

# Compact Hopper 2 seriale Manuale tecnico



Il presente documento è copyright di Money Controls Ltd. e non ne è consentita la riproduzione, né parziale né totale, con alcun mezzo, sia esso elettronico o meno, senza previa autorizzazione scritta di Money Controls Ltd. Money Controls Ltd. non accetta alcuna responsabilità relativa ad errori o omissioni contenuti nel presente documento. Money Controls Ltd. non subirà alcuna sanzione derivante dall'aderenza a questo standard, dalla sua interpretazione o dalla fiducia in esso riposta. Money Controls Ltd. fornirà pieno supporto per questo prodotto, purché sia utilizzato come descritto nel presente documento. L'utilizzo in applicazioni non trattate o al di fuori dell'ambito del presente documento potrebbe non essere supportato. Money Controls Ltd. si riserva il diritto di emendare, migliorare o modificare in qualsiasi momento il prodotto a cui si fa riferimento nel presente documento o il documento stesso.

## Sommario

<b>1. Elenco delle modifiche</b> .....	<b>4</b>
<b>2. Introduzione</b> .....	<b>5</b>
<b>3. Descrizione meccanica</b> .....	<b>5</b>
<b>4. Caratteristiche meccaniche</b> .....	<b>6</b>
4.1 Dimensioni dell'erogatore.....	6
4.2 Peso dell'erogatore.....	6
4.3 Gamma delle dimensioni delle monete.....	6
4.4 Capacità di contenimento monete.....	6
<b>5. Dimensioni generali</b> .....	<b>7</b>
<b>6. Installazione e rimozione</b> .....	<b>8</b>
6.1 Fissaggio della base dell'erogatore.....	8
6.2 Uso dei fori da 4 mm.....	8
6.3 Uso dei fori predisposti.....	8
6.4 Smontaggio dell'erogatore.....	8
6.5 Montaggio dell'erogatore.....	8
<b>7. Perché seriale?</b> .....	<b>9</b>
<b>8. Caratteristiche del prodotto</b> .....	<b>10</b>
<b>9. Parametri di progettazione ccTalk</b> .....	<b>12</b>
9.1 Tipo connettore seriale.....	13
9.11 PIN DI USCITA DEL CONNETTORE SERIALE.....	13
9.2 Tipo di connettore ausiliario.....	14
9.21 PIN DI USCITA DEL CONNETTORE AUSILIARIO.....	14
9.3 Operatività.....	14
9.4 Selezione indirizzo.....	15
9.5 Connettore del sensore di livello.....	16
<b>10. Meccanismo di criptaggio</b> .....	<b>17</b>
10.1 Meccanismo del codice PIN.....	19
<b>11. Ripristino dopo una caduta di corrente</b> .....	<b>21</b>
11.1 Interruzione di corrente: dettagli.....	21
<b>12. Che cosa accade...?</b> .....	<b>24</b>
12.1 Che cosa accade dopo l'avvio?.....	24
12.2 Che cosa accade dopo un reset dell'hardware?.....	24
12.3 Che cosa accade dopo un reset del software?.....	25
<b>13. Erogazione a moneta singola</b> .....	<b>26</b>
<b>14. Inizializzazione completa e sequenza di erogazione</b> .....	<b>27</b>
<b>15. Command List (English)</b> .....	<b>30</b>
<b>16. Elenco dei comandi (Italian)</b> .....	<b>31</b>
16.1 Dettagli dei singoli comandi.....	32
<b>17. Distribuzione dell'alimentazione in un bus a caduta multipla</b> .....	<b>46</b>
<b>18. Disturbo elettrico – Misure fisiche di protezione</b> .....	<b>47</b>
<b>19. Disturbo elettrico – Misure di protezione software</b> .....	<b>47</b>
<b>20. Specifiche elettriche</b> .....	<b>48</b>
20.1 Reset esterni.....	48
20.2 Rilascio monete.....	48
20.3 Conformità prodotto.....	49
20.31 EMISSIONI.....	49
20.32 IMMUNITÀ.....	49
20.33 SICUREZZA.....	49
20.4 Caratteristiche ambientali.....	49
<b>21. Programma di manutenzione</b> .....	<b>49</b>
21.1 Eliminazione inceppamento.....	50
<b>22. Ricerca anomalie e soluzione dei problemi</b> .....	<b>50</b>

<b>23. Appendice A</b> .....	<b>51</b>
23.1 Equazioni di conversione e valori di default.....	51
23.2 Limiti .....	51
23.3 Reset software.....	51
<b>24. Appendice B</b> .....	<b>52</b>
24.1 Parametri di sincronizzazione.....	52
24.2 Sincronizzazione sicurezza opto .....	52
24.3 Inizializzazione erogatore .....	52
<b>25. Appendice C</b> .....	<b>53</b>
25.1 Circuito di interfaccia ccTalk .....	53
<b>26. Appendice D</b> .....	<b>54</b>
26.1 Descrizione della mappa di memoria non volatile.....	54
<b>27. Appendice E</b> .....	<b>56</b>
27.1 Mk2 e Mk1 .....	56
<b>28. Appendice F</b> .....	<b>57</b>

## Figure

Figura 1: Dimensioni erogatore .....	7
Figura 2: Connettore SCH2 .....	13
Figura 3: Connettore ausiliario SCH2.....	14
Figura 4: Connettore del sensore di livello .....	16
Figura 5: Circuito di interfaccia ccTalk.....	53

## Tabelle

Tabella 1: Capacità di contenimento monete.....	6
Tabella 2: Selezione indirizzo .....	15
Tabella 3: Command List.....	30
Tabella 4: Elenco dei comandi.....	31
Tabella 5: Tabella delle azioni dei flag.....	44
Tabella 6: Specifiche elettriche.....	48
Tabella 7: Tasso di rilascio monete .....	48
Tabella 8: Requisiti ambientali.....	49
Tabella 9: Programma di manutenzione.....	49
Tabella 10: Ricerca anomalie e soluzione ai problemi.....	50
Tabella 11: Equazioni di conversione e valori di default.....	51
Tabella 12: Parametri di sincronizzazione .....	52
Tabella 13: Descrizione della mappa di memoria non volatile.....	54
Tabella 14: Differenze fra Mk2 e Mk1 .....	56

## 1. Elenco delle modifiche

Issue 1.0 – Prima edizione .....	18 agosto 2003
➤ Informazioni basate su “SCH2 Technical Manual TSP016 V2.2”.	
Issue 1.1 .....	1 dicembre 2003
➤ Added sezione <a href="#">9.5 Connettore del sensore di livello</a> .	
Issue 1.2 .....	30 giugno 2004
➤ Footer cambiato	

## 2. Introduzione

Il Serial Compact Hopper Mk2 o SCH2 è una versione a controllo seriale del classico Compact Hopper prodotto dalla Money Controls e una versione aggiornata del Serial Compact Hopper originale, ora definita dalla sigla Mk1. L'interfaccia seriale adottata è **ccTalk**, ampiamente affermata come principale protocollo di controllo a bassa velocità nel campo dell'industria delle transazioni monetarie. Una caratteristica chiave di ccTalk è il suo bilanciamento ottimale fra semplicità e sicurezza.

## 3. Descrizione meccanica

Ogni disco è dotato di una serie di fori che contengono le monete; il disco viene gestito dal motore, tramite una coppia di ruote dentate. Quando il disco ruota, la moneta posta in fondo a una delle pile verrà in contatto con il dispositivo di rilascio che si sposterà indietro consentendo, contemporaneamente, alla moneta di essere indirizzata verso la feritoia d'uscita.

Quando la moneta raggiungerà la feritoia di uscita, il dispositivo di rilascio a molla sarà in grado di ritornare nella sua posizione originale e, in questo modo, spingerà la moneta al di fuori dell'erogatore.

Il rilevatore ottico per le monete è composto da trasmettitori a infrarossi e da fotorilevatori posti nella PCB (Scheda di circuito stampato). Il raggio di luce a infrarossi viene indirizzato da una luce guida alla feritoia di uscita e, quando la moneta attraverserà la feritoia, il raggio di luce verrà interrotto generando un segnale di emissione moneta.

Vi sono a disposizione una serie di dischi, di dispositivi di rilascio e di piastre di regolazione in grado di garantire le prestazioni ottimali per monete di una particolare serie.

## 4. Caratteristiche meccaniche

### 4.1 Dimensioni dell'erogatore

Le dimensioni generali dei serbatoi, degli elementi di montaggio, dei punti di ingresso e di uscita delle monete vengono riportate in dettaglio nella [Figura 1](#). Si tenga presente che, quando si utilizza la modalità di uscita laterale, la parte posteriore del motore fuoriesce dal retro dell'alloggiamento anche se rimane all'interno delle dimensioni generali del contenitore dell'erogatore.

### 4.2 Peso dell'erogatore

570 g.

### 4.3 Gamma delle dimensioni delle monete

Gli erogatori possono essere configurati per emettere monete con una gamma di diametri che va da 16,25 a 31,00 mm e con spessori da 1,25 a 3,20 mm. Tuttavia ogni moneta dovrà essere preventivamente ed individualmente definita.

*Per ulteriori informazioni relative alla definizione preventiva delle monete si prega di contattare il dipartimento tecnico della Money Controls.*

### 4.4 Capacità di contenimento monete

La capacità di contenimento delle monete dell'erogatore va da 200 a 1000, a seconda della loro dimensione fisica.

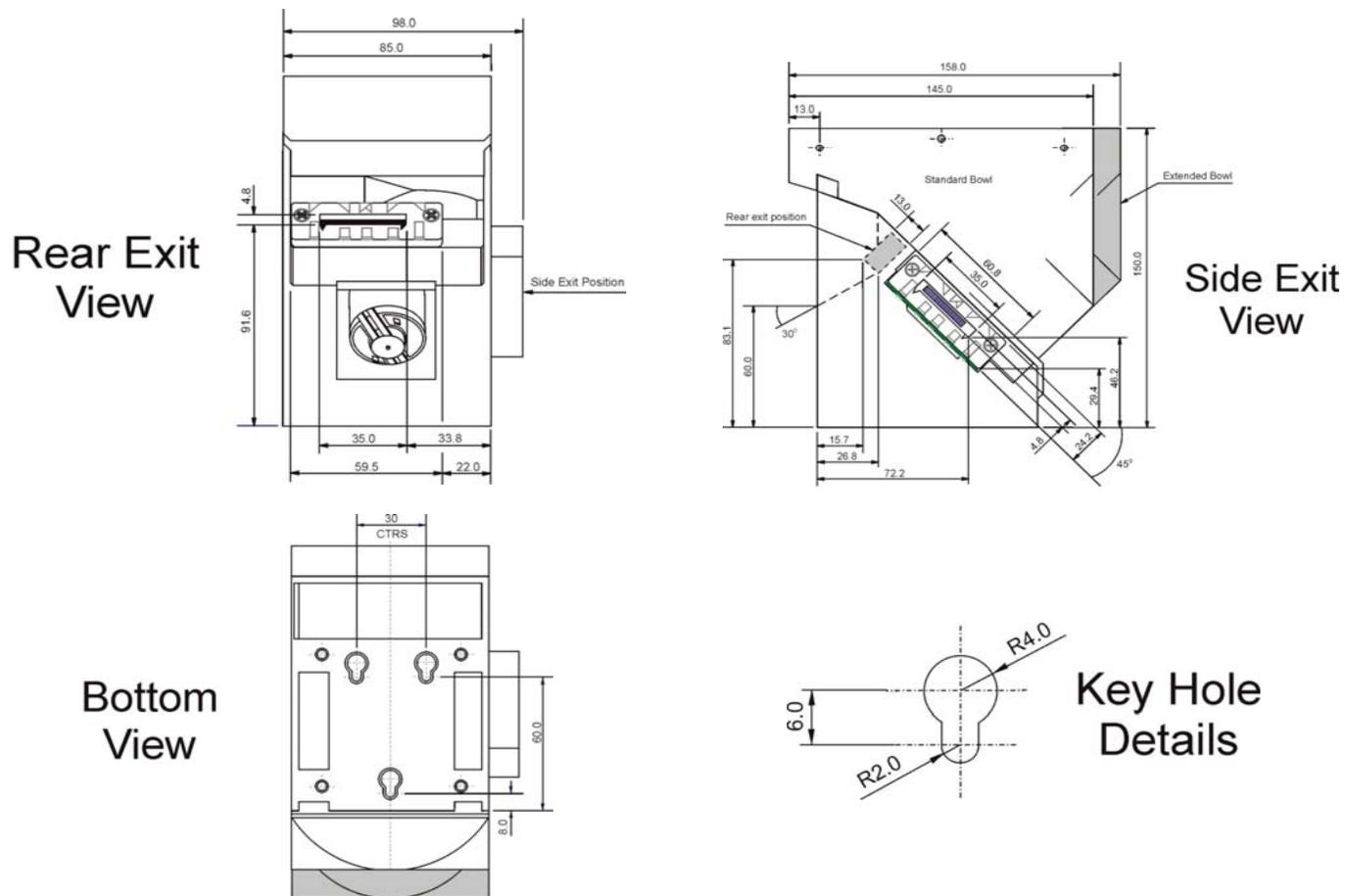
*Tabella 1: Capacità di contenimento monete*

Moneta	Diametro (mm)	Spessore (mm)	SCH2 Standard	SCH2 Esteso
£2	28,40	2,50	189	252
£1	22,50	3,10	235	314
50p (nuovo)	27,30	1,80	282	376
20p	21,40	1,80	459	613
10p	24,50	1,83	349	465
5p	18,00	1,73	679	905
2p	25,90	1,80	316	421
1p	20,25	1,43	652	870

**Nota: Queste capacità di contenimento sono soggette a un errore del +/- 10%.**

### 5. Dimensioni generali

Figura 1: Dimensioni erogatore



## 6. Installazione e rimozione

**Avvertenza: Prima della rimozione accertarsi di aver scollegato la corrente all'erogatore.**

### 6.1 Fissaggio della base dell'erogatore

- Per consentire un fissaggio adeguato all'interno della macchina host, sulla base vi sono due serie di fori di fissaggio.

### 6.2 Uso dei fori da 4 mm

- Fare 4 fori centrali
- Smontare l'erogatore
- Posizionare la base in corrispondenza dei 4 fori e fissarla utilizzando viti M3.

