 con valore =32

Il valore 32 indica che l’indirizzo della periferica è settato via deep-sw

Header 164 “Enable Hopper”

byte inviati: 1 byte. Se byte=165 l’hopper è abilitato ai pagamenti.

byte ricevuti: Nessuno (l’hopper risponde con ACK)

Header 163 “Test Hopper”

byte inviati: nessuno

byte ricevuti: 2 byte che indicano lo stato dell’ hopper

Byte 1 (WARNING: l’hopper è comunque in servizio)

Bit0=1 Hopper fermato per superamento limite corrente e ripartito dopo sblocco

Bit1=1 Timeout durante erogazione (mancanza monete o motore bloccato)

Bit2=1 Motore ha effettuato manovra sblocco durante ultima erogazione

Bit3=1 Bit settato=1 al PowerOn e resettato al primo comando valido

Bit4=1 Hopper disabilitato (abilitarlo con comando 165)

Bit5=1 Hopper fermato da comando 132 (“Emergency stop value”)

Bit6= 1 Hopper in funzione / bit=0 hopper in IDLE (in attesa comandi)

Bit7= 1 Problema durante recupero moneta. Moneta non recuperata correttamente.

I bit si resettano (tranne il bit 3 e 6) ad ogni nuova abilitazione (comando 164) o al reset (comando 1)

Byte 2 (ERROR: se un solo bit=1 l'hopper è bloccato in "FUORI SERVIZIO")

Bit0=1 Sensori induttivi monete difettosi .
Bit1=1 Checksum dati monete errati.
Bit2=1 Sensore hall difettoso.
Bit3=1 Hopper bloccato per fotocelle LB2, LB3, M1, M2, M3 ostruite.
Bit4=1 Hopper bloccato per superamento limite corrente dopo 3 tentativi.
Bit5=1 Hopper bloccato per polling-timeout durante "purge hopper"
Bit6, 7= NU

Il bit di errore si resettano (e l'hopper ridiventa operativo) se le condizioni dell'errore non persistono più. I soli bit5,6 si resettano al "power on" (errori più gravi che richiedono la presenza dell'operatore).

Header 134 "Dispense hopper value"

Dopo aver abilitato l'hopper inviare il solo parametro per il conteggio
Se inviamo come parametro 0 i contatori saranno resettati e ripartono da 0, se inviamo 1 i contatori saranno progressivi.
byte ricevuti: nessuno. L'hopper risponde con ACK se comando corretto e precedentemente abilitato con il comando 164.
Ad ogni pagamento l'hopper viene disabilitato. Ad una richiesta di pagamento senza abilitazione l'hopper risponde con NAK. **DU-RANTE IL PAGAMENTO SONO AMMESSI SOLO I COMANDI 133 e 132 e 1 (gli altri ritornano un NAK). Se inviamo il reset (header=1) invece che un "Emergency stop value" (header 132) l'operazione si interrompe ugualmente ma chiaramente saranno resettati i contatori delle monete erogate**

Header 133 "Request hopper polling value"

byte inviati: nessuno
byte ricevuti: 6 byte.
Il primo byte [data1] è un "Event Counter" ovvero un contatore incrementato per ogni moneta pagata (una moneta qualsiasi ma deve essere pagata e non scartata). Viene posto=0 al reset e riparte da 1 all'overflow (0 al reset poi ciclare tra 1 e 255)
I Byte 2 e 3 indicano il valore in cent dell'importo da erogare valore=[data2]+256*[data3] **che però è sempre=0 dato che non ha alcun significato sul CCS**
I Byte 4 e 5 indicano il valore in cent dell'importo erogato valore=[data4]+256*[data5]
Il byte6 indica lo status dell'hopper riguardo il pagamento.
Se=7-9 pagamento in corso (7->motore fermo 8->motore avanti 9->motore indietro)
Se=0 hopper in idle e ultimo azionamento motore è stato per comando 134
Tutti i byte tranne "l'event counter" vengono resettati ad ogni nuovo comando 134.
Al power off i contatori vengono salvati, l'evento counter invece riparte da 0. Ad un comando di reset invece vengono tutti azzerati.

