CCTALK Technical Information.

CONTENTS

1.0 INTRODUCTION

The information contained in this document is specific to the CCtalk hopper range.

For additional information on CCtalk specification refer to: CCtalk Serial Communication Protocol, Generic Specification Iss 4.2 (www.cctalk.org)

2.0 ELECTRICAL

Due to the motor power requirement an additional 24V supply is required direct to the motor brake board on the larger hoppers. The red and black wires from the 4 way Amp EI series connector are supplied with free ends for this purpose.
However, the SH-400 and SH-300 series require only one supply to the CCTALK assembly, as the motor is powered directly from the CCTALK interface.

2.1 Serial Connector

The connection to the host machine is by Molex 10 way header
The Molex references for the suitable mating housings are:

Hirose: HIF3BA-10D-2.54C
Molex: 90142-0010

Pin	Function
1	Data (CCtalk)
2	N.C.
3	N.C.
4	N.C.
5	N.C.
6	N.C.
7	24V +ve
8	0V
9	N.C.
10	N.C.

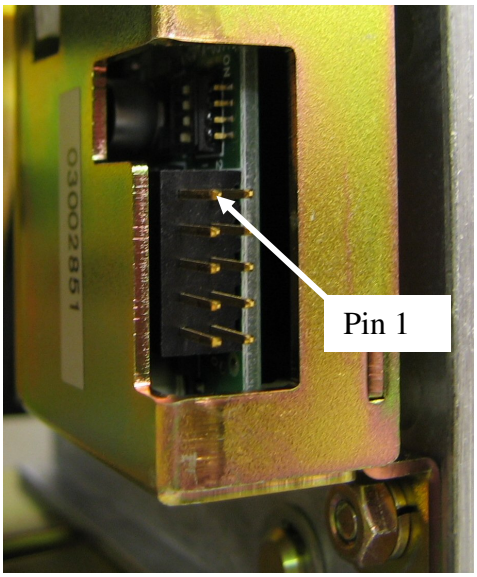
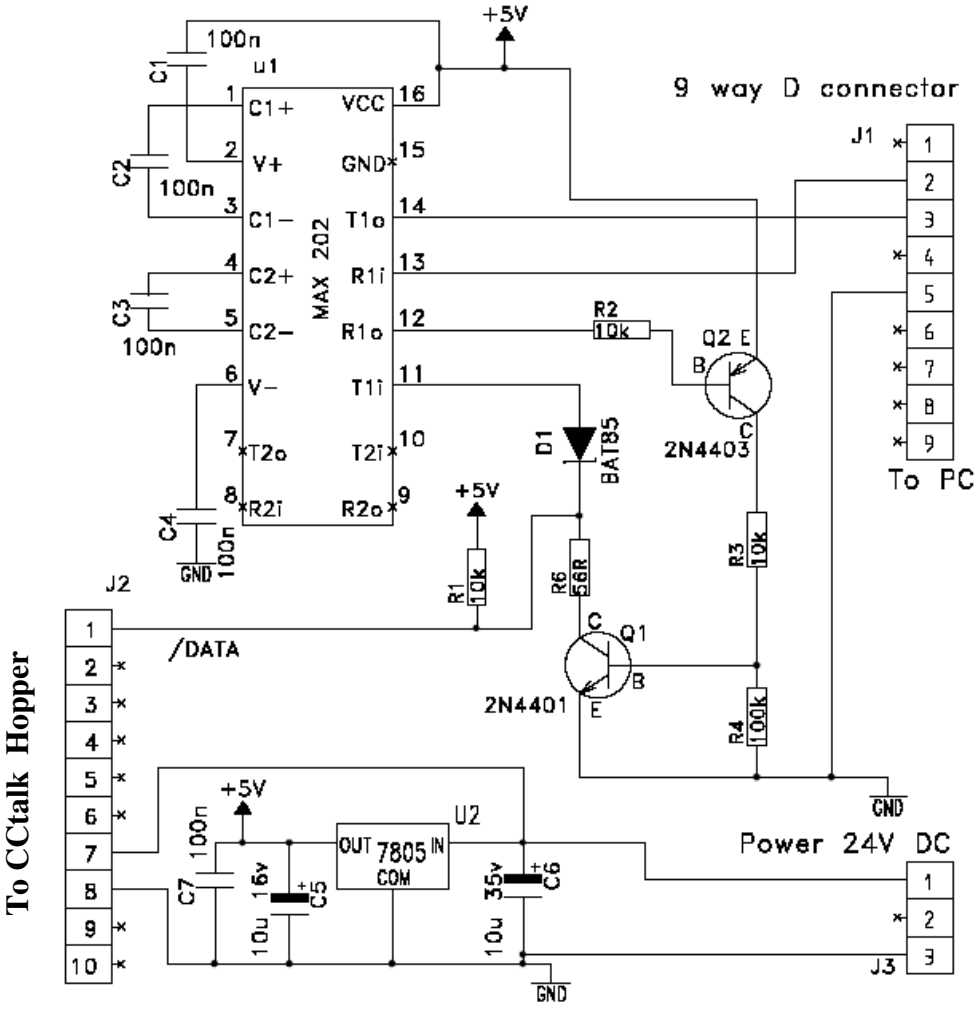