### 6.3 Uso dei fori predisposti

- Fare 3 fori in corrispondenza dei centri
- Allineare i 3 fori in corrispondenza dei centri
- Inserire 3 viti M3,5 SENZA stringerle
- Smontare l'erogatore
- Posizionare la base e spingerla verso il fondo per quanto possibile
- Stringere le viti per fissare la base nella posizione corretta

### 6.4 Smontaggio dell'erogatore

- Estrarre con attenzione i fermagli di sicurezza dalla parte posteriore della base
- Inclinare in avanti il contenitore per liberarlo dal blocco dei fermagli
- Fare scivolare in avanti il contenitore fino a quando le alette di posizionamento saranno libere dalla base
- Rimuovere il motore dalla base e scollegare il cavo

### 6.5 Montaggio dell'erogatore

- Collegare il cavo al motore
- Avvicinare l'unità motore alla base accertandosi che l'uscita delle monete sia nella posizione desiderata
- Localizzare le alette frontali del contenitore e inserirle nelle guide nella parte frontale della base
- Fare scorrere il contenitore con attenzione fino a quando non si sentiranno scattare i fermagli di sicurezza della base

## 7. Perché seriale?

I serbatoi per monete prevedono, di solito, una semplice interfaccia parallela. I sistemi comunemente utilizzati per l'erogazione di monete prevedono un "controllo motore logico" nel quale, per attivare o disattivare il motore, si può utilizzare un segnale di controllo a bassa tensione e un "conteggio di impulsi" nel quale, per poter emettere le monete (una moneta = un impulso), si può utilizzare un flusso di impulsi. Il metodo di "controllo motore logico" prevede che il controllo e il conteggio delle monete in transito dopo le unità opto di emissione venga gestito dal software della macchina host.

In alcune occasioni, la sicurezza di un metodo basato su un'interfaccia parallela è relativamente debole. Per svuotare il serbatoio delle monete è sufficiente forare il contenitore ed applicare al connettore o a i cavi di interfaccia la tensione appropriata. Un erogatore con un'interfaccia seriale, grazie alla complessità dell'istruzione di "rilascio monete", è in genere protetto da questo tipo di attacco.

L'interfaccia seriale fornisce un rimedio immediato a questi problemi di sicurezza in quanto, per prelevare monete dall'erogatore, necessita di un comando di rilascio **criptato** (si veda la nota più avanti) e, inoltre, esegue un conteggio delle monete emesse. A questo punto la possibilità di un attacco fraudolento esterno sono molto limitate così come è limitato anche il carico sul controller host in quanto quest'ultimo potrà eseguire altre operazioni in fase di rilascio di monete.

Un altro effetto benefico del comando seriale di rilascio multiplo di monete è rappresentato dalla possibilità di collegare più erogatori di monete allo stesso sistema di cavi o "bus". Questa operazione semplifica enormemente il cablaggio all'interno della macchina poiché si avrà la possibilità di concatenare tra di loro più serbatoi senza dover partire da un unico punto centrale di partenza. La quantità di segnali di controllo è di solito inferiore in una modalità seriale rispetto a una parallela. L'unico segnale di controllo nel protocollo ccTalk è una sola linea "dati" bidirezionale. Inoltre, in una modalità seriale si ha anche la possibilità di collegare altre periferiche per transazioni monetarie come, per esempio, unità di accettazione di monete e di banconote oppure unità di lettura di carte.

L'"espandibilità" intrinseca della modalità seriale consente un livello molto migliore di diagnostica e di rilevamento errori rispetto a quello presente in una modalità parallela (sempre che sia prevista). Invece di generare una condizione di allarme generale, si avrà la possibilità di inviare esternamente una segnalazione di un semplice inceppamento di monete oppure di un tentativo di deliberato attacco fraudolento all'erogatore.

### **Nota:**

**Vi è anche una versione Compact Hopper 2 seriale SENZA criptaggio.**

**Per ulteriori informazioni si faccia riferimento a [cctalk.org](http://cctalk.org).**

## 8. Caratteristiche del prodotto

SCH2 rappresenta, allo stato attuale delle cose, l'unità seriale di distribuzione monete più sofisticata al mondo. Per poter rispondere alle richieste dei principali produttori britannici di macchine, la Money Controls lo ha dotato di un livello di sicurezza che non trova precedenti.

L'SCH2 ha le seguenti caratteristiche.

- Sicurezza di erogazione delle monete. Il comando seriale per il rilascio delle monete utilizza una tecnica di criptaggio a 64 bit che viene modificata casualmente dopo ogni operazione. Ciò fa sì che un tentativo di "clonazione" seriale da parte di un apparato in grado di accedere al bus seriale per tentare di generare un comando di rilascio monete sia davvero un'operazione senza possibilità di successo. Un ciclo di rilascio monete prevede quindi 192 bit di informazioni casuali che transitano attraverso il bus.
- Modalità di erogazione. Il distributore di monete è, per default, impostato in modalità "pagamento multiplo" che è in grado di erogare, contemporaneamente, fino a 255 monete con un solo comando di rilascio. Un livello di sicurezza superiore si può ottenere impostando l'unità di distribuzione in modalità a "pagamento singolo", che consente il rilascio di una sola moneta alla volta.
- Sicurezza con codice PIN. Si ha la possibilità di proteggere l'unità di erogazione con un codice PIN in modo che, se l'erogatore viene rimosso dalla macchina host e collegato a un'altra, non possa essere funzionante. È l'equivalente della generazione di un blocco meccanico della macchina che può essere risolto solo dal produttore della stessa.
- Sicurezza opto. In fase di stato di attesa (non viene erogata alcuna moneta) le uscite optoisolate vengono soggette a impulsi casuali. Se in fase opto viene rilevato un blocco, oppure se in una fase non opto viene rilevato un cortocircuito, verrà generato un allarme. I blocchi sono in genere causati da inserimento di oggetti in corrispondenza dell'uscita dell'unità di erogazione mentre i cortocircuiti vengono causati da fasci di luce di torce o laser diretti alle unità ottiche.
- Protezione terminale motore. I terminali dei motori sono protetti da un relè meccanico ad alte prestazioni che evita corti circuiti quando sono in stato di attesa. Ogni tentativo di forzare un rilascio di monete tramite una batteria da 9 o da 12V collegata ai terminali del motore manderà semplicemente in corto la batteria stessa.
- Operazioni anti-incastro. Nel caso in cui l'erogatore rilevi un incastro nel corso di un'emissione di monete, per tentare di risolvere il problema verrà automaticamente attivata la funzione di funzionamento a ritroso del motore.
- Fusibile software. Nel caso in cui venga superata la soglia di corrente massima (preimpostata in fabbrica) verrà sospesa l'erogazione delle monete ed emesso un codice di errore.
- Protezione interruttore multiplo. I terminali di gestione dei motori sono protetti con un interruttore multiplo che fornisce un'ulteriore protezione contro i sovraccarichi.
- Protezione contro le cadute di alimentazione. Una memoria non volatile memorizza le monete erogate. Se nel corso di un'operazione di erogazione viene a mancare la corrente, dopo la reinizializzazione della macchina verrà letto il numero di monete residue non ancora erogate.
- Codice seriale univoco. Ogni erogatore di monete è associato a un proprio codice seriale a 24 bit che non può essere modificato da sistemi esterni.

- Conteggio delle monete. Vi sono due contatori che registrano il numero delle monete che vengono erogate. Uno può essere reimpostato dall'utente mentre l'altro è un contatore permanente. Entrambi sono implementati nella memoria non volatile.
- Integrità dei dati. Tutti i valori del contatore delle monete presenti nella memoria non volatile vengono memorizzati con un checksum a 8 bit per garantire l'integrità dei dati.
- Supporto piastra di livello. Vi è un'opzione per controllare le piastre di alto/basso livello e il loro stato può essere letto dalla macchina host in modo seriale.
- Configurazione remota. I parametri del motore quali funzionamento a ritroso e timeout di erogazione possono essere modificati con comandi seriali. Non viene richiesta alcuna modifica ai componenti elettronici.
- Operazione "erogazione multipla". Allo stesso bus seriale possono essere collegati vari erogatori seriali. Per default verranno presi in esame quelli determinati dalla sequenza di cablaggio, ma questa sequenza potrà essere modificata via software tramite un qualsiasi valore a 8 bit.
- Memoria utente. Vi sono 10 byte di memoria non volatile che potranno essere usati senza restrizioni dalla macchina host. Questa caratteristica consente di avviare diverse operazioni di sicurezza e di verifica.
- Set di comandi esteso. A seconda dell'applicazione, il software host può implementare una piccola o una gran parte del set di comandi disponibili. Sono disponibili comandi per inibire l'unità di erogazione, per leggere lo stato delle uscite optoisolate, per controllare il livello di revisione del software ecc.
- Diagnostica e rilevamento degli errori. In modalità seriale si ha pieno accesso alla diagnostica e ai codici di errore.
- Protezione codice. Il software è protetto da un circuito interno, indipendente e di controllo. Una situazione di "crash" del software genererà una semplice reimpostazione dei codici.

## 9. Parametri di progettazione ccTalk

Per una spiegazione del protocollo ccTalk e della sua implementazione su qualsiasi piattaforma, si veda la versione 4.2 di “ccTalk Serial Communication Protocol / Generic Specification”.

Questo prodotto è configurato come...

**ccTalk b96.p0.v24.a5.d0.c8.m0.x8.i1.r3**

In altre parole...

9600 baud / interfaccia a collettore aperto / alimentazione +24V / dati +5V /  
assorbimento alimentazione / connettore tipo 8 / periferica slave / checksum 8 bit /  
implementazione livello 1 / spec. versione 3

L'unità di erogazione può operare **solo** a 9600 baud.

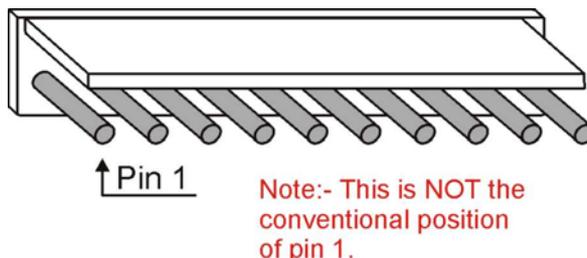
## 9.1 Tipo connettore seriale

### Connettore PCB

Passo 2,54 mm (0,1 pollice) a 10 vie con parete bloccante

N. parte: Molex 22-27-2101 o equivalente

Figura 2: Connettore SCH2



### 9.11 PINI DI USCITA DEL CONNETTORE SERIALE

Pin	Funzione
1	Seleziona indirizzo 3 - MSB
2	Seleziona indirizzo 2
3	Seleziona indirizzo 1 - LSB
4	+Vs
5	+Vs
6	0V
7	0V
8	/DATI (ccTalk)
9	N/C
10	/RESET

L'operatività sarà possibile con soli 3 fili...

- +24V al pin 4
- Terra al pin 6
- Linea dati seriali bi-direzionale al pin 8

I pin 4 e 5 e i pin 6 e 7 sono connessi internamente. La presenza di altri pin facilita la produzione di un cavo di caduta multipla utilizzando un cavo più spesso per la corrente. È anche previsto un pin di "ingresso corrente" e uno di "uscita corrente" e la possibilità di concatenamento delle unità di erogazione...

**Nota: L'unità di erogazione può operare solo a 9600 baud.**

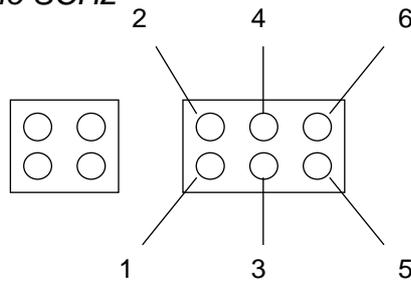
## 9.2 Tipo di connettore ausiliario

Figura 3: Connettore ausiliario SCH2

**Connettore PCB**  
Passo 2,54 mm (0,1 pollici)

N. parte: TBD

**Keying Information**



### 9.21 PIN DI USCITA DEL CONNETTORE AUSILIARIO

Pin	Funzione
1	Piastra di alto livello
3	Piastra di basso livello
5	Piastra comune
2	Link di alto livello
4	Link di basso livello
6	Link comune

## 9.3 Operatività

Per notificare all'unità di erogazione che sono stati accettati i sensori di livello della piastra, i pin dei link devono essere connessi come segue...

Modalità	Connessioni
Solo piastre di alto livello	Da pin 2 a pin 4
Solo piastre di basso livello	Da pin 4 a pin 6
Piastre di alto e basso livello	Da pin 2 a pin 4 a pin 6

In caso contrario non sarà possibile effettuare nessuna connessione.

Le piastre di livello devono esse stesse essere connesse tramite il pin di piastra corrispondente (pin 1 per l'alto livello e pin 3 per il basso livello) e tramite la piastra comune (pin 5).

## 9.4 Selezione indirizzo

L'indirizzo del bus di default di ccTalk per una periferica di erogazione è 3 e corrisponde al Compact Hopper seriale nel caso in cui sul connettore non siano state effettuate connessioni ai pin di selezione indirizzo (pin da 1 a 3).

Per le applicazioni che richiedono più di un erogatore sul bus seriale, una o più delle linee di selezione di indirizzo possono essere connesse a +Vs. In questo modo può essere generato un totale di 8 bus di indirizzo univoci, compresi nella gamma da 3 a 10.

Tabella 2: Selezione indirizzo

X = Connesso a +Vs (Pin 4, 5)			Indirizzo seriale
Selezione indirizzo 3	Selezione indirizzo 2	Selezione indirizzo 1	
			3
		X	4
	X		5
	X	X	6
X			7
X		X	8
X	X		9
X	X	X	10

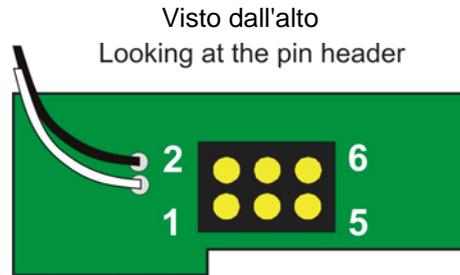
E possibile cablare singolarmente vari connettori accoppiati di un cavo bus a caduta multipla per garantire unicamente l'operatività di più erogatori. Poiché la selezione dell'indirizzo viene effettuata esternamente, sarà possibile collegare ogni Compact Hopper seriale a una qualsiasi posizione nel bus e starà alla macchina host riconoscere quale di questi dovrà erogare una particolare moneta.

La determinazione dell'indirizzo dal connettore viene eseguita solo al momento dell'accensione o in fase di reset. La modifica delle linee di selezione di indirizzo eseguita successivamente non avrà effetto.

**Nota: Gli indirizzi possono essere modificati a livello software per adattarli a valori diversi da quelli della tabella sopra riportata. Si vedano, a questo proposito, i comandi "Modifica indirizzo" e "Indirizzo casuale". Questi valori verranno persi quando si spegne la macchina oppure in fase di reimpostazione.**

## 9.5 Connettore del sensore di livello

Figura 4: Connettore del sensore di livello



Pin	Funzione
1	Piastra livello alto
3	Piastra livello basso
5	Piastra comune
2	Ponte livello alto
4	Ponte livello basso
6	Ponte comune

### Operazione

Per comunicare al software dell'hopper che i sensori della piastra di livello sono attivi, i pin del ponte devono essere collegati come segue:

Modo	Connettori
Solo piastre di livello alto	pin 2 a pin 4
Solo piastre di livello basso	pin 4 a pin 6
Piastre di livello alto e basso	pin 2 a pin 4 a pin 6

In caso contrario non c'è nessun collegamento.