Header 132 "Emergency stop value"

byte inviati: nessuno
byte ricevuti: 2 byte (il primo è l'LSB) che indicano il valore in cent dell'importo pagato fino allo stop che quindi vale [data1]+256*[data2]
L'hopper dopo questo comando viene anche disabilitato.
Ripetendo il comando il valore è sempre ripetuto e conservato fino al reset, allo spegnimento o ad un nuovo comando di pagamento (header 134).
NB: la risposta a seconda dello stato del percorso delle monete può essere molto ritardata quindi prevedere un timeout di almeno 500ms

Header 131 "Request hopper coin value"

byte inviati: 1 byte che indica il canale (1-16) corrispondente alla moneta di cui si vuole sapere il codice e valore.
byte ricevuti: 8 byte. I primi 6 indicano il codice moneta standard ccTalk per es "EU200" per i 2€. Gli ultimi 2 byte (il primo è l'LSB) indicano il valore in cent che quindi vale [data7]+256*[data8]

Header 130 "Request indexed hopper dispense count"

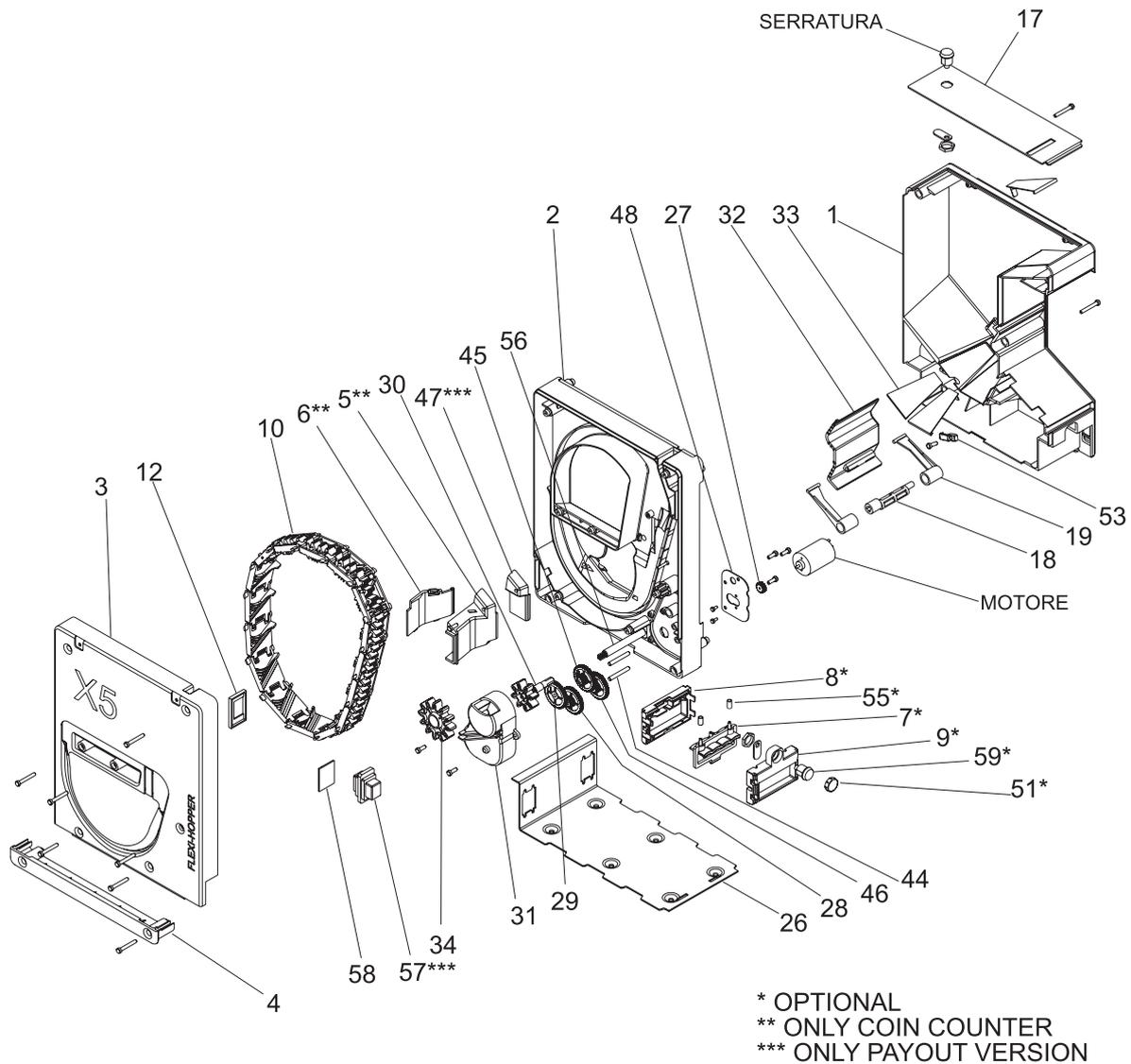
byte inviati: 1 byte che indica il canale (1-16) corrispondente alla moneta di cui si vuole sapere il numero di pezzi erogati.
byte ricevuti: 2 byte (il primo è l'LSB) che indicano il numero delle monete di quel canale erogate a partire dal reset. Il valore quindi vale [data1]+256*[data2]
Il valore dei primi 8 contatori (canali 1-8) è conservato allo spegnimento mentre al comando reset sono tutti azzerati

