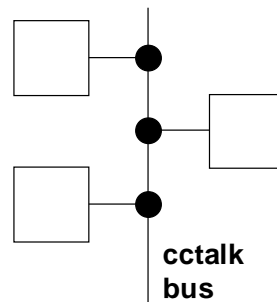


C435S Product Manual



Version 2.1

This document is the copyright of Money Controls Ltd and may not be reproduced in part or in total by any means, electronic or otherwise, without the written permission of Money Controls Ltd. Money Controls Ltd does not accept liability for any errors or omissions contained within this document. Money Controls Ltd shall not incur any penalties arising out of the adherence to, interpretation of, or reliance on, this standard. Money Controls Ltd will provide full support for this product when used as described within this document. Use in applications not covered or outside the scope of this document may not be supported. Money Controls Ltd. reserves the right to amend, improve or change the product referred to within this document or the document itself at any time.

Revision History

<u>Issue</u>	<u>Date</u>	<u>Comments</u>
1.0	11/11/98	First draft
1.1	27/11/98	Position E of rotary switch is for routing control
1.2	21/12/98	Additional information on coin counters Diagnostic polarity corrected
1.3	22/12/98	Minor text revisions
1.4	08/02/99	Clarification of security setting - lower & upper banks
1.5	23/04/99	Clarification of connector orientation
1.6	18/06/99	Change contact details to Money Controls Refer to Piezo as option
1.7	16/09/99	Typo in 17.2 Overrides... data should be shown as 8 bits not 9 bits. The 'Modify sorter override status' command only works in EEPROM routing mode.
1.8	10/04/00	Note added to section 'Error Codes Explained'
1.9	31/08/99	Clarification... Rotary switch functions only operate in parallel mode
2.0	15/04/03	Issued TSP number
2.1	30/06/04	Changed footer

Contents

Cover Sheet	1
Revision History	2
Contents	3
1 Product Overview	5
2 Product Specification	5
Parallel Operation	
3 Parallel Connectors	6
3.1 Main	6
3.2 Overrides	6
3.3 Routing Plug	7
3.4 Rundown	7
4 Bank Operation	8
5 BACTA Binary Credit Mode	8
6 Parallel Credit Mode	9
7 Rotary Switch	9
7.1 Position 0 : Token Disable	9
7.2 Position 1 to C : Token Select	9
7.3 Position D : Mode Switch Control	10
7.3.1 Switch 1	10
7.3.2 Switch 2	10
7.4 Position E : Routing Control	10
7.4.1 Switch 1	10
7.5 Position F : Teach-and-Run	11
8 Diagnostics	11
8.1 Bottom Link	11
8.2 Top Link	12
8.3 Top & Bottom Links	13
Serial Operation	
9 Serial Protocol	14
10 Serial Interface Specification	14
11 Serial Interface Connector	14
12 Serial Interface Circuit	15
13 Serial or Parallel Mode	15
14 Where Do I Start ?	16
15 Protocol Determination	17
16 Credit Polling	17
17 Initial Settings	18
17.1 Inhibits	18
17.2 Overrides	18
17.3 Token Selection	18
18 Command List	18
19 Command List Explained	20
20 PIN Protection	33

21 Command Response Timing	33
22 Data Lengths	36
22.1 Buffer Sizes	38
23 Error Code Table	38
24 Error Codes Explained	38
25 Fault Code Table	41
26 Fault Codes Explained	41
27 Fault Code Priority	42
28 Status Code Table	43
29 Coin Counters	43

Specification Sheets

30 Miscellaneous Timings	44
30.1 Parallel Inhibit Timing	44
30.2 Parallel Override Timing	44
30.3 Serial Inhibit / Override Timing	44
31 Coin Sizes	44
32 Mechanical Data	45
33 Electrical Data	45
33.1 Power Supply	45
33.2 Current Requirements	45
33.3 Output Drive Capability	45
34 Flammability Data	45
35 Environmental Data	45
35.1 Temperature	45
35.2 Humidity	46
36 EMC Performance	46
36.1 Emissions	46
36.2 Immunity	46
37 Shock / Vibration	46
38 Cleaning	46
39 Lifetime	46

1 Product Overview

C435S is a revolutionary new coin acceptor designed for applications in the AWP, vending and ticketing sectors of the coin industry. The most important design feature is that it can operate in both **parallel and serial interface** modes. In other words, it is a *dual-mode* product. It can operate in parallel mode to the BACTA binary interface standard for GB coinage and is fully compatible with the existing C435A product. In serial mode, it conforms to the cctalk serial protocol, recently agreed as a future standard by the BACTA manufacturers committee.

C435S can be used as a drop-in replacement for existing C435A validators, or may be designed into new, intelligent, gaming machines. The serial interface architecture is easily scalable - it can be used as a simple replacement to the binary credit codes or it can be used in more sophisticated applications to monitor accept rates and faults, and to take the necessary actions which result.

The benefits of a serial interface to the coin industry are reduced down-time and increased revenue through real-time performance monitoring in an area where traditionally this data has not been available.

The impressive fraud discrimination ability of C435S is carried over from C435A GB+ which uses state-of-the-art technology to parameterise bi-metallic coins, such as GB £2 and Euro coins. It also has a piezo sensor option to eliminate the common problem of lead and solder slugs. Any gaming machine using C435S will benefit from the same advanced discrimination technology found on C435A.

2 Product Specification

If you are unfamiliar with C435A...

The C435S is an electronic coin validator capable of discriminating between 16 pre-programmed coin types. It is available in an industry-standard 5 inch body and is capable of sorting coins down 1 of 4 paths (expandable to 8 paths) as well as the reject channel.

Features include...

- Token support via an extra 12 pre-programmed token windows on top of the 16 coin windows and an optional 'dual coin entry' token rundown
- Built-in diagnostic functions
- 'Teach-and-run' for programming special operator tokens
- Alarm functions for detecting coin jams and fraud attempts
- Sorter override function for diverting coins away from full tubes or hoppers
- Front or top entry build options
- 12V to 24V power supply

Parallel Operation

If you are only interested in hooking up to the serial connector of this product then proceed directly to section 9.

3 Parallel Connectors

3.1 Main

Pin	Signal	Comments
1	Accept 6	BACTA binary credit mode identification
2	Accept 5	-
3	Accept Common	COMA voltage (positive or negative)
4	Accept 1	-
5	Polarising Key	-
6	Accept 2	-
7	Accept 3	-
8	Credit Mode Selection	Connect to 0V for binary credits
9	Accept 4	Binary credit 'strobe' line
10	Inhibit 4	-
11	+Vs	Power supply
12	0V	Power supply return
13	Inhibit 3	-
14	Inhibit 2	-
15	Inhibit 1	-
16	Inhibit 5	-
17	Inhibit 6	-

3.2 Overrides

Pull the override line to 0V to prevent a coin going down a particular path.