Il pin delle piastre di livello deve essere collegato al pin corrispondente (pin 1 per livello alto, pin 3 per livello basso) e alla piastra comune (pin 5).

## 10. Meccanismo di criptaggio

Per rendere particolarmente difficile un tentativo fraudolento di fare erogare monete all'SCH2 viene utilizzato un meccanismo di criptaggio a 64 bit. La chiave di questo meccanismo è un algoritmo segreto, non pubblicato in questo documento, che può essere richiesto alla Money Controls dopo che saranno state espletate le necessarie procedure di approvazione.

Per mostrare la procedura per il rilascio di una moneta, viene qui fornito un esempio con un meccanismo di criptaggio “non significativo”. anche se la procedura globale è la medesima. I valori di byte racchiusi fra parentesi quadre sono riportati in formato esadecimale.

Prima di tutto è necessario avviare il generatore di numero casuale dell'erogatore inviandogli 8 byte di dati casuali...

### Comando = Pump RNG (avvia RNG)

Dati trasmessi: [ 34 ] [ A2 ] [ D7 ] [ 0F ] [ 35 ] [ 17 ] [ 55 ] [ 94 ]  
 Dati ricevuti: ACK

Non si tratta di un passo essenziale ma è utile per ampliare lo spettro delle chiavi cifrate che verranno trasmesse al bus seriale e che potrebbero essere “memorizzate” da un hacker. Poiché si suppone che la macchina host sia una macchina AWP (Amusement With Prizes) con un generatore di numeri casuali sofisticato, che va ben oltre la capacità del microcontroller dell'erogatore, potremmo comunque prendere in considerazione questa operazione. Si tenga presente che il valore generato non reimposta né “alimenta” il generatore di numeri casuali poiché ciò andrebbe a inficiare il meccanismo di sicurezza, ma si limita a rimescolarlo ulteriormente. I dettagli precisi dell'algoritmo di rimescolamento non verranno documentati.

Successivamente si dovrà richiedere una chiave cifrata...

### Comando = Request cipher key (Richiesta chiave cifrata)

Dati trasmessi: <nessuno>  
 Dati ricevuti: [ E5 ] [ 88 ] [ 13 ] [ 07 ] [ 46 ] [ FE ] [ 29 ] [ 05 ]

Prima di erogare monete dovrà essere sempre richiesta una chiave cifrata. Non vi è nessuna occasione in cui sarà possibile utilizzarne una vecchia in quanto questa chiave cambia dopo ogni comando di erogazione monete. Il comando di richiesta della chiave cifrata può essere ripetuto nel caso in cui si verifichi un errore di comunicazione e, in questo caso, la chiave cifrata, invece di essere rigenerate, verrà nuovamente trasmessa.

Ora sarà necessario combinare la chiave cifrata con il numero di monete da erogare (in questo esempio 20 monete, oppure il valore 14 esadecimale) inserendone il risultato alla fine del blocco dati.

Dati non criptati [ E5 ] [ 88 ] [ 13 ] [ 07 ] [ 46 ] [ FE ] [ 29 ] [ 05 ] [ 14 ]

In questo caso si presume che la funzione CMF (Cryptographic Mapping Function, Funzione di mappatura crittografata) sia semplicemente un'inversione di tutti i byte (nuovi dati = FF – vecchi dati).

Eseguito questo calcolo su ognuno dei byte si otterrà...

Dati criptati [ 1A ] [ 77 ] [ EC ] [ F8 ] [ B9 ] [ 01 ] [ D6 ] [ FA ] [ 14 ]

Si tenga presente che il numero di monete da erogare non è criptato ma il suo valore viene utilizzato nella funzione CMF.

Ora sarà necessario inviare il dato all'erogatore perché rilasci una moneta...

### **Comando = Dispense hopper coins (Rilascio monete dall'erogatore)**

Dati trasmessi: [ 1A ][ 77 ][ EC ][ F8 ][ B9 ][ 01 ][ D6 ][ FA ][ 14 ]  
Dati ricevuti: [ 5 ] – contatore evento di esempio

La volta successiva che verrà erogata una moneta, la chiave cifrata sarà stata modificata e, a meno che l'algoritmo non sia noto, un semplice comando di "clonatura" non avrà effetto.

In pratica la funzione CMF sarà molto più complessa di quella mostrata in questo esempio e, per essere svelata, richiederà una potenza di elaborazione considerevole e un lungo periodo di osservazione. L'invio di dati casuali all'erogatore risulterà anch'essa un'operazione inutile perché, prima di trovare una combinazione utile, passerebbe un periodo di tempo astronomico.

*La Money Controls è abbastanza realista per pensare che la funzione CMF possa cadere in mani sbagliate, sia grazie a grossi sistemi informatici sia tramite documentazione resa nota illegalmente. Per questa ragione nel nostro sistema è stata inserito un meccanismo non documentato per poter modificare la funzione CMF in caso di future necessità industriali senza, per questo, dover intervenire sull'hardware.*

Un ovvio sistema fraudolento potrebbe essere quello di intercettare e modificare in un valore più alto il numero delle monete erogate tramite il comando di rilascio monete dall'erogatore. Nell'algoritmo di sicurezza si è presa in considerazione anche questa possibilità, e il tentativo non avrebbe alcun effetto.

#### **Nota:**

**Vi è una versione del Compact Hopper 2 seriale SENZA criptaggio.**

**Per ulteriori informazioni fare riferimento a [cctalk.org](http://cctalk.org).**

## 10.1 Meccanismo del codice PIN

Come caratteristica di sicurezza aggiuntiva **opzionale** dell'SCH2, vi è la possibilità di adottare un codice PIN. Per default il prodotto viene evaso con il meccanismo per il codice PIN non abilitato. Se questa caratteristica non viene richiesta oppure se risulta essere troppo restrittiva potrà essere ignorata.

Se l'erogatore viene programmato con un codice PIN, nel caso in cui venga spento o spostato in un'altra postazione, non potrà erogare monete a meno che non si sia a conoscenza del codice PIN. Si tratta di un altro livello di difesa contro la determinazione di un hacker che voglia cercare di violare il meccanismo di criptaggio. Tuttavia l'uso del codice PIN richiede che la macchina host conservi una traccia dei codici PIN di ogni erogatore inserito nella stessa.

Vi sono varie possibilità...

### 1. Non utilizzare un codice PIN

Soluzione bella e semplice.

### 2. Impostare sempre il codice PIN allo stesso valore per tutti gli erogatori

Si può fare ma non è un sistema molto sicuro. Se si viene a conoscenza del codice PIN, allora è come se non esistesse alcuna protezione PIN per tutti gli erogatori. Semplice da gestire e sarà improbabile scordare il codice PIN adottato.

### 3. Rimescolare un codice PIN e memorizzarlo nella memoria dell'utente

Si tratta di un'idea intelligente in quanto ciò significa avere a disposizione un codice PIN casuale su ogni erogatore e, fintanto che ci si ricorderà come si è deciso di rimescolare il codice, lo si potrà sempre riportare alla sua versione originale e inviarlo all'erogatore nel corso della fase di inizializzazione. La sicurezza sta nel mantenere segreto l'algoritmo di rimescolamento.

### 4. Inserimento del codice PIN o del codice seriale

Poiché ogni erogatore è associato a un proprio numero seriale, questo potrà essere usato al posto di un codice PIN per memorizzarlo in un database centralizzato al quale possono avere accesso tutte le macchine collegate in rete. Si tratta del sistema più sicuro perché, a meno che a livello del bus non venga intercettata al momento giusto e per un determinato erogatore la transazione del codice PIN, l'unico sistema per ottenere il PIN sarebbe quello di una ricerca completa in 4,3 miliardi di combinazioni che, alla velocità di 245 ms per codice completo, richiederebbe un tempo di circa 16,7 anni.

Nel caso in cui accadesse di dimenticare il codice PIN di un determinato erogatore, sarà necessario contattare la Money Controls per avere istruzioni sui sistemi di recupero che potrebbero essere disponibili al momento dell'evento.

Continued.....

### Esempio

Ecco come è possibile impostare il codice PIN 1-2-3-4 per un nuovo erogatore che non ne avesse già uno attivo...

#### **Comando = Enter new PIN number (Inserisci nuovo codice PIN)**

Dati trasmessi: [ 1 ][ 2 ][ 3 ][ 4 ]  
Dati ricevuti: ACK

L'uso successivo di questo comando restituirà di nuovo un messaggio "ACK" ma, in effetti, non modificherà un codice PIN esistente assegnandone uno nuovo. Si tratta di un comando che funziona solo una volta.

Non appena si è definito un codice PIN, verrà sospeso il comando di erogazione monete fintanto che il codice PIN non verrà reimpresso con il comando sotto descritto.

Allo stesso modo, dopo aver acceso SCH2 con il meccanismo di codice PIN abilitato, sarà necessario immettere il codice PIN prima che le monete possano essere erogate.

#### **Comando = Enter PIN number (Inserisci codice PIN)**

Dati trasmessi: [ 1 ][ 2 ][ 3 ][ 4 ]  
Dati ricevuti: ACK

Si tenga presente che, anche nel caso in cui si inserisca un codice PIN scorretto, viene comunque emesso un messaggio "ACK". È un altro modo per aumentare il livello di sicurezza.

## 11. Ripristino dopo una caduta di corrente

SCH2 è dotato di una memoria non volatile (EEPROM) destinata alla memorizzazione dei risultati di conteggio delle monete. Pertanto, nel caso in cui nel corso di un'erogazione venisse a mancare la corrente, potranno essere rilasciate le monete ancora da erogare prima della perdita di corrente. Questa condizione viene gestita dal software della macchina host in quanto l'erogatore, di per sé, non è in grado di rilasciare automaticamente monete quando la corrente viene di nuovo riallacciata.

Vengono salvati i seguenti dati del contatore:

- [Ultima erogazione: monete rilasciate] x 1 byte
- [Ultima erogazione: monete non ancora rilasciate] x 1 byte
- [Conteggio temporaneo di rilascio dell'unità di erogazione] x 3 bytes
- [Conteggio permanente delle monete rilasciate dall'unità di erogazione] x 3 bytes insieme ai valori di checksum corrispondenti.

I byte [Ultima erogazione: monete rilasciate] e [Ultima erogazione: monete non ancora rilasciate] possono essere letti tramite il comando "Richiesta stato erogatore"

I byte [Conteggio temporaneo dell'unità di erogazione] possono essere letti con il comando "Richiesta conteggio rilascio dell'erogatore".

I byte [Conteggio permanente delle monete rilasciate dall'unità di erogazione] possono essere letti leggendo il blocco 3 della memoria non volatile servendosi del comando "Leggi blocco dati". Per una descrizione della mappa della si veda l'[Appendice D](#).

Dopo un'inizializzazione a seguito di accensione, la macchina host è in grado di leggere il byte [Ultima erogazione: monete non ancora rilasciate] per determinare se vi sono ancora alcune monete che devono essere erogate dopo che si è verificata l'interruzione di corrente. La decisione di erogare le monete rimanenti viene presa dalla macchina host e non dall'unità di erogazione.

### 11.1 Interruzione di corrente: dettagli

La sequenza per il salvataggio del contatore delle monete nella memoria non volatile è governata dalle seguenti condizioni...

- Improvvisa perdita di corrente con motore in funzione
- Ricezione di un comando di arresto di emergenza

Nel caso di un'improvvisa interruzione di corrente con motore in funzione l'unità di erogazione interromperà immediatamente l'operatività del motore e aggiornerà il contenuto della memoria non volatile fintanto che verrà rilevata tensione. L'SCH2 non è protetto da batteria tampone ma prevede solo la presenza di una riserva del condensatore.

Nel caso in cui la perdita di corrente si verifichi quando è stata completata un'operazione di erogazione (l'unità di erogazione è in stato di attesa), verrà azzerato il bit [Ultima erogazione: monete non ancora rilasciate], e ciò indipendentemente dal valore contenuto nel bit stesso. Ciò si verifica in quanto si presume che la macchina host abbia completato l'ultima sequenza di rilascio monete e abbia eseguito l'azione appropriata. Non avrebbe nessun senso assegnare un flag di monete non erogate alla successiva inizializzazione di avviamento.

Se la macchina host ha ricevuto una notifica di mancanza di corrente, potrà inviare un comando di arresto di emergenza all'unità di erogazione. Questo comando interrompe l'operatività del motore e restituisce alla macchina host l'informazione relativa al numero di

monete non erogate. Questo valore **deve essere memorizzato dalla macchina host** prima che si verifichi l'interruzione di corrente.

L'uso della linea di reset dell'hardware (pin 10 del connettore) non deve determinare l'eliminazione della procedura di rilascio monete poiché, in questo caso, non consentirebbe la memorizzazione del valore del contatore delle monete. **I contatori delle monete risulteranno scorretti nel caso in cui il pin di reset dell'hardware venga usato nel corso di una procedura di erogazione monete.**

Esempi...

a) Interruzione di corrente nel corso di una procedura di erogazione monete

Contatori	Valore iniziale	Erogate 3 su 10 e quindi interruzione di corrente
Ultima erogazione: monete rilasciate	0	3
Ultima erogazione: monete non ancora rilasciate	0	→7←
Conteggio temporaneo di rilascio dell'unità di erogazione	0	3
Conteggio permanente delle monete rilasciate dall'unità di erogazione	N	N + 3

**Monete rimanenti = 7**

b) Comando "Arresto di emergenza" emesso durante una procedura di rilascio monete

Contatore	Valore iniziale	Erogate 3 monete su 10 e quindi "Arresto di emergenza"	Ciclo spegnimento e accensione
Valore di ritorno del comando "Arresto di emergenza"	0	→7←	0
Ultima erogazione: monete rilasciate	0	3	3
Ultima erogazione: monete non ancora rilasciate	0	7	ZERO
Conteggio temporaneo di rilascio dell'unità di erogazione	0	3	3
Conteggio permanente delle monete rilasciate dall'unità di erogazione	N	N + 3	N + 3

**Monete rimanenti = 7**

c) Comando “Arresto di emergenza” inviato nel corso di una sequenza di rilascio e rilevamento di una moneta dopo che l'unità di erogazione riprende la procedura di erogazione.