Fig: 2.1 Connector detail

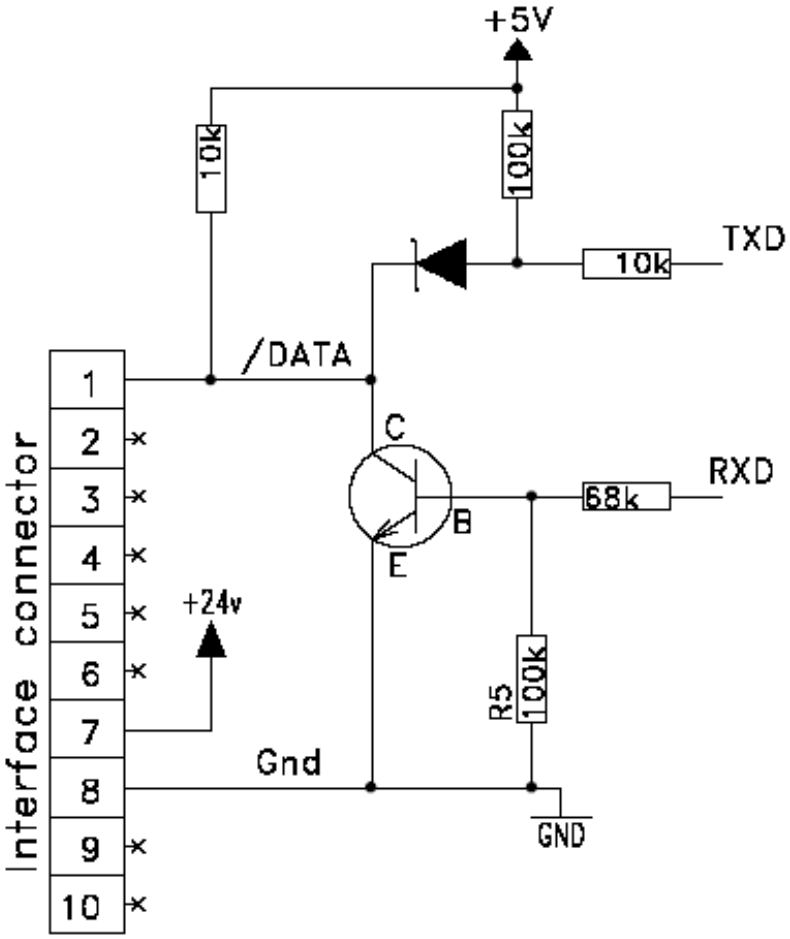
2.2 PC interface circuit

Recommended circuit for connecting the hopper to a PC Com port via a RS 232 interface.



2.3 Internal Interface Circuit

For information only. This drawing shows the interface part of the hopper control board.



Internal interface circuit

2.4 Device Address and DIP Switch Settings

All Asahi Seiko hoppers leave the factory with address 3 stored in the EEPROM. Up to eight different addresses can be configured by using the dipswitch mounted on the control board.

Device Address	SW3	SW2	SW1
3	ON	ON	ON
4	ON	ON	OFF
5	ON	OFF	ON
6	ON	OFF	OFF
7	OFF	ON	ON
8	OFF	ON	OFF
9	OFF	OFF	ON
10	OFF	OFF	OFF

Alternatively up to 255 addresses can be selected using serial command 251.

N.B. Once the address has been set using header 251 the dip-switches become irreversibly ineffective.

3.0 COMMUNICATION PROTOCOL

CCtalk is a serial communication protocol developed by Money Controls to interconnect various cash handling devices in machines used in the gaming and amusement industry.

3.1 Message Structure.

The message is assembled in the following manner.