Pin	Signal
1	0V
2	Polarising Key
3	Override 7 (path C)
4	Override 6 (path B)
5	Override 5 (path A)
6	Override 4 (path A + manifold drive)
7	Override 3 (path B + manifold drive)
8	Override 2 (path C + manifold drive)
9	Override 1 (path D + manifold drive)
10	+Vs

3.3 Routing Plug

This is an industry standard routing plug. For example, to send coin 2 down sorter path B, connect pins 10 and 15. It is possible to send more than one coin type down the same sorter path if diodes are used (anodes to coin signals). It is also possible to connect paths together to give an override sequence - in which case diodes are not needed. Coins are always routed in the order **D then C then B** then default.

Pin	Signal	Pin	Signal
1	Coin 7	2	-
3	-	4	-
5	0V	6	-
7	Path D	8	Coin 1
9	-	10	Coin 2
11	0V	12	Coin 3
13	Path C	14	0V
15	Path B	16	Coin 4
17	Coin 6	18	Coin 5

The routing plug connector is a dual row type with pins as viewed from the front :

2	4	6	8	10	12	14	16	18
1	3	5	7	9	11	13	15	17
				Key				

3.4 Rundown

Pin	Signal
1	Opto drive (+5V through 220R)
2	Token opto receive
3	Coin opto receive
4	Not used
5	Polarising Key
6	0V

4 Bank Operation

C435S is organised internally the same as C435A, that is to say there are 16 coin windows, grouped into 2 banks of 8 coins. This grouping simplifies the inhibit interface on the parallel connector as each inhibit controls 2 coin windows, e.g. inhibit 1 controls coins 1 & 9, inhibit 2 controls coins 2 & 10... inhibit 8 controls coins 8 and 16. The bank select switches can be used to control whole banks at a time. Bank select switch 1 controls coins 1 to 8, and bank select switch 2 controls coins 9 to 16.

Position	Function
Up (on)	Coin bank disabled
Down	Coin bank enabled

The usual configuration is both bank switches in the down position.

Note that conformance to the BACTA specification for binary and parallel operation places further restrictions on inhibit control as there are only 6 inhibit lines connected to the outside world.

5 BACTA Binary Credit Mode

A4 = Strobe for binary credit code

A6 = Always on

X = Accept line on (pin outputs at COMA voltage)

- = Accept line off (high impedance)

Inhibit line 2 is the 'remainder' function, mopping up miscellaneous coins.

Inhibit line 4 is not used.

No.	Coin	Inhibit	A6	A5	A4	A3	A2	A1
1	£1	I1	X	X	X	-	X	-
2	50p	I2	X	-	X	X	-	X
3	20p	I3	X	-	X	-	X	X
4	10p	I2	X	X	X	X	-	-
5	Token	I5	X	-	X	X	X	-
6	£2	I6	X	X	X	X	X	X
7	5p	I2	X	-	X	-	-	-
8	50p old	I2	X	X	X	-	-	X
9	Extra 1	I1	X	X	X	X	X	-
10	Extra 2	I2	X	X	X	X	-	X
11	Extra 3	I3	X	X	X	-	X	X
12	Extra 4	I2	X	-	X	X	X	X
13	Extra 5	I5	X	X	X	-	-	-
14	Extra 6	I6	X	-	X	X	-	-
15	Extra 7	I2	X	-	X	-	X	-
16	Extra 8	I2	X	-	X	-	-	X
-	Alarm	-	X	X	-	X	X	X

6 Parallel Credit Mode

X = Accept line on (pin outputs at COMA voltage)

- = Accept line off (high impedance)

£2 is always inhibited in this mode.

No.	Coin	Inhibit	A6	A5	A4	A3	A2	A1
1	£1	I1	-	-	-	-	-	X
2	50p	I2	-	-	-	-	X	-
3	20p	I3	-	-	-	X	-	-
4	10p	I4	-	-	X	-	-	-
5	Token	I5	-	X	-	-	-	-
6	£2	-	-	-	-	-	-	-
7	5p	I6	X	-	-	-	-	-
8	50p old	I2	-	-	-	-	X	-
9	Extra 1	I1	-	-	-	-	-	X
10	Extra 2	I2	-	-	-	-	X	-
11	Extra 3	I3	-	-	-	X	-	-
12	Extra 4	I4	-	-	X	-	-	-
13	Extra 5	I5	-	X	-	-	-	-
14	Extra 6	-	-	-	-	-	-	-
15	Extra 7	I6	X	-	-	-	-	-
16	Extra 8	I2	-	-	-	-	X	-
-	Alarm	-	-	X	-	X	X	X

7 Rotary Switch

Do not operate the validator in a machine with the rotary switch in positions D, E or F. It will not accept any coins.

Before using the rotary switch to change anything ensure that the serial loom is disconnected and that the validator is operating in PARALLEL mode.

7.1 Position 0 : Token Disable

In this position, token acceptance is disabled.

To use : Turn to this rotary switch position and press the program button.

7.2 Positions 1 to C : Token Select

11 tokens (or usually groups of similar tokens) can be pre-programmed into the validator and selected on the rotary switch. Token 12 ('C' on the rotary switch) is reserved for the teach-and-run token.

1 to B	Token Groups
C	Teach-and-run Token

To use : Turn to this rotary switch position and press the program button.

7.3 Position D : Mode Switch Control

This position can be used to control 2 modes of operation of the validator

- Coin / token entry mode
- Alarm codes

7.3.1 Switch 1

Position	Function
Up (on)	Industry-standard mode
Down	CCL mode using a DCE rundown

7.3.2 Switch 2

Position	Function
Up (on)	Alarm codes disabled
Down	Alarm codes enabled

Note that alarm codes can also be disabled in serial mode, although most operators would prefer to receive the maximum amount of information over the cctalk bus.

To use : Make a note of the existing rotary switch position and the bank select switches. Turn to the new rotary switch position and set the bank select switches to the desired operating mode. Press the program button. Return the rotary switch position and the bank select switches to their last setting. Press the program button.

7.4 Position E : Routing Control

The inhibit setting mode of C435 has been dropped on this product due to the complexities of all the various operating modes. Position E is now used to control the routing mode when a sorter is fitted. Sorter routes can be obtained from the routing plug (see section 3.3) or from factory values programmed into EEPROM.