Contatore	Valore iniziale	3 monete erogate su 10 e quindi “Arresto di emergenza + uscita ultima moneta	Ciclo spegnimento e accensione
Valore di ritorno del comando “Arresto di emergenza”	0	7 →6← al nuovo tentativo	0
Ultima erogazione: monete rilasciate	0	4	4
Ultima erogazione: monete non ancora rilasciate	0	→6←	<b>ZERO</b>
Conteggio temporaneo di rilascio dell'unità di erogazione	0	4	4
Conteggio permanente delle monete rilasciate dall'unità di erogazione	N	N + 3	N + 4

#### Monete rimanenti = 6

In questo caso più complicato, il conteggio temporaneo e quello permanente relativi all'unità di erogazione verranno associati a valori corretti anche se dalle unità opto viene rilevata una moneta dopo la fermata del motore. Perché l'unità host possa rilevare il valore corretto delle monete non ancora rilasciate sarà necessario che venga rinviato il comando “Arresto di emergenza” oppure utilizzare il comando “Richiesta stato erogatore” **prima** dell'interruzione di corrente – in caso contrario verrebbero rilevate 7 monete ancora da rilasciare e non 6.

Pertanto se è necessario venire a conoscenza del numero delle monete rimanenti nel corso di un'interruzione di corrente e se si desidera usare il comando “Arresto di emergenza”, accertarsi di avere il tempo sufficiente per inviare il comando stesso e un comando “Richiesta stato erogatore” prima dell'interruzione di corrente. Questa procedura garantisce la massima accuratezza possibile. La macchina host richiede almeno **100ms** per ricevere l'informazione prima che l'alimentazione scenda al di sotto del valore minimo della tensione operativa (Vtrip) (si veda l'[Appendice B](#)).

## 12. Che cosa accade...?

### 12.1 Che cosa accade dopo l'avvio?

Quella che segue vuole essere una guida a ciò che accade quando si verifica un'interruzione di corrente e un successivo riavvio.

#### **Indirizzo dispositivo**

Per default corrisponde all'indirizzo del connettore.

#### **Codice PIN**

Viene conservato ma è necessario immetterlo di nuovo.

#### **Variabili motore**

[limite corrente] = valore di default

[ritardo arresto motore] = valore di default

[timeout erogazioni] = valore di default

[corrente massima misurata] = ZERO

Per i valori di default si rimanda all'[Appendice A](#).

#### **Flag**

Fare riferimento alla tabella relativa all'azione dei flag della descrizione del comando "Verifica erogatore".

Si noti che il flag "Power-up" (Acceso) viene impostato per indicare quando l'alimentazione risulta effettivamente persa e i che il default dell'erogatore è in modalità di erogazione a più monete. L'erogatore viene avviato come inibito e necessita di essere abilitato prima di iniziare di nuovo a erogare monete.

I flag del checksum vengono aggiornati.

#### **Contatori**

Conteggio temporaneo di rilascio dell'unità di erogazione = ultimo valore

Conteggio permanente delle monete rilasciate dall'unità di erogazione = ultimo valore

#### **Richiesta stato erogatore**

[contatore evento] = ZERO

[rimanenza monete da erogare] = ZERO

[ultima erogazione: monete rilasciate] = ultimo valore

[ultima erogazione: monete non ancora rilasciate] = ZERO oppure ultimo valore se la corrente è venuta a mancare nel corso dell'operazione di rilascio monete

#### **Richiesta variabili di stato comunicazioni**

[timeout ricezione] = ZERO

[byte ricezione ignorati] = ZERO

[errato checksum ricezione] = ZERO

### 12.2 Che cosa accade dopo un reset dell'hardware?

La sola differenza fra un reset hardware e un'accensione è la seguente:

- Se il flag di accensione è stato cancellato da un reset del software allora il flag verrà mantenuto tale.

## 12.3 Che cosa accade dopo un reset del software?

Quella che segue vuole essere una guida di ciò che accade quando viene avviato un reset del software, ovvero quando all'erogatore viene inviato il comando "Reset dispositivo".

### Indirizzo dispositivo

Valori di default dell'indirizzo del connettore.

### Codice PIN

Viene conservato ma è necessario immetterlo di nuovo.

### Variabili motore

[limite corrente] = valore di default

[ritardo arresto motore] = valore di default

[timeout erogazioni] = valore di default

[corrente massima misurata] = ZERO

Per i valori di default si rimanda all'[Appendice A](#).

### Flag

Si noti che il flag "Power-up" (Acceso) viene svuotato. L'erogatore viene avviato come inibito e necessita di essere abilitato prima di iniziare a erogare di nuovo monete.

I flag del checksum vengono aggiornati.

### Contatori

Conteggio temporaneo di rilascio dell'unità di erogazione = ultimo valore

Conteggio permanente delle monete rilasciate dall'unità di erogazione = ultimo valore

### Richiesta stato erogatore

[contatore evento] = ZERO

[rimanenza monete da erogare] = ZERO

[ultima erogazione: monete rilasciate] = ultimo valore

[ultima erogazione: monete non ancora rilasciate] = ultimo valore

### Richiesta variabili di stato comunicazioni

[timeout ricezione] = ZERO

[byte ricezione ignorati] = ZERO

[errato checksum ricezione] = ZERO

### 13. Erogazione a moneta singola

Quanto segue riporta il set minimo dei comandi richiesti per erogare alcune monete dopo aver installato un erogatore nuovo di fabbrica. Si tratta di informazioni che possono essere utili quando si scrivono driver per un software o in fase di test.

Si presume che l'erogatore sia all'indirizzo 3 e che sia stato acceso.

Inviare i seguenti comandi...

- Enable hopper (Abilita erogatore)
- Request cipher key (Richiesta chiave cifrata)
- Dispense hopper coins (Rilascia monete dall'erogatore)
- Request hopper status (Richiesta stato erogatore)

In dettaglio...

I pacchetti di messaggi vengono riportati per intero (non solo la parte dati). Si presume che l'indirizzo dell'host sia 1 [valori byte] e che sia espresso in forma decimale.

#### Abilita erogatore

Dati trasmessi: [ 3 ][ 1 ][ 1 ][ 164 ][ 165 ][ 178 ]  
 Dati ricevuti: [ 1 ][ 0 ][ 3 ][ 0 ][ 252 ] = ACK

#### Richiesta chiave cifrata

Dati trasmessi: [ 3 ][ 0 ][ 1 ][ 160 ][ 92 ]  
 Dati ricevuti: [ 1 ][ 8 ][ 3 ][ 0 ][ chiave 1 ][ chiave 2 ][ chiave 3 ]  
 [ chiave 4 ][ chiave 5 ][ chiave 6 ][ chiave 7 ][ chiave 8 ]  
 [ checksum ]

I byte di dati restituiti da questo comando sono utilizzati nell'algoritmo di criptaggio. Per ragioni di sicurezza, in questo documento non verranno rivelati i dettagli dell'algoritmo di criptaggio.

#### Rilascia monete dall'erogatore

Dati trasmessi: [ 3 ][ 1 ][ 1 ][ 167 ][ sic 1 ][ sic 2 ][ sic 3 ][ sic 4 ]  
 [ sic 5 ][ sic 6 ][ sic 7 ][ sic 8 ][ N monete ]  
 [ checksum ]  
 Dati ricevuti: [ 1 ][ 1 ][ 3 ][ 0 ][ contatore eventi ][ checksum ]

Con questo comando è possibile erogare da 1 a 255 monete.

#### Richiesta stato erogatore

Dati trasmessi: [ 3 ][ 0 ][ 1 ][ 166 ][ 86 ]  
 Dati ricevuti: [ 1 ][ 4 ][ 3 ][ 0 ][ contatore eventi ]  
 [ rimanenza monete da erogare ][ ultima erogazione:  
 monete rilasciate ][ ultima erogazione: monete non  
 ancora rilasciate ][ checksum ]

Questo comando dovrebbe essere ripetuto fintanto che [rimanenza monete da erogare] = 0. Potrebbe allora venire richiesta un'altra chiave cifrata per rilasciare altre monete.

## 14. Inizializzazione completa e sequenza di erogazione

Quello che segue è un esempio di un'inizializzazione completa e di una sequenza di erogazione su un SCH2...

Nota: i comandi ccTalk vengono riportati evidenziati.

Facoltativo... **Risoluzione indirizzo**

If { determinazione rete } then

**Address poll** ⇒ risolve gli indirizzi con l'indirizzo BROADCAST

While { **Address clash** su un qualsiasi indirizzo } then

**Address random** ⇒ rimescola gli indirizzi con l'indirizzo BROADCAST

**Address poll** ⇒ risolve gli indirizzi con l'indirizzo BROADCAST

**Request equipment category id** ⇒ identifica ogni dispositivo sul bus

**Address change** ⇒ assegna, se richiesto, un nuovo indirizzo

*In altre parole, se non si sa dove si trovino nel bus gli erogatori seriali, sarà necessario localizzarli. Se uno qualsiasi degli indirizzi di dispositivo entra in conflitto allora sarà necessario rimescolarli e avviare una nuova ricerca. Tuttavia, poiché il cablaggio dell'erogatore dovrebbe inizializzare gli indirizzi di dispositivo in un modo non ambiguo, questa risoluzione di indirizzo software potrebbe anche essere evitata.*

### Inizializzazione

**Simple poll** ⇒ restituisce un "ACK" (riconoscimento) per confermare che il dispositivo è collegato, acceso e che la comunicazione a 9600 baud è operativa

**Request equipment category id** ⇒ restituisce "Payout" (Erogato). In caso contrario, probabilmente, si starà cercando di fare rilasciare una moneta da un'unità di accettazione!

**Request variable set** ⇒ indirizzo connettore (posizione fisica all'interno della macchina)

**Request serial number** ⇒ memorizzato per riferimento

Facoltativo... **Dati prodotto**

**Request manufacturer id** ⇒ per esempio "Money Controls"

**Request product code** ⇒ per esempio "SCH2"

**Request software revision** ⇒ per esempio "SCH2-Vx.y"

**Request comms revision** ⇒ livello comunicazione ccTalk

**Request hopper coin** ⇒ nome moneta, se memorizzato

**Request build code** ⇒ dettagli livello sensore

### Sblocco codice PIN

**Test hopper** ⇒ controlla se il meccanismo del codice PIN è abilitato

If { PIN abilitato } then

**Enter PIN number**

else

**Enter new PIN number** ⇒ effettua una ricerca casuale e lo inserisce invece del numero seriale per un futuro riferimento

Facoltativo... **Write data block** ⇒ inserisce l'ID dell'host nella memoria dati dell'utente

*Il meccanismo del codice PIN dovrà essere abilitato nel caso in cui la macchina host abbia una qualche possibilità di memorizzarlo! Il sistema più semplice è impostare tutti i codici PIN uguali, ma non è un sistema molto sicuro. Sarebbe molto meglio generarli in modo casuale ma, in questo caso, potrebbero sorgere problemi nel caso in cui gli erogatori venissero scambiati tra le macchine host. Si potrebbe invece creare un database centrale di codici PIN associati a numeri seriali o, anche, utilizzare la sezione dati della memoria dell'erogatore per memorizzarvi un codice PIN criptato.*

### Configurazione prodotto

**Modify variable set** ⇒ imposta il limite corrente  
 Imposta ritardo di arresto motore  
 Imposta timeout di erogazione monete  
 Imposta modalità di erogazione a moneta singola

*Se lo si desidera, si possono utilizzare i valori di default e, in questo caso, questo comando non potrà essere evitato. La modalità di erogazione a moneta singola è più sicuro poiché, una volta attivata questa modalità, un attacco esterno al bus potrebbe vedere l'erogazione di una sola moneta alla volta invece che svuotare il contenuto dell'erogatore. Ciò presume, tuttavia, che siano stati violati tutti gli altri sistemi di sicurezza. Si tenga tuttavia presente che un'erogazione a moneta singola è più lenta.*

### Rimozione dell'inibizione all'erogazione

**Enable hopper** ⇒ prepara il rilascio di monete consentendo l'avvio del comando di erogazione

### Controllo di rilasci in sospeso dovuti a interruzione di corrente

**Test hopper** ⇒ controlla un errore di scrittura nella memoria non volatile o un errore di checksum

If { errore contatore } then

**Write data block** ⇒ se esiste una copia host valida allora il contatore verrà corretto

else

**Request hopper status** ⇒ controlla se vi sono monete residue dopo l'ultima operazione di erogazione. In questo caso si prenderà la decisione se erogare quelle mancanti

Facoltativo... **Request hopper dispense count** ⇒ controlla l'ultima copia host nota per verificare se è stato eseguito un rilascio monete illegale

*In condizioni operative normali non dovrebbero verificarsi errori nella memoria dell'erogatore, Tuttavia, se uno dei contatori presenta errori e se la macchina host ha una funzione di backup, sarà possibile reintegrarne il contenuto.*

### Rilascia monete

**Test hopper** ⇒ controlla se i flag di errore sono vuoti

If { flag di errore } then

< azione appropriata contro eventuali attacchi fraudolenti >

**Reset device** ⇒ svuota i flag di errore

**Enter PIN number** ⇒ se abilitato

**Modify variable set** ⇒ imposta limite corrente, ritardo di arresto, timeout e modalità a moneta singola

**Enable hopper** ⇒ riabilita l'erogatore dopo un reset

**Pump RNG** ⇒ invia all'erogatore alcuni numeri casuali

**Request cipher key** ⇒ pronto per l'algoritmo di criptaggio

**Dispense hopper coins** ⇒ eroga una o più monete

If { NAK erogazione } then

Qualche cosa non ha funzionato? – Ricercarne la causa

continua...

**Verifica erogazione**

**Request hopper status** ⇒ controlla il contatore degli eventi

If { contatore evento non incrementato } then

Riprova **Dispense hopper coins**

While { monete da erogare > 0 }

**Request hopper status**

**Test hopper** ⇒ controlla se i flag errore sono vuoti

If { serie flag di errore } then

Risoluzione delle seguenti condizione di errore...

<inceppamento moneta> nel caso di superamento di corrente max.

<erogatore vuoto> nel caso di timeout di rilascio

<tentativo di frode opto> nel caso di un flag di errore opto

Facoltativo... **Livello di verifica**

**Request hopper dispense count** ⇒ memorizza la copia host per una verifica successiva

**Request payout high / low status** ⇒ l'erogatore è quasi vuoto?

**Notifica preventiva di mancanza di corrente**

**Emergency stop**

**Request hopper status** ⇒ memorizza le monete non rilasciate per la successiva  
reinizializzazione in fase di accensione

Attendere la mancanza di alimentazione

## 15. Command List (English)

34 commands are supported...