Header 4 "Request Comms Revision"

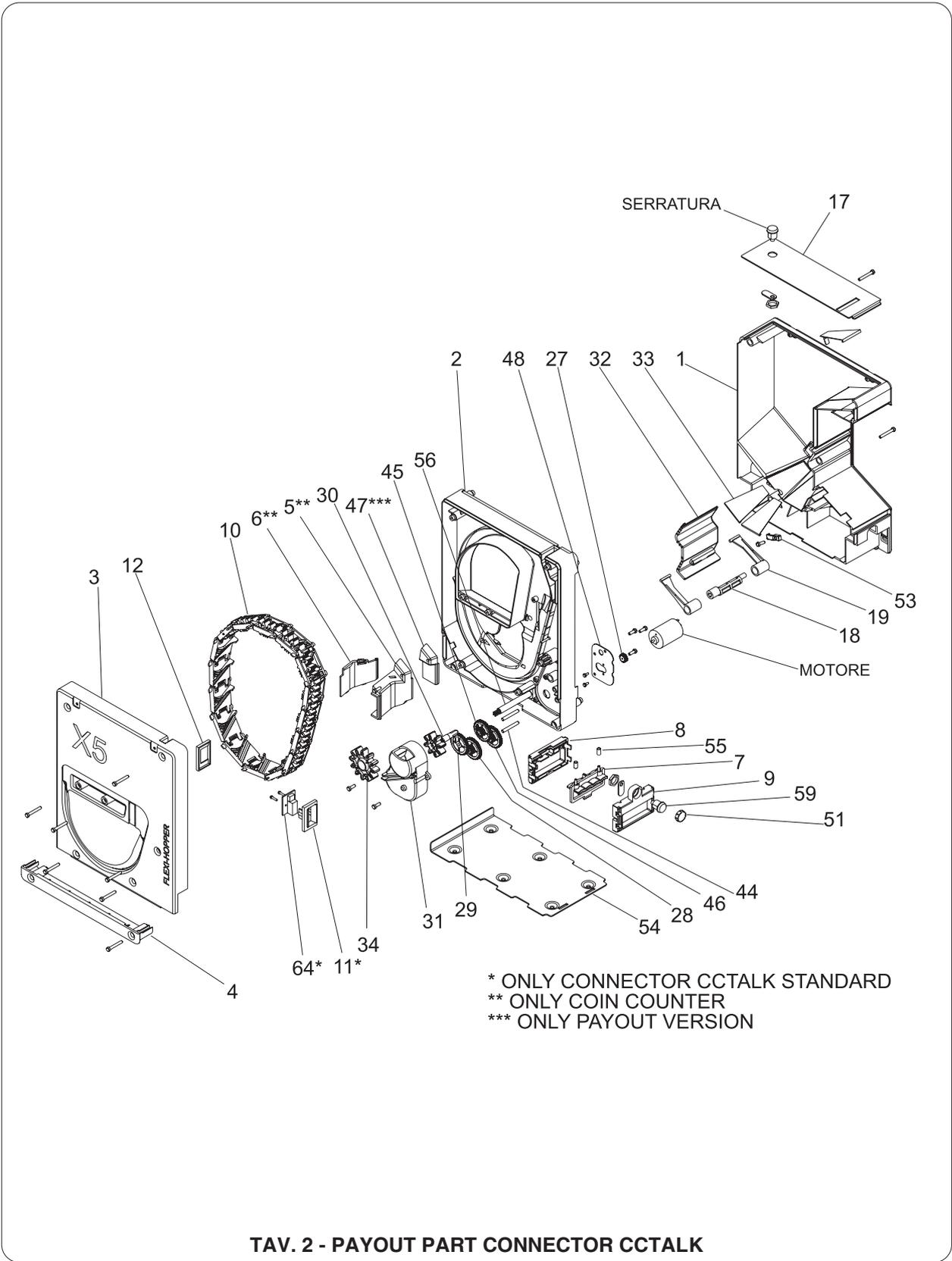
byte inviati: nessuno
byte ricevuti: 3 byte [1],[4],[3]