7.4.1 Switch 1

Position	Function
Up (on)	Use the Routing Plug
Down	Use routing codes in EEPROM

Note that the routing mode can only be changed if the coin acceptor is operating in parallel mode. In serial mode, changing the operating mode using the rotary switch will have no effect (the same applies to other rotary switch positions). Pin 9 on the 10-way connector controls whether the coin acceptor operates in parallel or serial mode.

When operating in serial mode it is recommended that the coin acceptor is put into EEPROM routing mode due to a restriction on the use of serial overrides - see the 'Modify sorter override status' command. This can be done on the rotary switch as shown above but units can be ordered from the factory with the correct setting already configured. Remember to select 'EEPROM routing' at order entry time.

7.5 Position F : Teach-and-Run

This position can be used to teach a new token into token window 12. Any number of tokens can be used during the teach process, but between 8 and 20 is optimal. Try to

use a representative sample of tokens rather than re-inserting the same token over and over again.

To use : Turn to the new rotary switch position and press the program button. Enter the tokens not faster than 3 per second. Return the rotary switch position to position C to accept the newly programmed tokens. Press the program button.

8 Diagnostics

To enter diagnostic mode, the validator needs to be powered-up or reset (press the program button under the rotary switch) **in parallel mode** with suitable links on the diagnostic connector.

The links are made horizontally across the pins.

8.1 Bottom Link

This mode is used for checking the solenoid drives.

Inhibit 1 → Sorter solenoid 1, plastic flap (enable coin to activate solenoid)

Inhibit 2 → Sorter solenoid 2, metal flap

Inhibit 3 → Manifold solenoid

Inhibit 4 → Accept gate solenoid

Inhibit 5 → N/C

Inhibit 6 → N/C

Inhibit 7 → N/C

Inhibit 8 → N/C

Note that if the validator does not have a sorter fitted then inhibits 1 & 2 will not do anything. Likewise, if the active manifold is not fitted for 8-way routing, then inhibit 3 will not do anything.

Advanced Feature

The accept lines are used to report hardware faults on the inductive coils.

- Accept 1 → Status of inductive sensor 1 (if not active then fault)
- Accept 2 → Status of inductive sensor 2
- Accept 3 → Status of inductive sensor 3
- Accept 4 → Status of inductive sensor 4
- Accept 5 → On
- Accept 6 → On

The most easily picked-up fault is when a coil wire has been broken or comes away from a solder pad.

8.2 Top Link

This mode is used for checking the inhibit and accept lines in a loop-back test. The status of the inhibit lines (1 to 6 only) are reflected onto the accept lines.

- Inhibit 1 → Accept 1 (enable coin to activate accept line)
- Inhibit 2 → Accept 2
- Inhibit 3 → Accept 3
- Inhibit 4 → Accept 4
- Inhibit 5 → Accept 5
- Inhibit 6 → Accept 6
- Inhibit 7 → N/C
- Inhibit 8 → N/C

Advanced Feature

By turning the rotary switch in this mode (any number of positions, clockwise or anti-clockwise), the degree of electrical noise on the inductive measurement system can be measured. The response of the validator is to 'clap' the accept gate a number of times, dependent on the level of noise present. The more 'claps', the more noise.

- 0 claps no detectable noise
- 1 to 5 claps small amount of noise, can be tolerated
- > 5 claps unacceptable amount of noise

If there is an unacceptable amount of noise then a number of standard checks can be performed. Is the power supply waveform clean ? Is there any supply ripple ? Is the validator positioned too close to a VDU or fluorescent tube ? Is there some close proximity motor noise ?

8.3 Top & Bottom Links

This mode is used for checking whether the various opto electronic devices used in the product are working correctly.

Accept 1 → DCE coin opto (ON = opto blocked or not fitted)

Accept 2 → DCE token opto

Accept 3 → Sorter optos

Accept 4 → C.O.S. opto switch

Accept 5 → Off

Accept 6 → Off

The test is interactive, so the optos can be blocked and unblocked with a small piece of card to see if they are working correctly. Note that there is no way to distinguish between a block (a fault) or a missing opto (item not fitted).

The default C435S build is to **not** have a C.O.S. switch fitted.

Serial Operation

9 Serial Protocol

C435S implements the **cctalk serial protocol**. This protocol has been developed specifically for the coin industry and is a simple to implement and highly efficient language.

Details of what cctalk is and how it works are not given in this document. Please refer to...

cctalk Serial Communication Protocol - Generic Specification - Issue 3.0

If you do not already have a copy, then contact the Technical Services department at Money Controls who will be able to send or e-mail you an electronic copy as it currently runs to 78 pages.

10 Serial Interface Specification

The serial interface code for this product is...

cctalk b96.p0.v12.a12.d0.c5.m0.x8.i1.r3

The most important points are that it runs at 9600 baud with a nominal +12V pull-up, open-collector interface. Messages conform to release 3.X of the cctalk specification.

11 Serial Interface Connector

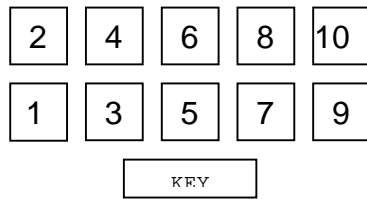
A 10-way, dual-row connector is provided which supports both the Mars HI² (Host Intelligent Interface © Mars Electronics International) interface and the Coin Controls cctalk interface. The pins are on a 0.1 inch grid.

(1) /DATA	← cctalk interface
(2) DATA 0V	internally connected to 0V
(3) /BUSY	not used in cctalk (not connected)
(4) BUSY 0V	internally connected to 0V
(5) /RESET	optional use in cctalk
(6) /PF	not used in cctalk (not connected)
(7) +12V	← cctalk interface
(8) 0V	← cctalk interface
(9) /SERIAL MODE	← cctalk interface, connect to 0V for serial operation
(10) +12V alt.	not used in cctalk (not connected)

Mates with :

Molex 40312 Series MX50 ribbon cable connector system
P/N 15-29-9710 (15µ gold, centre polarisation, strain relief)