*Tabella 3: Command List.*

<a href="#">Intestazione 254</a>	Simple poll
<a href="#">Intestazione 253</a>	Address poll
<a href="#">Intestazione 252</a>	Address clash
<a href="#">Intestazione 251</a>	Address change
<a href="#">Intestazione 250</a>	Address random
<a href="#">Intestazione 247</a>	Request variable set
<a href="#">Intestazione 246</a>	Request manufacturer id
<a href="#">Intestazione 245</a>	Request equipment category id
<a href="#">Intestazione 244</a>	Request product code
<a href="#">Intestazione 242</a>	Request serial number
<a href="#">Intestazione 241</a>	Request software revision
<a href="#">Intestazione 236</a>	Read opto states
<a href="#">Intestazione 219</a>	Enter new PIN number
<a href="#">Intestazione 218</a>	Enter PIN number
<a href="#">Intestazione 217</a>	Request payout high / low status
<a href="#">Intestazione 216</a>	Request data storage availability
<a href="#">Intestazione 215</a>	Read data block
<a href="#">Intestazione 214</a>	Write data block
<a href="#">Intestazione 192</a>	Request build code
<a href="#">Intestazione 172</a>	Emergency stop
<a href="#">Intestazione 171</a>	Request hopper coin
<a href="#">Intestazione 169</a>	Request address mode
<a href="#">Intestazione 168</a>	Request hopper dispense count
<a href="#">Intestazione 167</a>	Dispense hopper coins
<a href="#">Intestazione 166</a>	Request hopper status
<a href="#">Intestazione 165</a>	Modify variable set
<a href="#">Intestazione 164</a>	Enable hopper
<a href="#">Intestazione 163</a>	Test hopper
<a href="#">Intestazione 161</a>	Pump RNG
<a href="#">Intestazione 160</a>	Request cipher key
<a href="#">Intestazione 004</a>	Request comms revision
<a href="#">Intestazione 003</a>	Clear comms status variables
<a href="#">Intestazione 002</a>	Request comms status variables
<a href="#">Intestazione 001</a>	Reset device

## 16. Elenco dei comandi (Italian)

Vi sono 34 comandi disponibili ...

Tabella 4: Elenco dei comandi

<a href="#">Intestazione 254</a>	Verifica singola
<a href="#">Intestazione 253</a>	Verifica indirizzo
<a href="#">Intestazione 252</a>	Conflitto indirizzo
<a href="#">Intestazione 251</a>	Modifica indirizzo
<a href="#">Intestazione 250</a>	Indirizzo casuale
<a href="#">Intestazione 247</a>	Richiesta serie variabili
<a href="#">Intestazione 246</a>	Richiesta ID produttore
<a href="#">Intestazione 245</a>	Richiesta ID categoria macchinario
<a href="#">Intestazione 244</a>	Richiesta codice prodotto
<a href="#">Intestazione 242</a>	Richiesta numero seriale
<a href="#">Intestazione 241</a>	Richiesta revisione software
<a href="#">Intestazione 236</a>	Legge stati opto
<a href="#">Intestazione 219</a>	Inserire nuovo codice PIN
<a href="#">Intestazione 218</a>	Inserire codice PIN
<a href="#">Intestazione 217</a>	Richiesta stato erogazione alto/basso
<a href="#">Intestazione 216</a>	Richiesta disponibilità memorizzazione dati
<a href="#">Intestazione 215</a>	Legge blocco dati
<a href="#">Intestazione 214</a>	Scrive blocco dati
<a href="#">Intestazione 192</a>	Richiesta codice build
<a href="#">Intestazione 172</a>	Arresto di emergenza
<a href="#">Intestazione 171</a>	Richiesta moneta erogatore
<a href="#">Intestazione 169</a>	Richiesta modalità indirizzo
<a href="#">Intestazione 168</a>	Richiesta conteggio rilascio da erogatore
<a href="#">Intestazione 167</a>	Rilascia monete dall'erogatore
<a href="#">Intestazione 166</a>	Richiesta stato erogatore
<a href="#">Intestazione 165</a>	Modifica serie variabili
<a href="#">Intestazione 164</a>	Abilita erogatore
<a href="#">Intestazione 163</a>	Verifica erogatore
<a href="#">Intestazione 161</a>	Avvia RNG (Generatore numeri casuali)
<a href="#">Intestazione 160</a>	Richiesta chiave cifrata
<a href="#">Intestazione 004</a>	Richiesta revisione comunicazioni
<a href="#">Intestazione 003</a>	Elimina variabili stato comunicazioni
<a href="#">Intestazione 002</a>	Richiesta variabili stato comunicazioni
<a href="#">Intestazione 001</a>	Reset dispositivo

## 16.1 Dettagli dei singoli comandi

Tutti i valori dei byte sono riportati in [decimale] salvo quando diversamente specificato.

### Intestazione 254: Verifica singola

Dati trasmessi: <nessuno>  
Dati ricevuti: ACK

Si tratta di un utile comando per confermare che un dispositivo è collegato all'indirizzo corretto, che è acceso e operativo. All'erogatore viene inviato un totale di 5 byte e questo risponderà con altri 5 byte.

### Intestazione 253: Verifica indirizzo

Dati trasmessi: <nessuno>  
Messaggio ricevuto: {ritardo variabile} <byte indirizzo slave>

L'erogatore restituisce un singolo byte invece che un pacchetto di messaggio ccTalk completo

Per ulteriori dettagli si vedano le specifiche generiche ccTalk

### Intestazione 252: Conflitto indirizzo

Dati trasmessi: <nessuno>  
Messaggio ricevuto: {ritardo variabile} <byte indirizzo slave>

L'erogatore restituisce un singolo byte invece che un pacchetto di messaggio ccTalk completo

Per ulteriori dettagli si vedano le specifiche generiche ccTalk

### Intestazione 251: Modifica indirizzo

Dati trasmessi: [ indirizzo ]  
Dati ricevuti: ACK

L'indirizzo specificato prevale su quello determinato dal cablaggio al connettore, Il nuovo valore verrà perso in caso di interruzione di corrente o di reinizializzazione.

### Intestazione 250: Indirizzo casuale

Dati trasmessi: <nessuno>  
Dati ricevuti: ACK

L'indirizzo viene casualmente impostato a un valore compreso tra 3 e 255. L'indirizzo "broadcast" 0, l'indirizzo master di default del bus 1 e l'indirizzo dell'unità di accettazione monete di default 2 vengono automaticamente evitati. Il nuovo valore viene perso in caso di interruzione di corrente o di reinizializzazione.

**Intestazione 247: Richiesta serie variabili**

Dati trasmessi: <none>  
Dati ricevuti: [ limite corrente ] [ ritardo arresto motore ] [ timeout rilascio ]  
[ massima corrente misurata ] [ tensione fornita ]  
[ indirizzo connettore ] [ limite corrente ]

Si tratta della soglia di corrente alla quale il motore inverte il senso di marcia per risolvere un inceppamento.

Per i dettagli sulla modifica del valore di default, si veda CURLMT dell'[Appendice A](#).

**[ ritardo arresto motore ]**

Si tratta del tempo in cui è ammesso che il motore rimanga in funzione quando rileva l'ultima moneta in una sequenza di erogazione: deve essere sufficiente per consentire la completa espulsione della moneta.

Per i dettagli sulla modifica del valore di default, si veda STOPDLY dell'[Appendice A](#).

**[ timeout rilascio ]**

Corrisponde al tempo totale concesso perché ogni moneta venga rilasciata, consentendo un tempo di inversione motore in caso di inceppamento. Dopo questo periodo di tempo il motore verrà arrestato.

Per i dettagli sulla modifica del valore di default, si veda PAYTIM dell'[Appendice A](#).

**[ massima corrente misurata ]**

Misurata con le stesse unità di misura di [limite corrente]. La corrente viene messa a confronto e mediata in fase di operatività del motore e dovrebbe essere utilizzata solo come elemento guida approssimativo.

Questa misurazione può essere portata a zero tramite il comando "Reimposta dispositivo".

**[ tensione fornita ]**

L'erogatore può misurare la propria tensione di alimentazione e renderla nota alla macchina host. Quando non sono attive sequenze di rilascio monete, la tensione viene continuamente analizzata.

Per i dettagli sulla modifica del valore di default, si veda SUPVOLTS dell'[Appendice A](#).

**[ indirizzo connettore ]**

Varia da 0 a 7.

Si tratta di un numero specificato dai pin di selezione di indirizzo del connettore. L'indirizzo del dispositivo sul bus ccTalk corrisponde a tale valore aumentato di 3 nel caso in cui non sia stato successivamente modificato serialmente.

Nota: questo valore viene memorizzato solo al momento dell'accensione.

***Nota per la progettazione software:** I prodotti futuri potrebbero prevedere la restituzione di alcune informazioni aggiuntive da parte di questo comando. Per garantire la compatibilità a ritroso, si dovrà mantenere il pacchetto dati esistente (sia nell'ordine sia nel tipo) e qualsiasi altra informazione riportata alla fine dello stesso. Nel caso in cui non venga gestita una particolare caratteristica il byte rilevante verrà riportato a zero.*

**Intestazione 246: Richiesta ID produttore**

Dati trasmessi: <nessuno>  
Dati ricevuti: "Money Controls"

**Intestazione 245: Richiesta ID categoria macchinario**

Dati trasmessi: <nessuno>  
Dati ricevuti: "Payout"

**Intestazione 244: Richiesta codice prodotto**

Dati trasmessi: <nessuno>  
Dati ricevuti: "SCH2"

SCH2 = Serial Compact Hopper Mk2.

**Intestazione 242: Richiesta numero seriale**

Dati trasmessi: <none>  
Dati ricevuti: [ seriale 1 - LSB ] [ seriale 2 ] [ seriale 3 - MSB ]

Si tratta di un numero seriale binario a 24 bit.

Le unità prototipo restituiscono...

[ 78 ] [ 97 ] [ 188 ] in decimale  
[ 4E ] [ 61 ] [ BC ] in esadecimale  
= 12,345,678

Se il risultato è pari a 5.136.828, i byte saranno in ordine inverso!

Le unità di produzione vengono fornite con un numero seriale univoco, incrementale e seriale. I numeri seriali non potranno essere facilmente modificati.

**Intestazione 241: Richiesta revisione software**

Dati trasmessi: <nessuno>  
Dati ricevuti: "SCH2-Vx.y"

x, y = 0, 1, 2... dipende dal livello di revisione del software.

**Intestazione 236: Legge stati opto**

Dati trasmessi: <nessuno>  
Dati ricevuti: [ opto rilascio ]

[ opto rilascio ]

Maschera bit:

B0 – unità opto rilascio A (0 = percorso disponibile, 1 = percorso bloccato)

B1 – unità opto rilascio B

B2 – unità opto rilascio C

B3 a B6 - non usato, restituisce 0

B7 – unità opto rilascio A + B + C

L'unità opto di rilascio il cui stato viene restituito pari a 7 consiste, in realtà, in 3 percorsi opto separati – A, B e C. Se in uno di questi percorsi opto viene rilevata una moneta, allora il bit verrà impostato a 7.

Le unità opto vengono continuamente verificate in background e questo comando restituisce anche lo stato attivo.

Si noti che questo comando è stato progettato solo per scopi di test (controllo della visibilità delle moneta nella optoelettronica) e non per il conteggio delle monete nel corso di una sequenza di erogazione! Il conteggio delle monete in corso di rilascio viene gestito automaticamente dal software e viene eseguito molto più rapidamente rispetto a una verifica di tipo seriale.

#### **Intestazione 219: Inserire nuovo codice PIN**

Dati trasmessi: [ PIN1 ] [ PIN2 ] [ PIN3 ] [ PIN4 ]  
Dati ricevuti: ACK

Un erogatore nuovo di fabbrica prevede la disabilitazione del meccanismo del codice PIN. Servendosi di questo comando un produttore potrebbe successivamente programmare il codice PIN associandolo a un qualsiasi valore predefinito e, dopo di ciò, il codice non potrà essere modificato, anche nel caso in cui venga immesso correttamente. Si tratta di un meccanismo che funziona solo una volta e lo spegnimento e la successiva riaccensione della macchina non eliminerà il codice PIN.

Un codice PIN [ 0 ] [ 0 ] [ 0 ] [ 0 ] è considerato accettabile e può essere immesso.

Il meccanismo del codice PIN è in grado di bloccare solo il comando “Rilascia monete dall'erogatore” mentre tutti gli altri comandi risulteranno regolarmente operativi.

L'immissione di un nuovo codice PIN dopo che ne è stato già programmato uno restituirà comunque un messaggio di riconoscimento (“ACK”) anche se il codice PIN rimarrà inalterato.

Per verificare se è stato programmato un codice PIN si può usare il comando “Verifica erogatore” – fare riferimento al bit 7 del registro 2 dello stato dell'erogatore.

#### **Intestazione 218: Inserire codice PIN**

Dati trasmessi: [ PIN1 ] [ PIN2 ] [ PIN3 ] [ PIN4 ]  
Dati ricevuti: ACK

Se il meccanismo del codice PIN è abilitato, il comando “Rilascia monete dall'erogatore” non risulterà operativo a meno che non venga immesso il codice PIN corretto. Il codice PIN attivo verrà perso nei seguenti casi:

- Interruzione di corrente
- Reset dell'hardware
- Reset del software
- Arresto di emergenza con motore in funzione

I codici PIN errati riceveranno sempre la risposta “ACK” come se fossero ritenuti validi.

**Intestazione 217: Richiesta stato erogazione alto/basso**

Dati trasmessi: <nessuno>  
Dati ricevuti: [ stato livello ]

Questo comando restituisce lo stato del sensore di livello.

[ stato livello ]

Maschera dei bit:

Bit 0 – Stato del sensore di basso livello (1 = inferiore al regolatore di basso livello)

Bit 1 – Stato del sensore di alto livello (1 = superiore o pari al regolatore di alto livello)

Bit 2 – Non utilizzato

Bit 3 – Non utilizzato

Bit 4 – Supporto sensore di basso livello (1 = caratteristica supportata e adattata)

Bit 5 – Supporto sensore di alto livello (1 = caratteristica supportata e adattata)

Bit 6 – Non utilizzato

Bit 7 – Non utilizzato

Per l'operatore...

Bit 0 verrà impostato nel caso in cui l'erogatore sia QUASI VUOTO.

Bit 1 verrà impostato nel caso in cui l'erogatore sia QUASI PIENO.

La condizione operativa normale dell'erogatore prevede i bit 1 e 0 vuoti.

Se gli input del sensore di livello vengono letti come QUASI VUOTO e QUASI PIENO, allora ci si trova di fronte a una condizione non corretta (è solo possibile una o l'altra condizione) e i bit 0 e 1 debbono essere lasciati entrambi vuoti.

Gli input dei sensori di livello vengono collegati a un tempo Tlevdeb (vedi [Appendice B](#)) per eliminare gli effetti di cambio moneta.