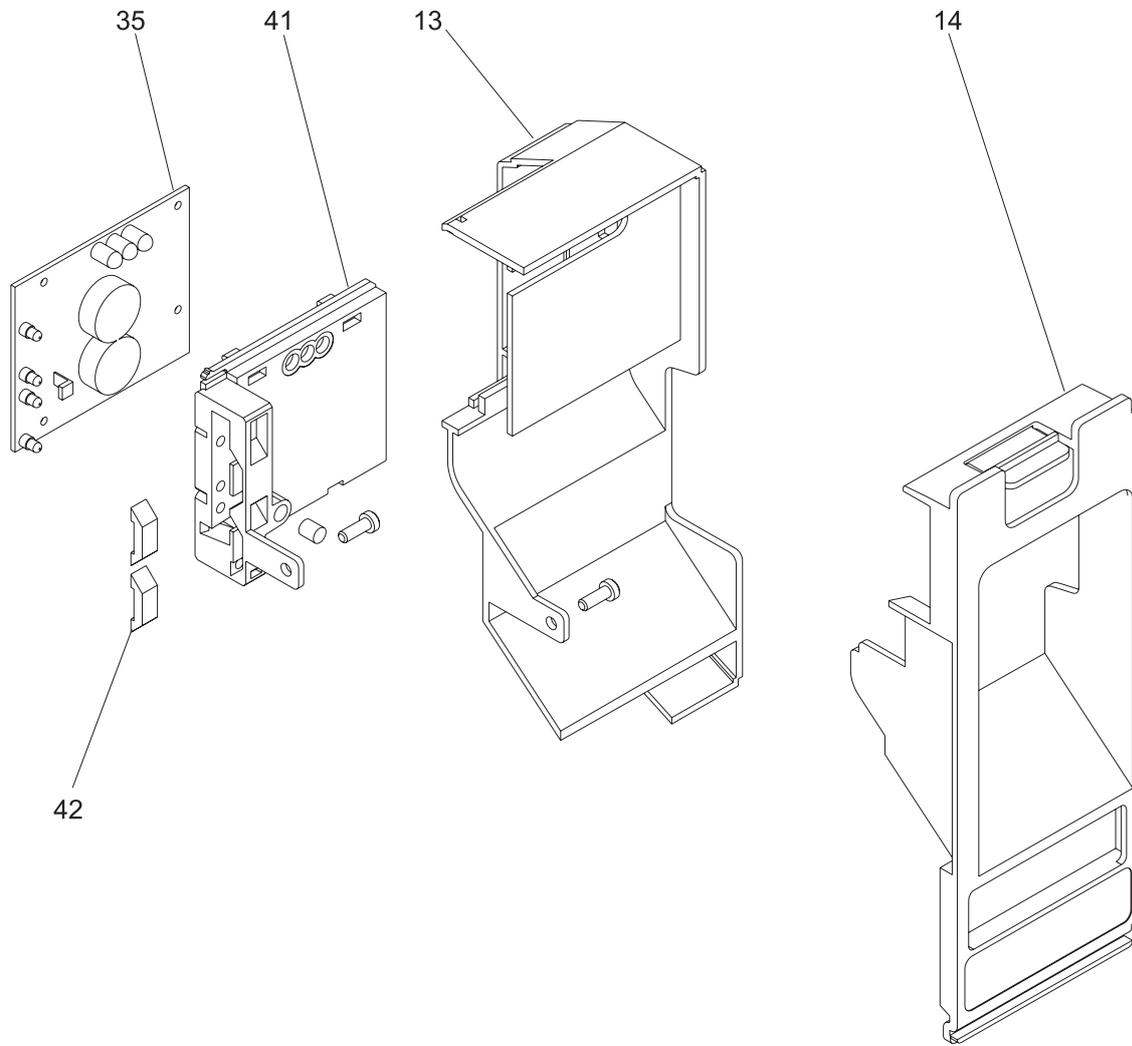
Header 1 "Reset device"