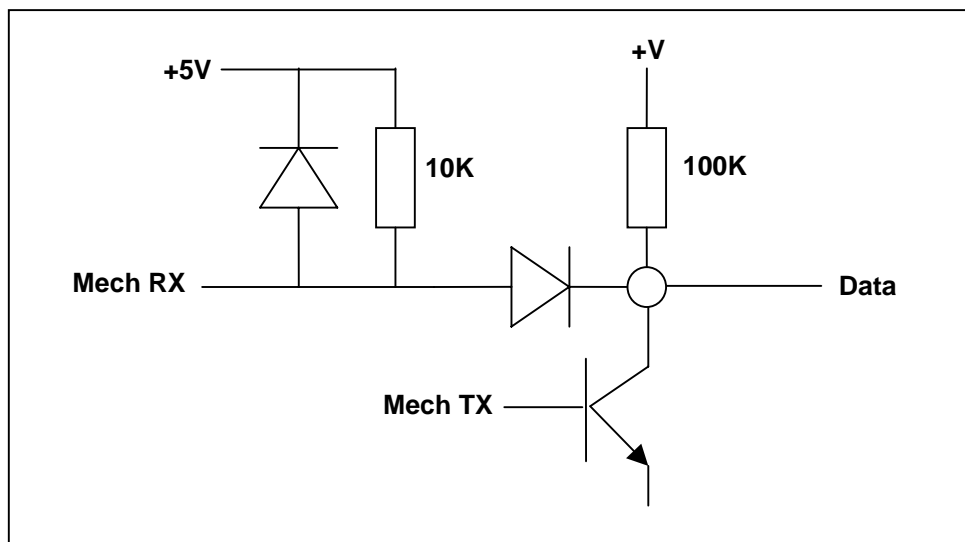
Pinout of Connector (as seen by mating connector) :



It is assumed that the mating connector will have pin 9 connected to 0V to place the product in serial mode as soon as this connector is plugged in.

12 Serial Interface Circuit

The schematic is shown as follows...



13 Serial or Parallel Mode ?

Pin 9 of the serial connector controls whether the validator operates in serial mode or parallel mode. For serial mode, the pin must be tied to 0V (pin 8 of connector).

It will be found that some serial commands still operate in parallel mode. For instance, even though the validator is sending out parallel credit codes, it is possible to read the electronic serial number. However, the 'Read buffered credit or error codes' command will not report any changes in the event counter.

It is recommended that if the serial interface is being used for any reason, the validator is always put into, and left in, serial mode. Correct operation cannot be guaranteed if serial commands are used in parallel mode, even though they appear to work.

14 Where Do I Start ?

When faced with an overwhelming amount of information and a new serial protocol, it is difficult knowing where to begin. Assuming that some kind of cctalk hardware interface has been lashed-up, try the following sequence of messages to get a feel for the protocol.

[xxx] indicates byte values in decimal.

Step 1 : Poll the validator to see if it is there

Send [2] [0] [1] [254] [255] to the validator.

You should receive [1] [0] [2] [0] [253] back. A suitable timeout can be set as '50ms' in case nothing comes back at all !

Note that because cctalk uses a single bi-directional data line, it is possible that all transmitted data is immediately returned on the receive line. This means that your receive buffer will contain the outgoing message as well as the subsequent reply. A suitable index into the receive buffer will provide you with the data of interest.

Step 2 : Check the credit buffer

Send [2] [0] [1] [229] [24]

The 5th byte of the receive message is the event counter. Drop a coin into the validator. Send the command again. The event counter should have incremented by one, regardless of whether the coin accepted or not.

Step 3 : Send some coin inhibit information

Send [2] [2] [1] [231] [255] [255] [22]

All coins should be enabled.

Send [2] [2] [1] [231] [0] [0] [20]

All coins should be inhibited.

Step 4 : Flap the accept gate

Send [2] [1] [1] [240] [1] [11]

The accept gate should open for 0.5s and then close again. The ACK message is not returned until the accept gate shuts.

Step 5 : Calculate the ROM checksum

Send [2] [0] [1] [197] [56]

A 4 byte ROM checksum should be returned within 1.5s.

That completes the introductory course. It should all be plain sailing now !

15 Protocol Determination

A host machine at power-up is faced with the problem of identifying which serial protocol has been implemented - HI² or cctalk.

Support for cctalk is best tested for by sending a 'Simple poll' command to the validator and checking for a valid cctalk message response, i.e. the correct number of returned bytes, a valid internal message structure and a valid checksum.

Send : [2] [0] [1] [254] [255]

Reply : [1] [0] [2] [0] [253] = valid cctalk peripheral

The reply codes (if any) for a HI² peripheral are currently unknown.

16 Credit Polling

The following serial credit codes are returned for the GB coin set :

No.	Coin	Code
1	£1	1
2	50p	2
3	20p	3
4	10p	4
5	Token	5
6	£2	6
7	5p	7
8	-	8
9	Extra 1	9
10	Extra 2	10
11	Extra 3	11
12	Extra 4	12
13	Extra 5	13
14	Extra 6	14
15	Extra 7	15
16	Extra 8	16

17 Initial Settings

In serial mode, the initial settings for coin inhibits, sorter overrides and token selection do not come from the physical switches and connectors, but from a ‘virtual’ control state stored in the validator.

At power-up, the initial conditions are :

17.1 Inhibits

Initialised to [11111111 - 11111111]

i.e. All coins enabled

Note that it is possible for a different set of default inhibits to be used if previously stored into EEPROM. See the ‘Configuration to EEPROM’ command for details of how to do this.

17.2 Overrides

Initialised to [01111111]

i.e. No overrides active

17.3 Token Selection

Initialised to [12]

i.e. The ‘teach-and-run’ token

18 Command List

The following commands are available for use...

Header	Function
254	Simple poll
253	Address poll to determine attached devices
252	Address clash to determine if any devices have the same address
251	Address change to change the address of the attached device
250	Address random to randomly change the address of a device
249	Request polling priority
248	Request status
246	Request manufacturer id
245	Request equipment category id
244	Request product code
243	Request database version
242	Request serial number
241	Request software revision
240	Test solenoids
238	Test output lines
236	Read opto states
233	Latch output lines

232	Perform self-check
231	Modify inhibit status
230	Request inhibit status
229	Read buffered credit or error codes
228	Modify master inhibit status
227	Request master inhibit status
226	Request insertion counter
225	Request accept counter
222	Modify sorter override status
221	Request sorter override status
219	Enter new PIN number
218	Enter PIN number
216	Request data storage availability
215	Read data block
214	Write data block
213	Request option flags
212	Request coin position
210	Modify sorter paths
209	Request sorter paths
205	Request audit information block
202	Teach mode control
201	Request teach status
199	Configuration to EEPROM
198	Counters to EEPROM
197	Calculate ROM checksum
196	Request creation date
195	Request last modification date
194	Request reject counter
193	Request fraud counter
192	Request build code
189	Modify default sorter path
188	Request default sorter path
184	Request coin id
181	Modify coin security setting
180	Request coin security setting
179	Modify bank select
177	Handheld function
176	Request alarm counter
173	Request thermistor reading
4	Request comms revision
3	Clear comms status variables
2	Request comms status variables
1	Reset device

19 Command List Explained

The following section contains information **in addition** to that given in the generic specification. Not all commands are 'explained'.