**Intestazione 216: Richiesta disponibilità memorizzazione dati**

Dati trasmessi: <nessuno>  
Dati ricevuti: [ tipo memoria ] [ leggi blocchi ] [ leggi byte per blocco ]  
[ scrivi blocchi ] [ scrivi byte per blocco ]

[ tipo memoria ] = 2, permanente (uso limitato)

[ leggi blocchi ] = 4

[ leggi byte per blocco ] = 8

[ scrivi blocchi ] = 3

[ scrivi byte per blocco ] = 8

In altre parole, sono disponibili per la lettura 32 byte di memoria non volatile e 24 byte per la scrittura, con accesso a 8 byte per volta.

Per i dettagli sulla mappa di memoria si faccia riferimento all'[Appendice D](#).

**Intestazione 215: Legge blocco dati**

Dati trasmessi: [ numero blocco ]  
Dati ricevuti: [ dati 1 ] [ dati 2 ] ... [ dati 8 ]

[ numero blocco ]  
Da 0 a 3

Consente l'accesso alla memoria non volatile.

Per i dettagli sulla mappa di memoria si faccia riferimento all'[Appendice D](#).

**Intestazione 214: Scrive blocco dati**

Dati trasmessi: [ numero blocco ] [ dati 1 ] [ dati 2 ] ... [ dati 8 ]  
Dati ricevuti: ACK

[ numero blocco ]  
Da 0 a 2

Consente l'accesso per la scrittura nella memoria non volatile.

Per i dettagli sulla mappa della memoria si faccia riferimento all'[Appendice D](#).

**Intestazione 192: Richiesta codice build**

Dati trasmessi: <nessuno>  
Dati ricevuti: 8 caratteri ASCII

Il codice build viene determinato automaticamente dall'erogatore in fase di inizializzazione. Si presume che ogni opzione build non venga modificata fintanto che l'alimentazione è attiva, in caso contrario è necessario emettere un reset del software.

'Lev HiLo' per l'adattamento dei sensori di livello alto e basso  
'Lev Hi ' per l'adattamento del solo sensore di alto livello  
'Lev Lo' per l'adattamento del solo sensore di basso livello  
'Standard' per il modello standard, nessuna opzione (come per SCH1)

## Intestazione 172: Arresto di emergenza

Dati trasmessi: <nessuno>  
Dati ricevuti: [ monete rimanenti da rilasciare ]

Se questo comando viene inviato nel corso di una procedura di rilascio monete il motore viene immediatamente arrestato come una sorta di preavviso di interruzione di corrente. Il byte del contatore restituito dovrebbe indicare il numero delle monete ancora da erogare.

L'effetto sul comando "Richiesta stato erogatore" dovrebbe essere il seguente:

[ rilascio monete rimanenti ] ⇒ Impostato a ZERO  
[ ultima erogazione: monete rilasciate ] ⇒ Monete erogate prima dell'arresto  
[ ultima erogazione: monete non ancora rilasciate ] ⇒ Monete non erogate al momento dell'arresto

Il comando di arresto di emergenza produce un'azione simile a un reset di software quando il motore è in funzione. Ciò significa che l'erogatore, prima di poter rilasciare altre monete, dovrà essere reinizializzato (abilitazione codice PIN ecc.).

Quando il motore non è operativo questo comando restituisce il valore [rilascio monete rimanenti] senza avviare un reset.

## Intestazione 171: Richiesta moneta erogatore

Dati trasmessi: <nessuno>  
Dati ricevuti: 6 caratteri ASCII

Questo comando restituisce una stringa ASCII composta da 6 caratteri che vengono memorizzati nella memoria non volatile.

I nomi delle monete vengono restituiti come [ C ] [ C ] [ V ] [ V ] [ V ] [ I ] – per una spiegazione sul formato si faccia riferimento alle specifiche generiche ccTalk.

Lo stato "non programmato" del nome di una moneta può, di solito, corrispondere a 6 codici ASCII 45 (-----) o 6 codici ASCII 0.

*Nota: Non vi è ancora disponibile nessuno strumento per la produzione software per programmare il nome della moneta che verrà erogata. Le monete che possono essere rilasciate da un erogatore sono soggette alle limitazioni fisiche della componente disco/letto piuttosto che ai parametri controllati da un qualsiasi software. I produttori delle macchine potranno programmare proprie etichette di monete servendosi del comando "Scrivi blocco dati".*

**Intestazione 169: Richiesta modalità indirizzo**

Dati trasmessi: <nessuno>  
Dati ricevuti: [ modalità indirizzo ]

Il metodo di selezione dell'indirizzo viene determinato dal prodotto, e in questo caso viene sempre restituito il valore **4A hex**. L'uso di questo comando è "esclusivamente informativo" ma potrà essere utilizzato in futuro quando il mercato vedrà l'installazione di molti erogatori seriali con diversi meccanismi di selezione di indirizzo.

**[ modalità indirizzo ]**

Maschera di bit:

- B0 – L'indirizzo viene memorizzato nella ROM
- B1 – L'indirizzo viene memorizzato nella RAM
- B2 – L'indirizzo viene memorizzato nella EEPROM o nella RAM con batteria tampone
- B3 – Selezione indirizzo via connettore di interfaccia
- B4 – Selezione indirizzo via collegamenti PCB
- B5 – Selezione indirizzo via interruttore
- B6 – L'indirizzo potrà essere modificato con comandi seriali (volatile)
- B7 – L'indirizzo potrà essere modificato con comandi seriali (non volatile)

4A hex =       Indirizzo memorizzato nella RAM  
                  Selezione indirizzo via connettore di interfaccia  
                  L'indirizzo potrà essere modificato con comandi seriali (volatile)

L'indirizzo dell'erogatore è, per default, quello definito dal cavo del connettore dopo un'accensione o dopo un reset hardware/software.

**Intestazione 168: Richiesta conteggio rilascio da erogatore**

Dati trasmessi: <nessuno>  
Dati ricevuti: [ n. monete 1 - LSB ] [ n. monete 2 ] [ n. monete 3 - MSB ]

Gamma da 0 a 16.777.215.

Il contatore del rilascio monete registra il numero di monete erogate dal momento in cui il contatore è stato svuotato per l'ultima volta.

Il valore del contatore restituito è quello "reimpostabile", non quello permanente che è accessibile con il comando "Legge blocco dati". Per ulteriori dettagli sul meccanismo di reimpostazione del contatore si faccia riferimento alla mappa di memoria descritta nell'[Appendice D](#).

## Intestazione 167: Rilascia monete dall'erogatore

Dati trasmessi: [ sic 1 ] [ sic 2 ] [ sic 3 ] [ sic 4 ] [ sic 5 ]  
[ sic 6 ] [ sic 7 ] [ sic 8 ] [ N monete ]  
Dati ricevuti: [ contatore evento ]

[ sic... ] byte di sicurezza

Il comando "Rilascia monete dall'erogatore" è protetto da un sofisticato meccanismo di criptaggio. In modalità a rilascio di monete multiplo potranno essere erogate da 1 a 255 monete e il comando viene trasmesso con un codice cifrato a 64 bit. Se la chiave cifrata non è corretta non verrà rilasciata alcuna moneta.

Se [N monete] è impostato a zero allora non verrà erogata alcuna moneta ma il messaggio verrà ugualmente gestito e il contatore di evento verrà incrementato.

[ contatore evento ]

Valore del contatore di evento dopo che è stato incrementato:

Gamma normale = da 1 a 255. Zero indica che si è verificato uno stato di accensione o un reset.

*Se il pacchetto dati ricevuto ha un errore di checksum allora il contatore di evento non sarà affidabile e questo comando non potrà essere ripetuto e ciò per evitare una sovraerogazione di monete. Il contatore di evento deve essere letto con il comando "Richiesta stato erogatore" e confrontato con l'ultimo valore noto.*

Per garantire un rilascio di monete corretto dovranno essere soddisfatti i seguenti requisiti:

- Messaggio ccTalk corretto – nessun errore né nel formato né nel checksum
- Il codice PIN deve aver precedentemente "sbloccato" l'erogatore (se il meccanismo è abilitato)
- Codice cifrato corretto
- N.monete = 1 in modalità di erogazione a moneta singola
- Erogatore abilitato – si veda il comando "Abilita erogatore".
- Nessun flag di errore "Superamento max corrente assoluta"
- Nessun flag di errore "Tentativo di frode opto, percorso bloccato in fase di attesa"
- Nessun flag di errore "Tentativo di frode opto, cortocircuito in fase di attesa"
- Nessun flag di errore "Tentativo di frode opto, cortocircuito in fase di erogazione"
- Nessun flag di errore "Opto bloccato permanentemente in fase di erogazione"

Nel caso in cui venga impostato uno qualsiasi dei flag sopra descritti (verificarli con il comando "Verifica erogatore") allora verrà bloccato il rilascio delle monete. Questi flag potranno essere azzerati con il comando "Reset dispositivo".

Se il rilascio delle monete risulta bloccato per una qualsiasi delle ragioni sopra esposte (a esclusione di un errore di comunicazione di basso livello, che non genera alcuna risposta) verrà emesso un **messaggio di non riconoscimento (NAK)** di ccTalk. Il contatore di evento verrà comunque incrementato e **l'erogatore genererà la chiave cifrata successiva.**

La ragione della mancata erogazione non viene deliberatamente comunicata per ragioni di sicurezza.

## Intestazione 166: Richiesta stato erogatore

Dati trasmessi: <nessuno>  
Dati ricevuti: [ contatore evento ] [ rilascio monete rimanenti ]  
[ ultima erogazione: monete rilasciate ]  
[ ultima erogazione: monete non ancora rilasciate ]

### [ contatore evento ]

Ogni comando “Rilascia monete dall’erogatore” corretto incrementa il contatore evento. “Valid” significa che nel pacchetto del comando non si sono verificati errori di comunicazione.

Il contatore evento, al momento dell’accensione o in fase di reset può avere solo il valore 0 – se il suo valore è 255 e viene ricevuto un altro comando di rilascio monete allora il contatore evento viene modificato in 1, quindi in 2, in 3 ecc.

Il contatore evento viene incrementato per ragioni di sicurezza – se la risposta a un comando “Rilascia monete dall’erogatore” è danneggiato o mancante a causa di disturbi allora il contatore evento dovrebbe venire precedentemente verificato prima di ritrasmettere il comando per evitare una sovraerogazione di monete. Un software host correttamente scritto dovrebbe sempre prevedere il controllo del contatore evento prima di rinviare un comando di nuovo rilascio.

### [ rilascio monete rimanenti ]

Dopo un comando “Rilascia monete dall’erogatore” il contatore viene impostato al numero di monete da erogare, per **decrementarsi** a ogni rilascio di moneta, fino a raggiungere il valore zero.

### [ ultima erogazione: monete rilasciate ]

Numero delle monete erogate in base all’ultimo comando “Rilascia monete dall’erogatore”. Questo contatore si **incrementa** a mano a mano che le monete verranno rilasciate.

### [ ultima erogazione: monete non ancora rilasciate ]

Numero delle monete che non sono state erogate a seguito di un comando “Rilascia monete dall’erogatore”. In fase di erogazione monete questo contatore viene **azzerato**.

Non appena viene ricevuto un comando di rilascio monete dall’erogatore, i byte di stato vengono aggiornati come segue:

[ rilascio monete rimanenti ] ⇒ “N monete” / **decrementa** a ogni rilascio di moneta  
[ ultima erogazione: monete rilasciate ] ⇒ Impostato a ZERO / **incrementa** a ogni rilascio di moneta  
[ ultima erogazione: monete non ancora rilasciate ] ⇒ Impostato a ZERO

Quando si è conclusa una procedura di erogazione (con successo o meno) i byte di stato diventano:

[ rilascio monete rimanenti ] ⇒ Impostato a ZERO  
[ ultima erogazione: monete rilasciate ] ⇒ Valore corretto dell’ultima operazione  
[ ultima erogazione: monete non ancora rilasciate ] ⇒ Valore corretto dell’ultima operazione

**Il software host dovrebbe sempre attendere che il valore [rilascia monete rimanenti] abbia raggiunto il valore zero prima di decidere come procedere.**

### Richiesta stato erogatore: Raccomandazioni di codifica

Servendosi dei risultati del test di durata, la Money Controls può ora emettere le seguenti raccomandazioni di codifica:

Sarebbe meglio che la verifica dello stato dell'erogatore dopo un comando "Rilascia monete dall'erogatore" avvenisse ogni **200ms**. È assolutamente essenziale l'adozione di un meccanismo di riprova in modo che, se non si ottiene nessuna risposta dal comando "Richiesta stato erogatore" dopo **50ms**, il comando venga ripetuto. Il numero dei nuovi tentativi **sequenziali** ammessi prima di poter pensare a un errore hardware dovrebbe essere impostato a **10**. La ragione di questo meccanismo di nuovi tentativi è legata ai disturbi generati dal motore. Se nel software vengono adottate misure adatte allora il collegamento di comunicazione seriale dovrebbe risultare affidabile al 100% e ciò è stato confermato anche dai test della Money Controls.

#### **Intestazione 165: Modifica serie variabili**

Dati trasmessi: [ limite corrente ] [ ritardo arresto motore ]  
[ timeout erogazione ] [ modalità moneta singola ]  
Dati ricevuti: ACK

Per ulteriori dettagli fare riferimento al comando "Richiesta serie variabili" – il formato dei dati è lo stesso.

#### [ modalità moneta singola ]

0 – imposta la modalità di rilascio a monete multiple (default)

1 – imposta la modalità di rilascio a moneta singola

(un qualsiasi altro valore viene trattato come modalità di rilascio a monete multiple).

**Per ragioni di sicurezza, quando un erogatore è stato impostato per la modalità di rilascio a moneta singola, questo comando non consente di ritornare alla modalità di rilascio a monete multiple, ma viene richiesto uno spegnimento o un reset hardware/software.**

Questo comando consente la modifica di alcune variabili per il controllo del motore ma non ne viene necessariamente richiesta l'emissione prima di usare l'erogatore. I valori di default elencati nell'[Appendice A](#) sono considerati ottimali.

Le modifiche al set delle variabili sono di tipo volatile. Qualsiasi valore personalizzato verrà perso allo spegnimento o a seguito di un reset.

***Nota per la progettazione software:** I prodotti futuri potrebbero prevedere la restituzione di alcune informazioni aggiuntive da parte di questo comando. Per garantire la compatibilità a ritroso, si dovrà mantenere il pacchetto dati esistente (sia nell'ordine sia nel tipo) e qualsiasi altra informazione riportata alla fine dello stesso. Nel caso in cui non venga gestita una particolare caratteristica il byte rilevante verrà riportato a zero.*

#### **Intestazione 164: Abilita erogatore**

Dati trasmessi: [ codice abilitazione ]  
Dati ricevuti: ACK

#### [ codice abilitazione ]

165 – abilita il rilascio di monete da parte dell'erogatore

non 165 – disabilita il rilascio di monete da parte dell'erogatore

L'erogatore deve essere abilitato prima che si possa verificare una sequenza di rilascio monete. L'unico valore decimale che abilita questa funzione è 165 (A5 hex, 10100101 binario). Tutti gli altri valori disabilitano l'erogatore.