[Destination Address] Abbreviated ADDR (This is the 'slave' address)
[Number of Data Bytes] e.g. '00'
[Source Address] e.g. '01' (Host address '1')
[Command Header] e.g. 'FE' (Simple Poll)
[Data1]**……..**[Data N] e.g. '00' (There are no data bytes)
[Checksum] Abbreviated 'Chk'

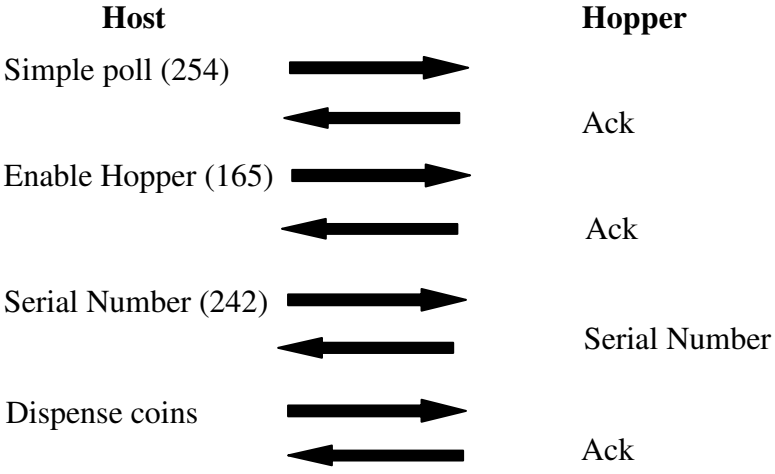
3.2 Checksum.

The checksum is derived by adding all the Bytes in the message, dividing by 256 and subtracting the remainder from 256.

The received message sum including checksum should equal zero provided no errors have occurred in transmission.

3.3 Recommended Command Sequence.

Example of command sequence to dispense coins.



3.4 Command Examples

The following examples show the actual hexadecimal commands and hopper responses.

3.4.1 Command Header 254, Simple Poll

Host sends: [ADDR][00][01][FE][Chk]

Hopper answer: [01][00][ADDR][00][Chk]

Actual messages assuming factory default address of 3:

Host sends: [03][00][01][FE][FE]

Hopper answer: [01][00][03][00][FC]

3.4.2 Command header 164, Enable Hopper.

Host sends: [ADDR][01][01][A4][d1][Chk]

Hopper answer: [01][00][ADDR][00][Chk]

Actual messages transmitted:

Host sends: [03][01][01][A4][A5][B2]

Hopper answer: [01][00][03][00][FC]

3.4.3 Command header 242, Request Serial Number.

Host sends [ADDR][00][01][F2][Chk]

Hopper answer [01][03][ADDR][00][LSB][Byte 2][MSB][Chk]

Actual messages transmitted:

Host sends [03][00][01][F2][0A]

Hopper answer [01][03][03][00][DA][5E][00][C1]

DA5E00 (Serial number in HEX LSB-MSB)
005EDA (Serial number in HEX MSB-LSB)
Actual serial number in Decimal = 24282

3.4.4 Command Header 167, Dispense coins (5 coin example)

Host sends [ADDR][04][01][A7][ser1][ser2][ser3][No. of coins][Chk]

Hopper answer [01][00][ADDR][00][Chk]

Actual messages transmitted:

Host sends [03][04][01][A7][DA][5E][00][05][14]

Hopper answer [01][00][03][00][FC]

3.5 EEPROM Logic Data List

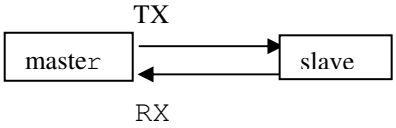
Number	Type	Size(byte)	Data ID	Descriptions
1	ASCII	4	Initial code	If already initialized, "asahi" is set
2	binary	3	Serial number	Unique number assigned for each device
3	ASCII	12	product code	Indicate product name (eg: SH-400)
4	binary	4	ROM checksum	Set checksum value of ROM
5	ASCII	10	build code	Set product code eg: “KSP8006”
6	binary	4	PIN number	Assigns 32bits PIN number. Currently no command to protect with PIN number but supporting PIN number entry function which is essential.
7	binary	1	Reserved	-
8	binary	1	Reserved	-
9	binary	1	Reserved	-
10	binary	1	h_addr(host address)	Setting in case of not using default value (=1)
11	ASCII	6	Coin ID	Sets a coin ID according to cctalk spec (eg: ”EU100A”(1Euro))
12	binary	2	Max. coin capacity	Sets each max. number of coins accommodated in accordance with denomination and the size of hopper
13	binary	2	Remaining amount	Set the amount remaining in the hopper (*1)
14	binary	3	Reserved	-
15	binary	1	s_addr(slave address)	To be set when instructed by Host
16	binary	2	last payout	Prior byte: Number of coins dispensed to the end. Subordinate byte: Number of remaining coins when dispensed to the end.