Request polling priority

Returns [2] [20] to indicate a recommended credit polling interval of less than or equal to 200ms. Polling at greater than this interval could possibly lead to lost credits (coin swallowing) during fast coin insertion.

Request status

Refer to the status code table in this document for a list of possible status codes.

Request manufacturer id

Returns 'Coin Controls Ltd'.

Request equipment category id

Returns 'Coin Acceptor'.

Request product code

Returns 'C435S'.

Request database version

Returns [0] to indicate that a remote coin database is not available for this product.

Request serial number

Returns 3 bytes, range 0 to 16,777,215.

Request software revision

Returns 'NL-X.XX', where X is any digit 0 to 9.

Test solenoids

The solenoid bit mask is defined as follows :

B0 - Accept gate

B1 - Sorter 1

B2 - Sorter 2

B3 - Manifold

The solenoids operate for 500ms.

Test output lines

Note : Although used on serial, this command is used to test the parallel output lines on the coin acceptor.

The output bit mask is defined as follows :

- B0 - Accept line 1
- B1 - Accept line 2
- B2 - Accept line 3
- B3 - Accept line 4
- B4 - Accept line 5
- B5 - Accept line 6

The accept lines are pulsed for 500ms.

It may be the case that accept line 6 is left on after this command finishes - this is perfectly normal and reflects the parallel credit mode that has been set.

In BACTA binary mode (16 coin mode) , accept line 6 is always on.
In parallel mode (6 coin mode) , accept line 6 is normally off.

Read opto states

The opto states are returned as follows :

- B0 - D.C.E. coin opto
- B1 - D.C.E. token opto
- B2 - Sorter optos
- B3 - C.O.S. switch opto

D.C.E. = Dual Coin Entry rundown

C.O.S. = Coin-on-string

A logic 1 indicates either the opto is blocked or the opto is missing. All these optos are build options and it may be the case that none have been fitted.

Latch output lines

Note : Although used on serial, this command is used to test the parallel output lines on the coin acceptor.

The bit mask is identical to the 'Test output lines' command, but the change is permanent - no timer is used to cancel the settings.

The polarity is similar to 'Test output lines', i.e. a logic 1 turns the accept line on.

Perform self-check

Refer to the fault code table in this document for a list of possible fault codes.

Modify inhibit status

The validator supports the full 16 coin windows shown in the generic specification.

❖ Changes to inhibit status are stored in RAM and are lost at power-down or reset.

Request inhibit status

The validator supports the full 16 coin windows shown in the generic specification.

Read buffered credit or error codes

This command should be sent to the validator on a frequent basis to avoid losing valuable credit information - this is a similar process to polling the parallel accept lines for credits.

The event counter byte is followed by a 10 byte credit buffer which is used to record the last 5 events. Each event (a credit or error) takes 2 bytes.

The format is as follows :

Credit Event

[credit code] [sorter path] ← 2 bytes per event

The range of the credit codes is 1 to 16.

The range of the sorter paths is 1 to 8.

Error Event

[0] [error code] ← 2 bytes per event

Refer to the error code table in this document for a list of possible error codes.

For the GB coin specification, the following serial credit codes have been assigned :

Coin	Credit Code
£1	1
50p	2
20p	3
10p	4
Token	5
£2	6
5p	7
-	8

The sorter path number reflects the standard 8-way sorting path notation used by Coin controls in a number of 'system' products.

To understand what the numbers mean when only a 4-way sorter module is fitted, refer to the table below.

Sorter Paths 1 to 8	Sorter Paths A to D	Manifold Drive ?
1	D	yes
2	C	yes
3	B	yes
4	A	yes
5	A	no
6	B	no
7	C	no
8	D	no

When routing with a 4-way module, whether the manifold drive is active or not is irrelevant, e.g. there is no difference between sorter path 3 and sorter path 6.

Remember that when using overrides, only paths 1 to 7 can have an active override. In parallel mode, 7 override pins are supported as path 8 is considered the default.

Event Counter

Remember that after 255, the event counter wraps around to 1, not 0. The only way for the event counter to be 0 is after a power-up or reset.

Pseudo-code :

```

If n = 0 then
    'power-up' detected
else
    n = eventCounter - lastEventCounter
    If n < 0 Then n = n + 255

```

Modify master inhibit status

❖ Changes to master inhibit status are stored in EEPROM and are retained at power-down or reset.

If the coin acceptor has its master inhibit status set, it is completely inhibited from accepting coins or performing security operations. No events are generated when using the 'Read buffered credit or error codes' command, but all serial requests for information can be responded to. It is like a partial shutdown mode.

This command cannot be used dynamically as coins are rolling through the validator as the response time is too slow. If a fast inhibit action is required then use the 'Modify inhibit status' command.

This command is protected by a PIN number such that sending this command without first issuing the correct pin number will result in the validator not responding (no ACK message). The default PIN number is not listed in this document - please contact Money Controls if you feel this 'shutdown mode' would be useful.

A validator that has the master inhibit status set would pass a self-test but not be able to accept any coins - be warned ! It would be possible to set a 'security lock' on a validator by sending a valid PIN number, setting the master inhibit status and then changing the PIN number to a new, secret one. This would prevent anybody else from using it.

Request insertion counter

A 3 byte insertion counter is supported in the validator with a range 0 to 16,777,215.

The insertion counter indicates how many coins (or tokens) were inserted into the validator, regardless of whether they accepted.

Reading the insertion counter does not reset it - it provides a running total of the coins inserted since the product was manufactured.

Request accept counter

A 3 byte accept counter is supported in the validator with a range 0 to 16,777,215.

The accept counter is only incremented if a coin generates a valid signal on the credit sensor and meets a strict timing window. Therefore, it is possible for there to be more coins in the cash-box than recorded by this counter, but it should tie up precisely with the number of credits generated.

Reading the accept counter does not reset it - it provides a running total of the coins accepted since the product was manufactured.

Modify sorter override status

❖ Changes to override status are stored in RAM and are lost at power-down or reset. **For this command to work correctly, the validator must be in EEPROM routing mode - see section 7.4.**

Request data storage availability

Returns [2][2][8][2][8]

This means the memory available for customer use is implemented in permanent but limited use memory, i.e. EEPROM.

A total of 16 bytes is available for use.

2 blocks of 8 bytes are available for reading.

2 blocks of 8 bytes are available for writing.