Se un erogatore viene disabilitato prima di un comando "Rilascia monete dall'erogatore" viene restituito un valore NAK (non riconoscimento periferica)

**Un comando “Reset dispositivo” o un ciclo di caduta di corrente genererà l’automatica disabilitazione dell’erogatore.**

Questo comando viene mantenuto per compatibilità con l’SCH1 anche se ha un effetto negativo sulla sicurezza.

### **Intestazione 163: Verifica erogatore**

Dati trasmessi: <nessuno>  
 Dati ricevuti: [ registro dello stato dell’erogatore ]

Questo comando restituisce vari flag di errore e operativi provenienti dall’erogatore ed equivale al comando di esecuzione di auto-controllo delle unità di accettazione monete.

#### [ registro stato erogatore 1 ]

Maschera bit:

B0 – Superamento corrente max. assoluta	1 = superata
B1 – Timeout rilascio monete	1 = verificatosi
B2 – Motore a ritroso durante l’ultima operazione di rilascio monete a causa di inceppamento	1 = motore a ritroso
B3 – Tentativo di frode opto, percorso bloccato in fase di attesa	1 = frode
B4 – Tentativo di frode opto, cortocircuito in fase di attesa	1 = frode
B5 – Opto bloccato permanentemente in fase di erogazione	1 = bloccato
B6 – Rilevamento accensione	1 = acceso
B7 – Rilascio monete disabilitato	1 = disabilitato

Il flag “Verificatosi timeout rilascio monete” viene azzerato prima di ogni operazione di erogazione.

Una volta impostato, il flag “Motore a ritroso durante l’ultima operazione di rilascio monete a causa di inceppamento” continua ad essere attivo anche nel corso delle operazioni di rilascio successive ma può essere azzerato con un reset software.

#### [ registro stato erogatore 2 ]

Maschera bit:

B0 – Tentativo di frode opto, cortocircuito in fase di attesa	1 = frode
B1 – Modalità di rilascio a moneta singola	1 = singola
B2 – Errore di checksum A	1 = errore
B3 – Errore di checksum B	1 = errore
B4 – Errore di checksum C	1 = errore
B5 – Errore di checksum D	1 = errore
B6 – Interruzione corrente in fase di scrittura nella memoria non volatile	1 = caduta
B7 – Meccanismo codice PIN	1 = abilitato

“Tentativo di frode opto, cortocircuito in fase di attesa” si verifica nel caso in cui un fascio di luce venga indirizzato alle uscite optoisolate dell’erogatore nel corso di un’operazione di rilascio monete.

Per i dettagli relativi ai tipi di checksum si faccia riferimento all’[Appendice D](#).

Tabella 5: Tabella delle azioni dei flag

Etichetta breve	Requisiti rilascio monete?	Azione in accensione o reset hardware	Azione in fase di reset software
Max. Curr.	Si	Svuotamento	Svuotamento
Timeout	-	Svuotamento	Svuotamento
Reverse	-	Svuotamento	Svuotamento
Idle block	Si	Svuotamento	Svuotamento
Idle s/c	Si	Svuotamento	Svuotamento
Block	Si	Svuotamento	Svuotamento
Power-up	-	Impostato solo in accensione	Svuotamento
Disabled	Si	Impostato	Impostato
Pay s/c	Si	Svuotamento	Svuotamento
Single	-	Svuotamento	Svuotamento
Check A	-	Calcolato	Calcolato
Check B	-	Calcolato	Calcolato
Check C	-	Calcolato	Calcolato
Check D	-	Calcolato	Calcolato
NV Fail	-	'NV Memory'	'NV Memory'
PIN	-	'NV Memory'	'NV Memory'

**Intestazione 161: Avvia RNG (Generatore numeri casuali)**

Dati trasmessi: [ casuale 1 ] [ casuale 2 ] [ casuale 3 ] [ casuale 4 ]  
[ casuale 5 ] [ casuale 6 ] [ casuale 7 ] [ casuale 8 ]

Dati ricevuti: ACK

Questo comando avvia il generatore di numeri casuali dell'erogatore associandolo a ulteriori variabili casuali in modo da rendere molto difficile la previsione del successivo numero casuale. L'uso è facoltativo ma assolutamente raccomandato quando è in gioco la sicurezza.

**Intestazione 160: Richiesta chiave cifrata**

Dati trasmessi: <nessuno>

Dati ricevuti: [ chiave 1 ] [ chiave 2 ] [ chiave 3 ] [ chiave 4 ]  
[ chiave 5 ] [ chiave 6 ] [ chiave 7 ] [ chiave 8 ]

Questo comando prevede l'uso della chiave cifrata richiesta per poter erogare monete. La chiave cifrata potrebbe essere richiesta ripetutamente nel caso di errori di comunicazione e cambierà anche in questi casi...

- comando "Rilascio monete dall'erogatore"
- comando "Avvia RNG"
- interruzione di corrente o qualsiasi altro tipo di reset

**Intestazione 004: Richiesta revisione comunicazioni**

Dati trasmessi: <nessuno>

Dati ricevuti: [ livello ccTalk ] [ revisione superiore ] [ revisione inferiore ]

[ livello ccTalk ] = 1

[ revisione superiore ] = 3

[ revisione inferiore ] = 2

In altre parole, il primo livello delle specifiche ccTalk 3.2.

**Intestazione 003: Elimina variabili stato comunicazioni**

Dati trasmessi: <nessuno>  
Dati ricevuti: ACK

Azzerare i valori dei contatori restituiti dal comando “Richiesta variabili stato comunicazioni”.

**Intestazione 002: Richiesta variabili stato comunicazioni**

Dati trasmessi: <nessuno>  
Dati ricevuti: [ timeout ricez. ] [ byte ricez. ignorati ] [ checksum ricez. errato ]

[ timeout ricez. ]

Numero dei timeout di messaggi ricevuti registrati dal processore.

Si tratta di un valore di timeout a byte intero di Trxout – si veda l'[Appendice B](#). Il protocollo ccTalk ha una struttura a pacchetto di lunghezza variabile ma il timeout dei dati può essere sempre rilevato e contato.

[ byte ricez. ignorati ]

Numero dei byte di ricezione ignorati dal processore (a causa di un overflow del buffer).

SCH2 ha un buffer di lunghezza di 15 byte per la parte dati del pacchetto – intestazione e checksum vengono memorizzati altrove. I pacchetti con più di 15 byte di dati faranno aumentare il valore del contatore di un valore pari al numero totale dei byte di dati. In questo modo l'invio di messaggi con 20 byte di dati faranno aumentare il contatore di 20. In modo simile un messaggio con 15 o meno byte di dati non influenzerà il contatore, uno di 16 lo aumenterà di 16 e un messaggio di 252 lo aumenterà di 252.

[ checksum ricez. errato ]

Numero di messaggi ricevuti dal processore con un checksum errato.

*Tutti questi contatori sono di tipo cumulativo e, raggiunto 255, vengono riportati a 0.*

Quando si verifica un nuovo driver software sarebbe opportuno verificare anche questi contatori soprattutto dopo una serie di transazioni di messaggio per confermare che tutto vada per il verso giusto.

**Intestazione 001: Reset dispositivo**

Dati trasmessi: <nessuno>  
Dati ricevuti: ACK

Si tratta del comando richiesto per un reset del software.

**Prima** dell'operazione di reset viene restituito un “ACK”.

Il software host dovrebbe consentire un ritardo dopo l'“ACK” prima di inviare il comando successivo per consentire il completamento del codice di inizializzazione dell'erogatore – si veda TSinit dell'[Appendice B](#).

Dopo un comando “Reset dispositivo” verranno azzerati vari flag di stato (si veda il comando “Verifica erogatore”).

Qualsiasi parametro di motore inviato con il comando “Modifica serie variabili” riporterà i parametri al loro valore di default. I valori di default sono riportati nell'[Appendice A](#).

## 17. Distribuzione dell'alimentazione in un bus a caduta multipla

Il bus a caduta multipla per i Compact Hopper seriali consiste in una linea di alimentazione, una di terra e una seriale. Quando al bus sono collegati più di un erogatore tutta l'alimentazione viene trasferita attraverso un singolo cavo e potrebbe verificarsi un significativo aumento del potenziale di terra.

Tenere presente le seguenti raccomandazioni:

- Rendere operativo un solo erogatore seriale alla volta. Non iniziare mai una seconda sequenza di erogazione quando una è già in esecuzione.
- Prendere in considerazione la possibilità di usare un condizionatore di segnale sul ricevitore della linea dei dati seriali come, per esempio, un filtro di frequenza o un comparatore di input di tensione con soglia media (2,5V).
- Prendere in considerazione di collegare gli erogatori a cavi separati per alleggerire il problema di potenziale di terra.
- Se si verificano ancora errori di comunicazione, si prenda in considerazione la possibilità di modificare la topologia della rete bus a caduta multipla. Una rete a stella dovrebbe distribuire l'alimentazione in un modo più uniforme rispetto a quella ad anello, ad albero o a interconnessione multipla.

## 18. Disturbo elettrico – Misure fisiche di protezione

Il protocollo ccTalk non è stato progettato per trasferimenti su lunghe distanze ma solo per il collegamento locale di varie periferiche all'interno di una macchina host. La lunghezza tipica dei cavi non dovrebbe di solito superare i pochi metri.

Vi sono vari rimedi per minimizzare gli effetti dei disturbi irradiati e condotti sul bus ccTalk:

- Utilizzo di un'alimentazione regolata e di buona qualità con rete di filtraggio. La prestazione di alimentazione dovrebbe essere sufficiente per gestire Compact Hopper seriale in caso di sovracorrente momentanea.
- Per quanto possibile, evitare di fare correre i cavi del bus a caduta multipla accanto a componenti elettrici di disturbo quali, per esempio, motori, relé, unità video, luci fluorescenti ecc. Se si rilevano problemi di sorta prendere in considerazione l'uso di cavi schermati.
- Cercare di limitare al massimo la lunghezza dei cavi.
- Accertarsi che la linea dati ccTalk sia dotata di un resistore di carico appropriato a livello del terminale host (di solito da 1K a 10K fino a +5V).
- Non posizionare troppe periferiche sul bus – tenere in considerazione gli effetti di carico di ogni circuito di interfaccia ccTalk. Il numero massimo ammesso dipende dal circuito ricetrasmittitore.

## 19. Disturbo elettrico – Misure di protezione software

C'è una grande differenza in termini di sicurezza e affidabilità fra una buona e una mediocre implementazione software di ccTalk.

Prendere attentamente in considerazione i concetti progettuali qui di seguito indicati.

Ricerca gli errori in ogni pacchetto di risposta ccTalk:

- Si è verificato un errore nella struttura dei byte di basso livello?
- Il messaggio conteneva il numero corretto di byte di dati?
- La restituzione dell'indirizzo di destinazione era corretto?
- Il valore di checksum era corretto?
- L'indirizzo sorgente della periferica era quello che ci si aspettava?
- L'intestazione restituita era pari a zero?

Se una di queste risposte ha restituito un errore, allora il comando ccTalk potrà essere trasmesso di nuovo per tutte le volte che si renderà necessario – si tratta di una caratteristica chiave di ccTalk.

Si tenga presente che il byte [checksum ricezione errato] restituito dal comando "Richiesta variabili di stato comunicazioni" è un utile elemento per rilevare i disturbi.

## 20. Specifiche elettriche

**MCL raccomanda un'alimentazione a 24V, 4A.**

Versione motore a 24 V

*Tabella 6: Specifiche elettriche*

Specifiche elettriche	
Alimentazione	+24V
Corrente operativa normale/Senza carico	0,35A
Corrente operativa normale/Carico max	0,9A
Sovratensione/Avvio e motore a ritroso	3,6A
Tasso di erogazione normale	Da 8 a 10 monete al secondo

### 20.1 Reset esterni

Segnale, basso attivo 10K pull-up a +5V

Volt input (basso)	0,6V max
Volt input (alto)	3,5V min (5,0V max)

### 20.2 Rilascio monete

*Tabella 7: Tasso di rilascio monete*

Modalità	Tasso rilascio
Erogazione a monete multiple	Circa 8-10 monete al secondo
Erogazione a moneta singola	Circa 2 monete al secondo

## 20.3 Conformità prodotto

### 20.31 EMISSIONI

Questo prodotto è conforme alle specifiche di test EMC EN50081-1 del 1992.

### 20.32 IMMUNITÀ

Questo prodotto è conforme alle specifiche di testo EMC EN50082-1 del 1997.

### 20.33 SICUREZZA

Questo prodotto è conforme alle norme di sicurezza EN60950 del 1992 + Amdt A1 e A2 del 1993 e A3 del 1995.

## 20.4 Caratteristiche ambientali

*Tabella 8: Requisiti ambientali*

<b>Temperatura operativa</b>	Da 0 a 60 <sup>0</sup> C
<b>Temperatura di stoccaggio</b>	Da -20 a 70 <sup>0</sup> C
<b>Umidità operativa</b>	Da 10 a 75% RH
<b>Umidità di stoccaggio</b>	Da 10 a 95% RH non condensante

## 21. Programma di manutenzione

*Tabella 9: Programma di manutenzione*

<b>Programma manutenzione</b>	
Da 50.000 a 100.000, a seconda del tipo di moneta	Utilizzare un detergente medio e un panno morbido. Non utilizzare solventi spray
Ogni 50.000 monete	Sostituire braccio di rilascio e molla
Ogni 1.000.000 monete	Sostituire piastra di regolazione
Vita prevista del prodotto	3 milioni di monete con programma di manutenzione

## 21.1 Eliminazione inceppamento

- Rimuovere tutte le monete dal serbatoio.
- Rimuovere il gruppo motore dalla base come descritto.
- Eliminare l'inceppamento delle monete:
  - i. Facendo ruotare manualmente il disco, prima in senso antiorario e quindi in senso orario per liberare la moneta;  
OPPURE
  - ii. Spingere indietro la moneta servendosi di un'altra moneta.
- Rimuovere qualsiasi detrito dal gruppo del disco.
- Pulire l'unità opto della finestra con un panno morbido asciutto e pulito.
- Montare di nuovo il gruppo come descritto.
- Riempire di nuovo e verificare la funzionalità dell'erogatore.