EEPROM Life time

BR24C01AF-W - Up to 100,000 re-writes

3.6 Header list for Asahi Seiko hoppers (Issue 3.1)

The table below lists all commands
Data length in Bytes without header



Class	Header	RX max. length	RX data	TX max. length	TX data	Function
Core	**254**	0	nothing	0	nothing	**Simple poll**
	246	20	ASCII string from hex values	0	nothing	**Request manufacturers ID** Returns ASCII letters"Asahi Seiko Co., Ltd"
	245	6	ASCII string from hex values	0	nothing	**Request equipment category ID** Returns ASCII letters"Payout"
	244	12	ASCII string from hex values	0	nothing	**Request product code**
	192	10	ASCII string from hex values	0	nothing	**Request build code**
Core Plus	**253**	1	address	0	nothing	**Address Poll** (Returns address with a time interval of 4 x the address)
	252	1	address	0	nothing	**Address Clash** (Returns address with a random time interval)
	251	0	nothing	1	address	**Address Change** Assign 0 (zero) to validate DIPSW setting Assign 1 –~255 to slave address
	250	0	nothing	0	nothing	**Address Random** (Creates random number as address) Never 0 or 1
	242	3	LSB-MSB	0	nothing	**Request Serial Number**
	241	16	ASCII string from hex values	0	nothing	**Request Software Revision** (e.g.: [v01.00' 03.07.14])
	197	4	checksum 1,2,3,4	0	nothing	**Calculate ROM Checksum**
	169	1	hex byte	0	nothing	**Request Address Mode**
	4	3	cctalk level, major revision, minor revision	0	nothing	**Request Communication Revision** cctalk level:1, major revision:3, minor revision:1
	3	0	nothing	0	nothing	**Clear Communication Status Variables** (Clears the three single byte cumulative event counters)
	2	3	rx timeouts,rx bytes ignored,rx bad checksums	0	nothing	**Request Communication Status Variables** Returns the three single byte cumulative event counters (receive time-outs, receive bytes ignored and bad checksums) NB Value 255 wraps around to 0
	1	0	nothing	0	nothing	**Reset Device**
Payout for serial hoppers	**247**	2	Data1：Device status, Data2：Device error	0	nothing	**Request Variable Set**
	219	0	nothing	4	PIN 1,2,3,4	**Enter New PIN Number** (Update PIN number)
	218	0	nothing	4	PIN 1,2,3,4	**Enter PIN Number** (In the event of an incorrect PIN number being entered, ACK response after approx 1s)
	217	1	level status	1	hopper no	**Request Payout high/low status**
	208	0	nothing	3	hopper no. and no. of coins	**Modify Payout Absolute Count** (Initialize to a known value, the absolute number of coins in the hopper)
	207	2	LSB-MSB	1	hopper no	**Request Payout Absolute Count** (Returns absolute number of coins in hopper)
	187	0	nothing	3	hopper no. and no. of coins	**Modify payout capacity** (Set max number of coins payout device can hold))

Class	Header	RX max. length	RX data	TX max. length	TX data	Function
Payout for serial hoppers	**186**	2	LSB-MSB	1	hopper no	**Request payout capacity** (Returns max number of coins payout device can hold)
	175	0	nothing	3	hopper no. and no. of coins	**Modify payout float** (Sets the working 'float' level for a payout device)
	174	2	LSB-MSB	1	hopper no	**Request payout float** (Returns the working 'float' level for a payout device)
	172	1	Hex byte	0	nothing	**Emergency Stop**
	171	6	ASCII string from hex values	0	nothing	**Request Hopper Coin** (Returns coin ID)
	168	3	LSB-MSB	0	nothing	**Request Hopper Dispense Count** (Returns number of coins dispensed by the hopper)
	167	0	nothing	4	serial number (3 bytes) and number of coins (1 byte)	**Dispense Hopper Coins** (serial number + assigned dispense number)
	166	4	event counter, payout coins remaining, last payout: coins paid, last payout: coins unpaid	0	nothing	**Request Hopper Status**
	164	0	nothing	1	enable code (A5h)	**Enable Hopper**
	163	1	status code (status register 1)	0	nothing	**Test Hopper**

3.7 Header detail

Header 247: Request variable set.