No restriction is placed on the type of data that can be stored. Product operation is not altered in any way by the data stored.

Request option flags

Returns [0]

Bit 0 is zero which means the serial credit codes report coin position rather than coin value.

Request coin position

The value of credit code sent to the validator should be 1 to 16.

The position of this coin within the internal window space is reported back as a 16-bit mask. This command is only really of use where credit codes have been changed from their default values, and the new inhibit position is required. The inhibit position tracks the window position rather than the credit code.

Credit Code	Default Coin Position
1	00000000 00000001
2	00000000 00000010
3	00000000 00000100
4	00000000 00001000
5	00000000 00010000
6	00000000 00100000
7	00000000 01000000
8	00000000 10000000
9	00000001 00000000
10	00000010 00000000
11	00000100 00000000
12	00001000 00000000
13	00010000 00000000
14	00100000 00000000
15	01000000 00000000
16	10000000 00000000

Modify sorter paths

In line with C435, C435S has 16 coin windows, each with 4 possible sorter routes. The first sorter route is used unless there is an override present on that path, in which case the second route is used. Eventually, the coin is diverted down the default path.

Send the coin position as 1 to 16

Send 4 sorter paths as 1 to 8

❖ Changes to the sorter paths are stored in EEPROM and are retained at power-down or reset. For this command to be of use, the validator must be in EEPROM routing mode - see section 7.4.

This command cannot be used dynamically as coins are rolling through the validator as the response time is too slow. If a change in sorter paths is required then use the 'Modify sorter override status' command.

Request audit information block

The format of data returned has been kept the same as another Coin Controls product, most fields of which are unsupported here. So the return data is mostly zeros.

[0] x 9 bytes
 [accept counter] x 3 bytes
 [reject counter] x 3 bytes
 [0] x 14 bytes

As the accept counter and reject counter are available through other cctalk commands, there is no requirement to use this command. Future products may pad this return data block out with something of interest.

Teach mode control

The coin position transmitted is defined as follows :

1 to 16	Coin Window
17	Token window

The token window is the same as position 'C' of the rotary switch, i.e., token window 12. Note that token windows 1 to 11 are factory pre-programmed and cannot be altered through a serial teach operation.

Request teach status

The 'teach completed' reply code is returned after 8 valid coin insertions. There is filter algorithm built into the teach software which rejects duff coin readings - typically those obtained from inserting bent or dirty coins, or by inserting a different type of coin to those in the set. If a coin is rejected in this manner during the teach process, the coin counter does not change.

Note that the first coin in the teach sequence must be a good one else all other coins entered will be ignored. To break out of this loop, send the abort code option of this command.

Configuration to EEPROM

The only configuration data that can be stored in EEPROM with this command is the inhibit status. As described in 'Modify inhibit status' command, this data is normally lost at power-down or reset.

This command can be used to save the inhibit status which is then restored at the next power-up.

Counters to EEPROM

This command can be used to write the insertion, accept and fraud counters into EEPROM. Although this is done automatically after coin insertion, it can be done prior to switching the power-off to give greater counter accuracy.

The counters are written unconditionally - even if a previous checksum error has been reported. Refer to the 'Request insertion counter' command for more details.

Calculate ROM checksum

The ROM checksum method is a straight addition of the code bytes into an unsigned long - 4 bytes or 8 hex digits. Due to the limited code space of the microcontroller used on this product, the most significant byte returned will be zero.

The ROM checksum is not defined in this document as prior to masking it is liable to change. For determining which revision of software is in use, refer to the 'Request software revision' command.

Request creation date

The PRODUCT_BASE_YEAR as defined in the generic specification is defined as the year 1998.

Request last modification date

The PRODUCT_BASE_YEAR as defined in the generic specification is defined as the year 1998.

Request reject counter

A 3 byte reject counter is supported in the validator with a range 0 to 16,777,215.

The reject counter is calculated dynamically by subtracting the accept counter from the insertion counter. The reject counter is not stored anywhere in the product.

For continuous monitoring of acceptor performance, it is recommended that the reject counter is not read with this command - it is handled internally as a low priority task and the response can be delayed by coin validation. It is much better to read the insertion and accept counters instead.

Request fraud counter

A 3 byte fraud counter is supported in the validator with a range 0 to 16,777,215.

The fraud counter is only incremented if the validator recognises a coin as a fraud. Fraud coins are always rejected.

A fraud coin for C435S is defined in 2 ways :

- A coin that is probably the same physical size as the true coin but fabricated from lead or solder. The product makes use of a piezo sensor to detect this material.
- A coin that has been pre-programmed into the validator (occupying 1 of the 16 coin windows) and marked as a fraud coin. This is useful for determining which sites are being targeted with a batch of fraud coins.

Reading the fraud counter does not reset it - it provides a running total of the coins deemed frauds since the product was manufactured.

Request build code

The product build code is returned as 8 ASCII characters.

The significance of fields within the build code is currently undefined.

Modify default sorter path

The default sorter path can be specified as 1 to 8.

The default for 4-way routing is 5 (path A).

Request coin id

There is limited supported for coin identification strings in the current product. Each coin window, 1 to 16, has a 6 character string associated with it, but this string is either 'Coin' or 'Token', right-padded with spaces.

Since C435S supports the existing C435 operating modes, some coin windows may be replaced by token windows which have their own window space. Typically, if the product is operating in industry-standard mode, coin window 5 is replaced by a token window and the coin id will reflect this.

There is no way to modify the coin identification string.

Modify coin security setting

The coin position is 1 to 16, the security setting as described in the generic specification. Note that the upper and lower coin banks share the same security setting, so the security setting for coin position 1 is the same as coin position 9... and coin position 8 is the same as coin position 16.

Typical values for the security setting are given below.

253	Coin acceptance	⊗ ⊗ ⊗
254	Coin acceptance	⊗ ⊗
255	Coin acceptance	⊗
0	Default setting	-
1	Fraud rejection	⊗
2	Fraud rejection	⊗ ⊗
3	Fraud rejection	⊗ ⊗ ⊗

If the coin acceptance is made too high, then some coins will return 'Multiple window error' as the windows all start to overlap with each other.

If the fraud rejection is made too high, then some coins will return 'Reject coin' far too often as the windows become too tight.

It is recommended that the security setting is left at the default value (zero) unless there is a specific fraud problem. The factory set-up procedure is carefully optimised for the best compromise between coin acceptance and fraud rejection, and takes into account a true coin population distribution based on sound statistical methods.

❖ Changes to security settings are stored in RAM and are lost at power-down or reset.