byte inviati: nessuno
byte ricevuti: nessuno
L'hopper risponde con ACK dopo aver eseguito un soft reset.

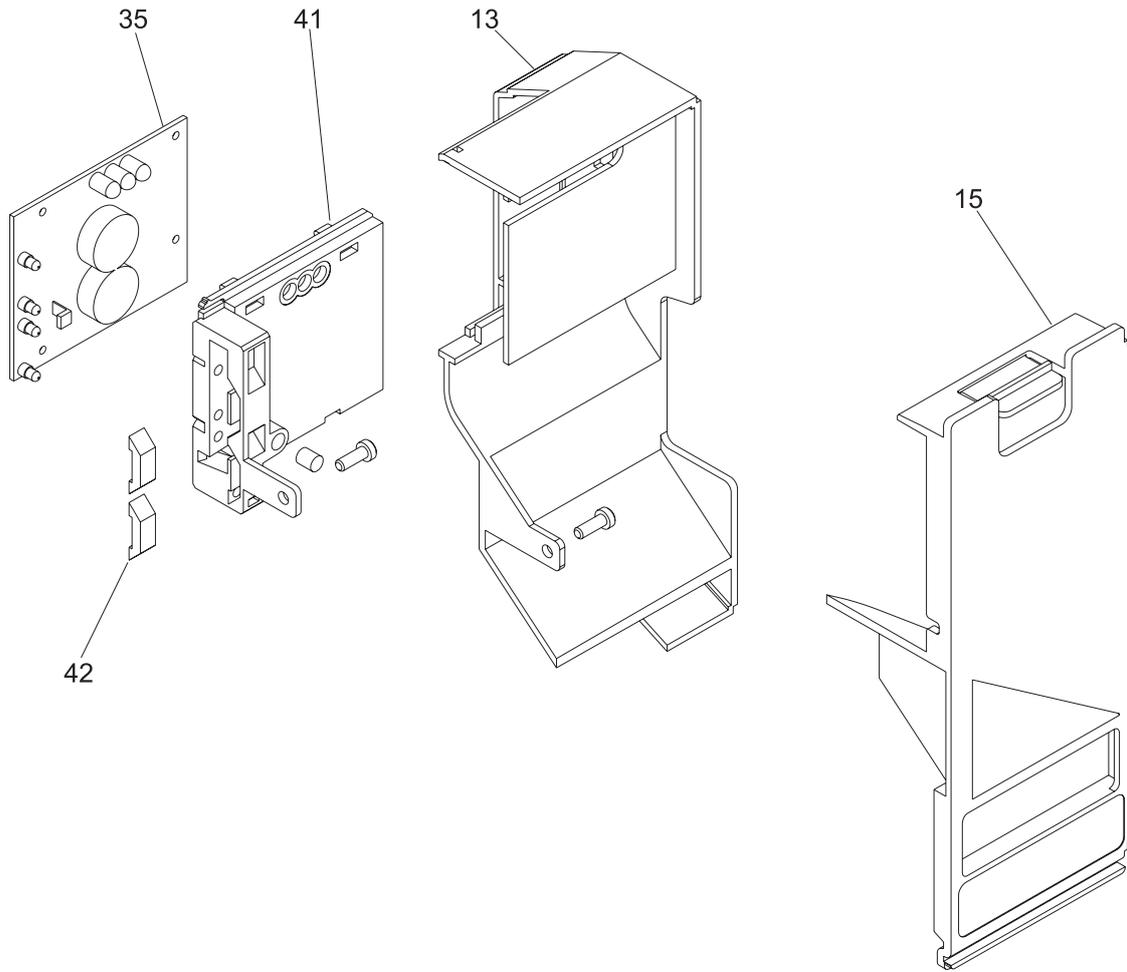


TAV. 1 - PAYOUT PART CINCH CONNECTOR