## 22. Ricerca anomalie e soluzione dei problemi

Tabella 10: Ricerca anomalie e soluzione ai problemi

Problema	Controllare	Soluzione
Le monete si incastrano	Accertarsi che l'uscita delle monete sia libera	Rimuovere il blocco dall'uscita delle monete
	Accertarsi della corretta immissione delle monete nell'erogatore	Riempire l'erogatore con le monete corrette
	Accertarsi che nell'erogatore non vi siano monete danneggiate	Rimuovere le monete danneggiate
Il motore non funziona	Fusibile	Sostituire il fusibile
	Dispositivo di protezione disinnescato	Attendere 30 secondi a macchina SPENTA
	Verificare se l'erogatore ha rilevato un'anomalia opto	Controllare i flag EEPROM
Sovraerogazione di monete	Controllare la pulizia dell'area opto/area di uscita monete	Pulire l'area opto/area uscita
Erogazione monete insufficiente	Controllare la pulizia dell'area opto/area di uscita monete	Pulire l'area opto/area uscita
	Accertarsi che l'erogatore contenga un numero adeguato di monete	Riempire i serbatoi dell'erogatore

## 23. Appendice A

### 23.1 Equazioni di conversione e valori di default

Tabella 11: Equazioni di conversione e valori di default

Etichetta	Graduazioni e unità	Valore di default del byte	Valore fisico di default
CURLMT	$N / 17,1 \text{ A}$	34	2,0A
STOPDLY	$N \text{ ms}$	0	0ms
PAYTIM	$N * 0.333 \text{ s}$	30	10s
SUPVOLTS	$0,2 + N * 0,127 \text{ Volt}$	188	24V

### 23.2 Limiti

Il valore di CURLMT non può essere impostato a un valore inferiore a 6 (corrispondente a 0,35 A) perché in questo caso l'unità non funzionerebbe correttamente. Se viene specificato un valore inferiore a 6 verrà mantenuto valido il valore originale.

Per questioni di sicurezza il valore di STOPDLY non può essere impostato a un valore superiore a 50 (50 ms). Se viene impostato un valore superiore a 50, il valore verrà automaticamente riportato a 50.

### 23.3 Reset software

Un reset software farà sì che i parametri del motore vengano riportati ai loro valori di default, e ciò indipendentemente dallo stato attivo.

## 24. Appendice B

### 24.1 Parametri di sincronizzazione

*Tabella 12: Parametri di sincronizzazione*

Vengono riportati i valori nominali

Etichetta	Descrizione	Valore	Unità
Xtal	Risonatore PIC	4,0	MHz
Brate	Tasso baud comunicazione seriale	9600	baud
Trxout	Timeout ricezione dati	25	ms
Vtrip	Soglia caduta di alimentazione (motore +24V)	16	V
TPinit	Tempo di inizializzazione accensione	600	ms
TRinit	Tempo inizializzazione reset hardware	20	ms
TSinit	Tempo di inizializzazione reset software	< 40	ms
TMRelOn	Ritardo relè motore acceso	25	ms
TMRelOff	Ritardo relè motore spento	25	ms
TMreverse	Tempo inversione motore	150	ms
Irelay	Tasso protezione relè motore (sovracorrente)	10	A
TEESave	Tempo salvataggio dati EEPROM	17	ms
Ifuse	Corrente assoluta max	5	A
Tlevdeb	Tempo eliminazione rimbalzo sensore di livello	2	s
	Ritardo risposta comunicazioni: Verifica semplice	< 2	ms
	Ritardo risposta comunicazioni: Immettere codice PIN (corretto)	< 2	ms
	Ritardo risposta comunicazioni: Immettere codice PIN (scorretto)	235	ms
	Ritardo risposta comunicazioni: Scrittura blocco dati	8	ms
	Ritardo risposta comunicazioni: Rilascio monete dall'erogatore	< 4	ms
	Ritardo risposta comunicazioni: Richiesta stato erogatore	< 2	ms
	Ritardo risposta comunicazioni: Avvia RNG	< 2	ms
	Ritardo risposta comunicazioni: Richiesta chiave cifrata	< 2	ms

### 24.2 Sincronizzazione sicurezza opto

In stato di attesa il software dell'erogatore verifica le unità opto su base pseudocasuale per rendere ancora più difficili possibili frodi. Il periodo di verifica varia da 15 a 255 ms. Quando vengono rilasciate monete, il periodo di verifica cambia e va da 64 a 127 ms.

### 24.3 Inizializzazione erogatore

Non è possibile alcuna comunicazione seriale nel corso di operazioni "TPinit", "TRinit" e "TSinit".

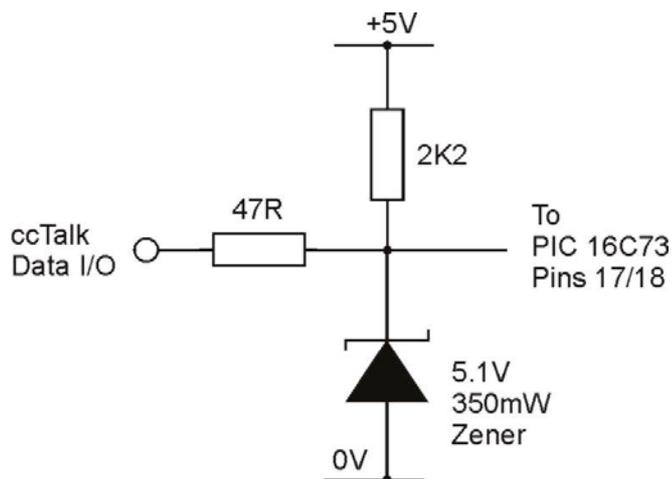
## 25. Appendice C

### 25.1 Circuito di interfaccia ccTalk

Questo è il circuito di interfaccia elettronica di un SCH2.

Vi sono molte opzioni disponibili per il circuito di interfaccia host, ma si consiglia vivamente un'unità a collettore aperto.

*Figura 5: Circuito di interfaccia ccTalk*



## 26. Appendice D

### 26.1 Descrizione della mappa di memoria non volatile

Tabella 13: Descrizione della mappa di memoria non volatile

N. blocco	Lunghezza byte	Descrizione	Permessi lettura/scrittura (R/W)
0	8	Dati utente	R / W
1	6	Nome moneta	R / W
1	2	Dati utente	R / W
2	3	Conteggio rilascio dall'erogatore	R / W
2	1	Checksum A	R / W
2	1	Ultima erogazione: monete rilasciate	R / W
2	1	Checksum B	R / W
2	1	Ultima erogazione: monete non ancora rilasciate	R / W
2	1	Checksum C	R / W
3	3	Conteggio rilascio totale monete dall'erogatore	Solo lettura
3	1	Checksum D	Solo lettura
3	1	Registratore scatola nera A	Solo lettura
3	1	Registratore scatola nera B	Solo lettura
3	1	Registratore scatola nera C	Solo lettura
3	1	Registratore scatola nera D	Solo lettura

#### Dati utente

Nella macchina host vi sono a disposizione 10 byte di dati utente per una qualsiasi memorizzazione. I dati possono essere in formato ASCII o binario – non vi è alcuna restrizione sul formato. Gli 8 bit del blocco 0 e i primi 2 byte del blocco 1 sono completamente disponibili.

#### Nome moneta

Nome della moneta all'interno dell'erogatore. Gli utenti potranno assegnare propri nomi. La stringa ASCII memorizzata in questa posizione viene restituita dal comando "Richiesta moneta erogatore".

#### Conteggio rilascio dall'erogatore

Numero totale di monete erogate a partire dall'ultimo reset del contatore.

Si noti che per leggere il valore di questo contatore si dovrebbe usare il comando ccTalk "Richiesta conteggio rilascio dall'erogatore". Il valore contenuto nella EEPROM non viene aggiornato se non dopo una interruzione di corrente o un reset.

**Ultima erogazione: monete rilasciate**

Numero delle monete erogate con l'ultimo comando.

**Ultima erogazione: monete non ancora rilasciate**

Numero delle monete che non sono state ancora erogate dopo l'ultimo comando.

**Conteggio rilascio totale monete dall'erogatore**

Numero totale delle monete erogate dal momento della produzione dell'erogatore.

Si noti che questo conteggio totale viene **aggiornato unicamente** quando viene aggiornata la memoria non volatile – per esempio quando viene staccata la corrente o viene eseguito un reset. Non conserva un valore periodico simile a quello del conteggio dei rilasci da parte dell'erogatore.

**Registratori checksum e scatola nera**

Ogni contatore viene protetto da un checksum nullo, come quello generato dall'aggiunta di 8 bit a tutti i contatori e dall'impostazione del byte checksum a zero.

Se il software rileva un'anomalia nel checksum, allora verrà impostato un flag di errore al checksum (fare riferimento al comando "Verifica erogatore") e il byte del registratore della scatola nera viene incrementato. I byte del registratore della scatola nera vengono usati per monitorare eventuali problemi connessi alle componenti elettroniche in caso di interruzione di corrente.

**Esempio – Azzeramento del conteggio di rilascio dell'erogatore**

Questa operazione può essere eseguita leggendo il blocco 2 e riscrivendone il contenuto con il byte appropriato riportato a zero.

Vengono qui riportati i pacchetti di messaggio. I valori dei dati sono espressi in forma decimale. L'erogatore ha indirizzo 3.

**Ciclo lettura**

Dati trasmessi: [ 3 ][ 1 ][ 1 ][ 215 ][ 2 ][ checksum ] – Legge blocco 2

Dati ricevuti: [ 1 ][ 8 ][ 3 ][ 0 ][ conta 1 ][ conta 2 ][ conta 3 ][ conta checksum ]  
[ dati 1 ][ dati 2 ][ dati 3 ][ dati 4 ][ checksum ]

**Ciclo scrittura**

Dati trasmessi: [ 3 ][ 9 ][ 1 ][ 214 ][ 2 ][ 0 ][ 0 ][ 0 ][ 0 ] – Scrive blocco 2  
[ dati 1 ][ dati 2 ][ dati 3 ][ dati 4 ][ checksum ]

Dati ricevuti: [ 1 ][ 0 ][ 3 ][ 0 ][ 252 ] - ACK

## 27. Appendice E

### 27.1 Mk2 e Mk1

Se si è già in possesso di un'interfaccia macchina per la versione Mk1 di Compact Hopper seriale, si consiglia di prendere in esame le considerazioni seguenti.

**Nella versione Mk2 sono state apportate le seguenti modifiche ...**

- Il connettore seriale è dotato di altri 2 pin finali – i pin da 1 a 8 sono uguali.
- Il comando “Rilascia monete dall'erogatore” è radicalmente diverso – richiede un codice criptato a 8 byte e restituisce un nuovo contatore di evento invece di un solo “ACK”.
- Il comando “Verifica erogatore” ora restituisce un flag di 2 byte invece che di 1 byte.
- Il comando “Modifica serie variabili” richiede un ulteriore byte.
- Sono state modificate le stringhe di identificazione prodotto.

Qui di seguito vengono riportate le differenze fra Mk2 e Mk1.

Tabella 14: Differenze fra Mk2 e Mk1

Comandi ccTalk	Supporto Mk2	Supporto Mk1	Comandi ccTalk	Supporto Mk2	Supporto Mk1
Intestazione 254: Verifica singola	X	X	Intestazione 214: Scrive blocco dati	X	
Intestazione 253: Verifica indirizzo	X	X	Intestazione 192: Richiesta codice build	X	X
Intestazione 252: Conflitto indirizzo	X	X	Intestazione 172: Arresto di emergenza	X	X
Intestazione 251: Modifica indirizzo	X	X	Intestazione 171: Richiesta moneta erogatore	X	X
Intestazione 250: Indirizzo casuale	X	X	Intestazione 169: Richiesta modalità indirizzo	X	X
Intestazione 247: Richiesta serie variabili	X	X	Intestazione 168: Richiesta conteggio rilascio da erogatore	X	X
Intestazione 246: Richiesta ID produttore	X	X	Intestazione 167: Rilascia monete dall'erogatore	Nota $\alpha$	X
Intestazione 245: Richiesta ID categoria macchinario	X	X	Intestazione 166 : Richiesta stato erogatore	X	X
Intestazione 244: Richiesta codice prodotto	X	X	Intestazione 165: Modifica serie variabili	Nota $\alpha$	X
Intestazione 242: Richiesta numero seriale	X	X	Intestazione 164: Abilita erogatore	X	X
Intestazione 241: Richiesta revisione software	X	X	Intestazione 163: Verifica erogatore	Nota $\alpha$	X
Intestazione 236: Legge stati opto	X	X	Intestazione 161: Avvia RNG – Nuova intestazione	X	
Intestazione 219: Inserire nuovo codice PIN	X		Intestazione 160: Richiesta chiave cifrata – Nuova intestaz.	X	
Intestazione 218: Inserire codice PIN	X		Intestazione 004: Richiesta revisione comunicazioni	X	X
Intestazione 217: Richiesta stato erogaz. alto/basso	X	X	Intestazione 003: Elimina variabili stato comunicazione	X	X
Intestazione 216: Richiesta disponibilità memorizz. dati	X	X	Intestazione 002: Richiesta variabili stato comunicaz.	X	X
Intestazione 215 : Read data block	X		Intestazione 001: Reset dispositivo	X	X

Nota  $\alpha$  : In questi comandi sono stati modificati i pacchetti di dati.

## 28. Appendice F

### Esempi del comando “Richiesta stato erogatore”

La risposta restituita dal comando “Richiesta stato erogatore” è la seguente.

[contatore evento] [monete da rilasciare rimanenti] [ultima erogazione: monete rilasciate]  
[ultima erogazione: monete non ancora rilasciate]

#### **Rilascio di 1 moneta in modalità di rilascio moneta singola**

```
Hopper status = 01 01 00 00  
Hopper status = 01 01 00 00  
Hopper status = 01 01 00 00  
Hopper status = 01 00 01 00 Paid 1 coin
```

#### **Rilascio di 5 monete in modalità di rilascio monete multiple**

```
Hopper status = 02 05 00 00  
Hopper status = 02 05 00 00  
Hopper status = 02 04 01 00  
Hopper status = 02 04 01 00  
Hopper status = 02 03 02 00  
Hopper status = 02 03 02 00  
Hopper status = 02 02 03 00  
Hopper status = 02 02 03 00  
Hopper status = 02 01 04 00  
Hopper status = 02 01 04 00  
Hopper status = 02 00 05 00 Paid 5 coins
```

#### **Rilascio di 5 monete in modalità di rilascio monete multiple. Solo due monete nell'erogatore**

```
Hopper status = 03 05 00 00  
Hopper status = 03 04 01 00  
Hopper status = 03 03 02 00  
Hopper status = 03 01 01 03 Partial result - ignore  
Hopper status = 03 00 02 03 Paid 2 coins, 3 remaining
```

Lo scopo di questo manuale è solo quello di assistere il lettore nell'uso di questo prodotto e, pertanto, la Money Controls non può essere ritenuta responsabile per qualsiasi perdita o danno emergente dall'uso delle informazioni o loro particolari, o per qualsiasi uso non corretto del prodotto. La Money Controls si riserva il diritto di modificare le specifiche del prodotto in qualsiasi sua parte senza darne alcun avviso.