The security settings have been made volatile so that the precision factory set-up cannot be permanently altered. It is up to the host machine to store the security settings when power is removed and replace them at the next power-up.

If it is desirable that the security settings are stored in the validator rather than the host machine, especially where validators are being swapped about, then use can be made of the 'Read data block' and 'Write data block' commands. This provides the machine software with 16 bytes of local storage - 8 of which can be used to store the security settings on coins 1 to 8.

Modify bank select

Although in parallel mode the validator operates with separate bank select switches as well as individual coin inhibits, a bank select system is not supported in serial mode. The modify bank select command is used to directly control the 16-bit inhibit register used by the 'Modify inhibit status' and 'Request inhibit status' commands.

Bank No.	MSB - Coin Inhibits - LSB	Comments
0	11111111 11111111	All coins enabled
1	00000000 11111111	Coins 9 to 16 disabled
2	11111111 00000000	Coins 1 to 8 disabled

Values of bank number outside the range 0 to 2 are ignored but an ACK message is still returned.

This command is slightly quicker than modifying the inhibit status as only 1 data byte is needed.

There is no mechanism to read bank the bank select number - it is not stored internally.

Handheld function

Some miscellaneous functionality has been ‘mopped up’ in the handheld function, designed for use in an external toolkit.

Mode	Function	Description
0	0	Read rotary switch position
0	1	Modify rotary switch position
0	2	Read routing plug
0	3	Read parallel inhibits
0	4	Binary to ASCII converter
0	5	Fix checksum
0	6	Solenoid life test

Read rotary switch position

This command is used in serial mode to read the ‘virtual’ rotary switch position. A number 1 to 12 is returned. The rotary switch controls which tokens are accepted.

It **cannot** be used in parallel mode to determine the actual rotary switch position.

Modify rotary switch position

This command is used in serial mode to modify the ‘virtual’ rotary switch position. Send a number 1 to 12 and an ACK is returned.

It **cannot** be used in parallel mode to change the actual rotary switch position.

Read routing plug

This command can be used to test the routing plug connector. Send a window code 1 to 7 and receive the routing pattern as a single byte bit mask. Bit 7 of the routing pattern is always zero. Bits 6 to 0 represent sorter paths 7 to 1 respectively. See the ‘Read buffered credit or error codes’ command for details of how the sorter numbers relate to sorter path letters.

The 'BACADAA' routing plug example...

Send	Receive R7 to R1	Path Number	Path Letter
1	0 0000100	3	B
2	0 0000000	-	A
3	0 0000010	2	C
4	0 0000000	-	A
5	0 0000001	1	D
6	0 0000000	-	A
7	0 0000000	-	A

A = default path

This function is unlikely to be used outside the factory.

Read parallel inhibits

Similar to the 'Request inhibit status' command but designed to work in parallel mode rather than serial mode. The resulting inhibit pattern takes into account the various operating modes and the BACTA standard.

This is a useful debugging tool.

Binary to ASCII converter

This is a 'helper' function for serial toolkits using LCD displays. It allows a 3 byte unsigned binary number to be converted into an ASCII string, and in the correct order for displaying. Leading zeros are automatically suppressed.

By example...

[0] [0] [0] returns [48] = 0

[255] [0] [0] returns [50] [53] [53] = 255

[255] [255] [255] returns [49] [54] [55] [55] [55] [50] [49] [53] = 16777215

A typical use would be to convert the 3 byte electronic serial number into a format suitable for displaying.

Fix checksum

This command allows the EEPROM checksum to be fixed. If a 'Perform self-check' command returns the 'EEPROM checksum corrupted' fault then this command can be used to fix the checksum. The question remains, of course, why was the checksum corrupted in the first place? The most likely explanation is powering off in the middle of a write sequence.

Solenoid life test

This command starts a continuous pulse test on any solenoids fitted to the product. This may include the accept gate, both sorter solenoids and the active manifold solenoid. As soon as the command is sent, an ACK is returned and the sequence begins. The only way to cancel the sequence is a software reset - see the 'Reset device' command, or by removing power. This means once the test is underway, the host interface equipment can be removed and used elsewhere.

The solenoid duty cycle is 50%, with 500ms on-time and 500ms off-time.

Request alarm counter

The product supports a simple 1 byte alarm counter to record the following events :

- C.O.S. mechanism activated (coin-on-string)
- Coin going backwards
- Credit sensor blocked
- Sorter opto blocked
- DCE opto timeout
- Credit sensor reached too early
- Credit sensor timeout

After the 255th event, the counter wraps round to zero.

Reading the counter automatically resets it to zero. To track the total number of alarm events, the host machine needs to accumulate counter values.

Note that all the above events are logged anyway in the 'credit buffer' accessed with the 'Read buffered credit or error codes' command and so this feature may have limited use.

Request thermistor reading

As stated in the generic specification, the thermistor reading should only be used as an approximate guide to temperature.

Request comms revision

Returns [1] [3] [0].

The cctalk level in this first AWP product is defined as 1. If extra functionality is added in future, this level number will be incremented to give host software a fast way of determining command support.

The communication specification documentation required for this product is at revision 3.0.

Request comms status variables

Note that the 3rd byte of the 3 byte return data could be of interest when testing the serial link. It reports the number messages received by the validator with bad checksums. This should always report back zero. Check for noise on the data line if errors are occurring.

Reset device

It is possible to perform a software reset of the validator using this command.

An ACK message is returned prior to the reset, and the reset is delayed if outputting a credit code on the parallel accept lines.

Although the reset is performed in software, a hardware reset pulse is produced on the PCB, resetting all the peripheral devices.

There should be no need to perform a software reset under normal operating conditions. There is additional hardware built into the product to ensure a clean start-up and a fast switch-off when the supply voltage crosses certain threshold voltages.

20 PIN Protection

The only command which is currently PIN-protected is the 'Modify master inhibit status' command. All other commands should work as described.

21 Command Response Timing

The response times in the table below are measured from the first start bit of the transmit message to the first start bit of the receive message. Since the serial message handling routine shares processing time with the coin validation tasks, the times are subject to wide variation, but **typical maximums** are shown.

Commands marked with a '*' are executed at a much higher priority and can be performed in the middle of coin validation without incurring extra delays. Other commands can have responses delayed by up to 300ms (over and above times listed in the table) if sent **while a coin is being validated**.

Commands marked with a 'w' perform EEPROM write operations and so care must be taken not to turn off the supply voltage before the command is completed.