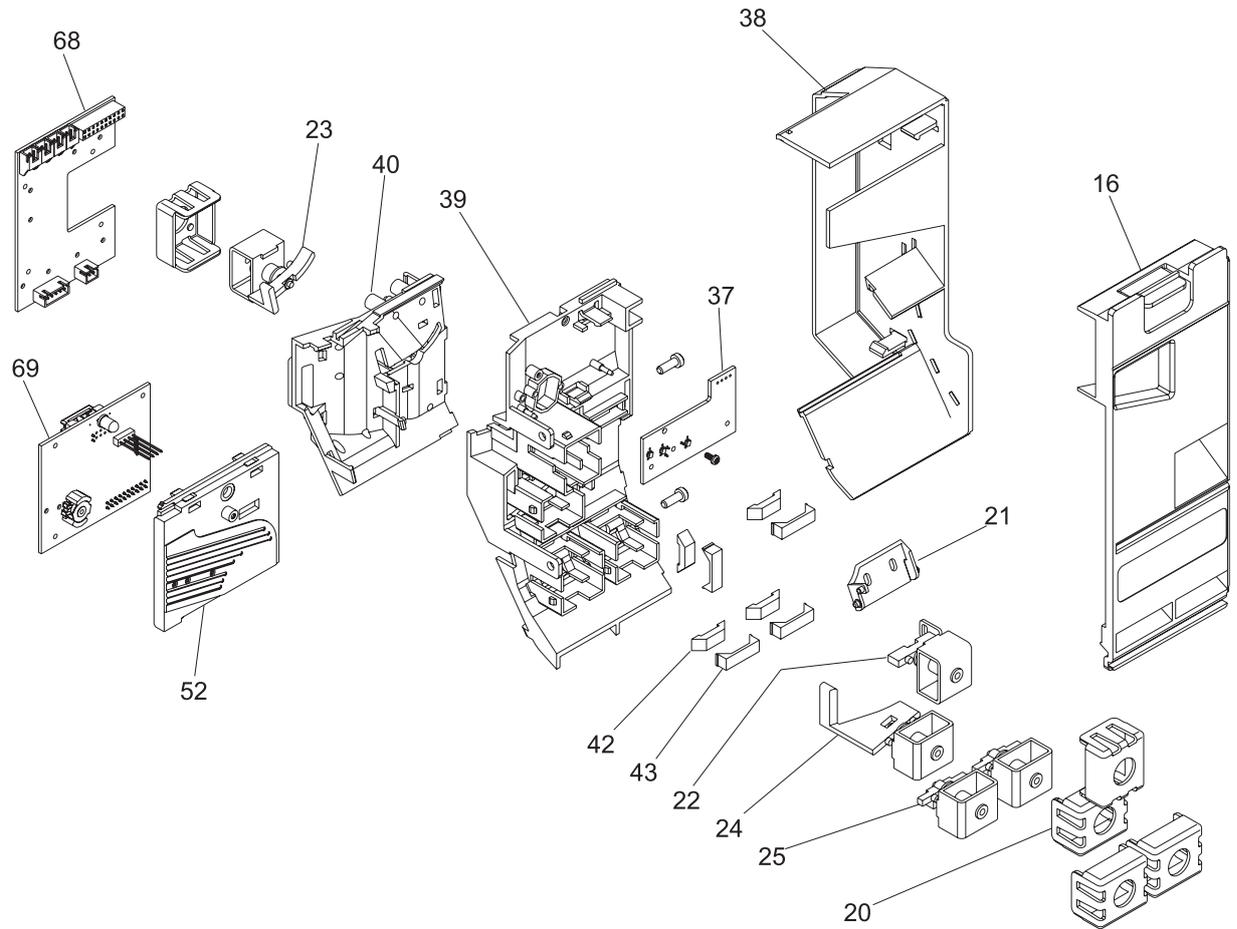
TAV. 3 - STANDART VERSION



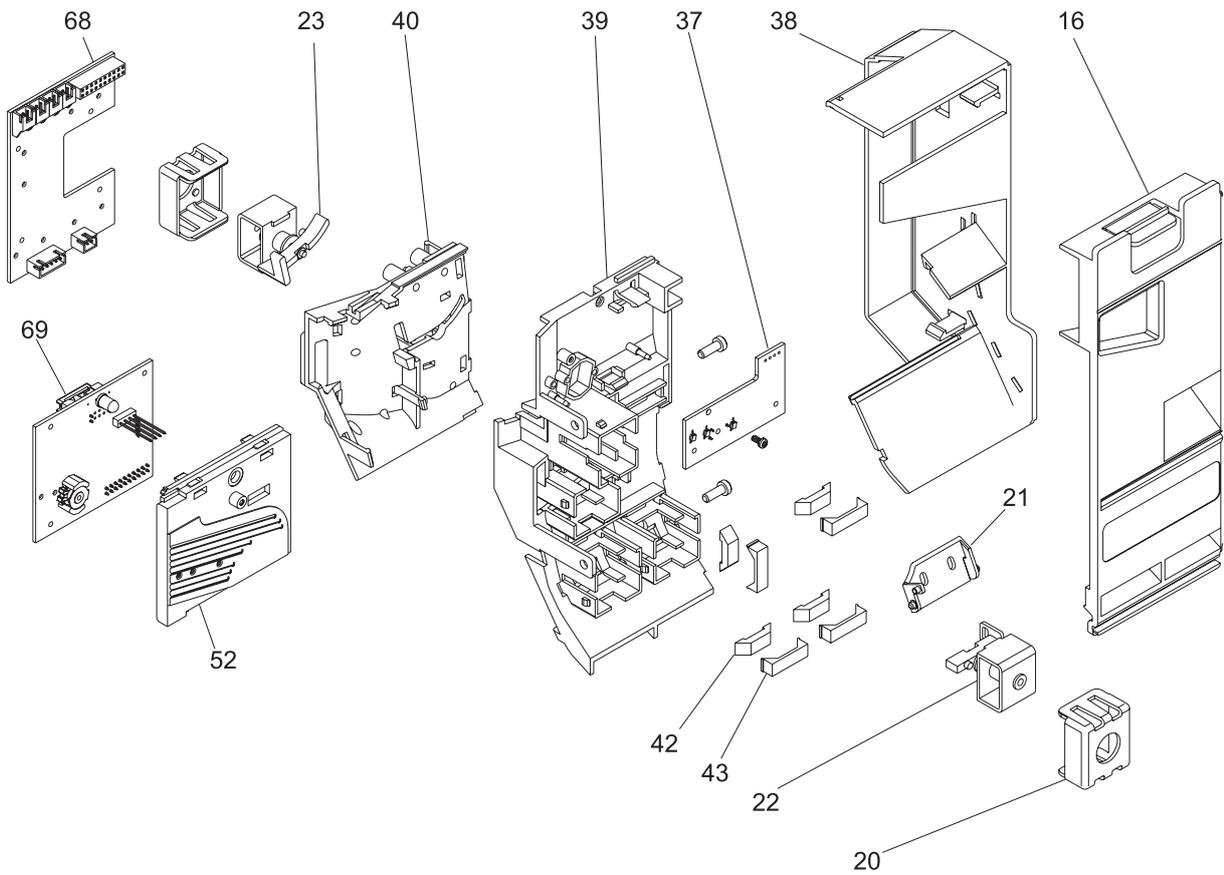
TAV. 4 - SIDE VERSION 1



TAV. 4a - SIDE VERSION 2



TAV. 5 - DYNAMIC PAY OUT



TAV.6 - COIN COUNTER AND SORTER

Pagina lasciata intenzionalmente bianca per esigenze di impaginazione



Microhard s.r.l.
Via dei Platani. 7 - 47042 Cesenatico (FC)
TEL.: 0039-0547 75450 FAX: 0039-0547 81247
info@microhard.it
www.microhard.it