Received data: [Device status][Device error]

This command specification originated from Asahi Seiko Ltd.
For more information about the device status, please refer to the “Device Status” table below.

code	Description	details
"0"(0x30)	device normal	
"1"(0x31)	device error 1	To be assigned
"2"(0x32)	device error 2	To be assigned
"3"(0x33)	device error 3	To be assigned
"4"(0x34)	device error 4	Device normal: empty detection
"5"(0x35)	device error 5	Device error: payout retry over reason: coin jam, shortage of coins, coins catching, etc
"6"(0x36)	device error 6	Device error: coin sensor error (except during payout)
"7"(0x37)	device error 7	Device error: over payment
"8"(0x38)	device error 8	To be assigned
"9"(0x39)	device error 9	Device normal: full detection
"A"(0x41)	device error A	Device error: coin sensor error (during payout)

Device status

Device status	Descriptions
Initialization(0x20)	On power up and receiving device reset.
Internal Reset (0x21)	Sensor error, dispense retry over or overpayment (the sensor should be checked)
Stand by for Enable(0x25)	Dispense disabled. Other commands are available. If enabled by receiving [164 : Enable hopper] the status will change.
EEPROM yet to be initialized(0x2A)	When the initial code in EEPROM cannot be properly read and initialization is incomplete or power shut down during EEPROM write.
STOP(0x30)	Hopper is stopped
BUSY(0x40)	Hopper dispensing
Shipment mode(0x80)	Reserved by the Factory for set-up and test.

Header 163: Test Hopper

Use Header 163 (Test Hopper), for hopper diagnostics.
N.B. One byte of data is returned (to issue 3.1).

Hopper status register 1:(Returned data details)
Bit 0 - Absolute maximum current exceeded
Bit 1 - Payout timeout occurred
Bit 2 - Motor reversed during last payout to clear a jam
Bit 3 – Opto. fraud attempt, path blocked during idle
Bit 4 – Opto. fraud attempt, short-circuit during idle
Bit 5 – Opto. blocked permanently during payout
Bit 6 - Power-up detected
Bit 7 - Payout disabled

Asahi Seiko CCTALK does not respond to Bits 0,2,4,6.

Below are the error condition definitions:

Bit1: Payout timeout occurred

This bit is ON when designated number of coins were not dispensed.

This error condition occurs when:
1) Insufficient number of coins remaining in hopper bowl.
2) Sufficient number of coins in hopper bowl, but coins failed to dispense.
3) Coin jam.

If the coin sensor does not detect a coin within 2 seconds, the coin disc stops for 0.5 sec and then re-starts for 2 sec. If again no coin is detected, it stops for another 0.5 sec then rotates for a third and final attempt to dispense a coin before 'time out' occurs.

Bit3: Opto fraud attempt, path blocked during idle

This bit is ON when a coin sensor error is detected during idle.

This error condition occurs when:
1) The coin sensor detects a coin (for more than 30ms) during idle.
2) The interface self-diagnostics detects a sensor fault condition during an idle state.

Error condition 2 above, may be due to the sensor connector becoming dislodged.
Very few sensor faults occur, in most cases, the problem is due to the sensor wiring or connector.

Bit 5: Opto blocked permanently during payout

This bit is ON when coin a coin sensor error is detected during dispense (busy).

This error condition occurs when a coin is detected for more than 300ms.
Check for a coin blocking the payout sensor.

Bit 7: Payout disabled

This bit is ON when dispense is inhibited.
Enable payout via 'Enable Hopper' command (Header 164)
When a coin sensor error occurs, the Enable Hopper command (Header 164) is disabled.

[example]
Receive:[01][01][03][00][88][73]
Send:[03][00][01][A3][59]

In this example, 88h is equivalent to 10001000(binary) and Bit 7 is ON, thus the Enable Hopper command is disabled.
Also Bit 3 is ON, thus either error condition 1) or 2) for bit 3 is present during the 'idle' state.

4.0 DOCUMENT CONTROL

Date	Issue No.	Reason for Revision	Page No.
April 2007	Issue 1	Original issue	
Sept 2008	Issue 2	Additional CCTALK info added	11 to 16
Mar 2009	Issue 3	Header 251 warning	7
April 2010	Issue 4	Additional information added.	4
June 11	Issue 5	3.1 Information added. Payout re try details corrected.	8 &15