Host software only needs the concept of 'timeouts' to avoid waiting indefinitely on the serial bus in the event of a hardware fault. For instance, the validator may not have been plugged into the machine ! The timeouts can be made suitably generous and given the same value for most commands e.g. 1 or 2 seconds. The only reason to shorten the timeouts would be when connecting to a number of validators in a multi-drop application. If credits are being polled in a time-critical fashion, it is undesirable for a fault on one validator to result in 'coin swallowing' on another.

Function	Key	Response Time
----------	-----	---------------

Simple poll		< 30ms
Address poll to determine attached devices		variable < 1500ms
Address clash to determine if any devices have the same address		variable < 1500ms
Address change to change the address of the attached device	w	< 120ms
Address random to randomly change the address of a device	w	< 120ms
Request polling priority		< 30ms
Request status		< 30ms
Request manufacturer id		< 30ms
Request equipment category id		< 30ms
Request product code		< 30ms
Request database version		< 30ms
Request serial number		< 30ms
Request software revision		< 30ms
Test solenoids		< 550ms
Test output lines		< 550ms
Read opto states		< 30ms
Latch output lines		< 30ms
Perform self-check		< 60ms
Modify inhibit status	*	< 10ms
Request inhibit status	*	< 10ms
Read buffered credit or error codes	*	< 10ms
Modify master inhibit status	w	< 120ms
Request master inhibit status		< 30ms
Request insertion counter	*	< 10ms
Request accept counter	*	< 10ms
Modify sorter override status	*	< 10ms
Request sorter override status	*	< 10ms
Enter new PIN number		< 35ms
Enter PIN number		< 30ms
Request data storage availability		< 30ms
Read data block		< 30ms
Write data block	w	< 50ms
Request option flags		< 30ms
Request coin position		< 30ms
Modify sorter paths	w	< 50ms
Request sorter paths		< 30ms
Request audit information block		< 30ms
Teach mode control		< 30ms
Request teach status	*	< 10ms
Configuration to EEPROM	w	< 40ms
Counters to EEPROM	w	< 50ms
Calculate ROM checksum		< 1300ms
Request creation date		< 30ms
Request last modification date		< 30ms
Request reject counter		< 30ms
Request fraud counter	*	< 10ms
Request build code		< 30ms

Modify default sorter path	w	< 40ms
Request default sorter path		< 30ms
Request coin id		< 30ms
Modify coin security setting		< 30ms
Request coin security setting		< 30ms
Modify bank select		< 30ms
Handheld function		See below
Request alarm counter		< 30ms
Request thermistor reading		< 30ms
Request comms revision		< 30ms
Clear comms status variables		< 30ms
Request comms status variables		< 30ms
Reset device	*	< 10ms

Handheld function :

Function	Key	Response Time
Read rotary switch position		< 30ms
Modify rotary switch position		< 30ms
Read routing plug		< 30ms
Read parallel inhibits		< 30ms
Binary to ASCII converter		< 40ms
Fix checksum	w	< 100ms
Solenoid life test		< 30ms

Key

- * high priority command, very fast response
- w EEPROM write cycle

22 Data Lengths

The 'data out' (from host machine to validator) and 'data in' values are the number of bytes in the data part of the message, not the overall message length. The complete message contains an additional 5 bytes for the header, addresses, length and checksum.

Function	Data Out	Data In
Simple poll	0	0
Address poll to determine attached devices	0	special
Address clash to determine if any devices have the same address	0	special
Address change to change the address of the attached device	1	0
Address random to randomly change the address of a device	0	0
Request polling priority	0	2
Request status	0	1
Request manufacturer id	0	17
Request equipment category id	0	13
Request product code	0	5
Request database version	0	1
Request serial number	0	3
Request software revision	0	7
Test solenoids	1	0
Test output lines	1	0
Read opto states	0	1
Latch output lines	1	0
Perform self-check	0	1 or 2
Modify inhibit status	2	0
Request inhibit status	0	2
Read buffered credit or error codes	0	11
Modify master inhibit status	1	0
Request master inhibit status	0	1
Request insertion counter	0	3
Request accept counter	0	3
Modify sorter override status	1	0
Request sorter override status	0	1
Enter new PIN number	4	0
Enter PIN number	4	0
Request data storage availability	0	5
Read data block	1	8
Write data block	9	0
Request option flags	0	1
Request coin position	1	2
Modify sorter paths	5	0
Request sorter paths	1	4
Request audit information block	0	29
Teach mode control	1	0
Request teach status	1	2
Configuration to EEPROM	0	0

Counters to EEPROM	0	0
Calculate ROM checksum	0	4
Request creation date	0	2
Request last modification date	0	2
Request reject counter	0	3
Request fraud counter	0	3
Request build code	0	8
Modify default sorter path	1	0
Request default sorter path	0	1
Request coin id	0	6
Modify coin security setting	2	0
Request coin security setting	1	1
Modify bank select	1	0
Handheld function	variable	variable
Request alarm counter	0	1
Request thermistor reading	0	1
Request comms revision	0	3
Clear comms status variables	0	0
Request comms status variables	0	3
Reset device	0	0

Handheld function :

Function	Data Out	Data In
Read rotary switch position	1	1
Modify rotary switch position	2	0
Read routing plug	1	1
Read parallel inhibits	1	2
Binary to ASCII converter	4	1 to 8
Fix checksum	1	0
Solenoid life test	1	0

22.1 Buffer Sizes

If the transmit and receive messages are stored in RAM prior to sending and after receiving (though this need not be the case if memory is tight), space needs to be reserved for the ‘worst case’ commands.

The total size of the transmit buffer = 5 + 9 = **14 bytes**

If we ignore the ‘Request audit information block’ command...

The total size of the receive buffer = 5 + 17 = **22 bytes**

23 Error Code Table

Code	Error
1	Reject coin
2	Inhibited coin
3	Multiple window (ambiguous coin type)
6	Credit sensor timeout
8	2 nd close coin error (coin insertion rate too high)
14	Credit sensor blocked
15	Sorter opto blocked
17	Coin going backwards
20	C.O.S. mechanism activated (coin-on-string)
21	DCE opto timeout
22	DCE opto not seen
23	Credit sensor reached too early
24	Reject coin (repeated sequential trip)
25	Reject slug
254	Coin return mechanism activated (flight deck open)

24 Error Codes Explained

‘Auto-lockout on Error’ Feature

Error codes which signify a permanent alarm code or coin jam situation are issued once and the validator is automatically ‘locked out’ until the error condition is cleared. The host software does not have to send a ‘Modify inhibit status’ command to perform this function.

The following error codes are available on the serial interface.

Reject coin

If a coin is not a recognised type, then this error is generated. The coin will be physically rejected from the validator. This coin may be a fraud coin, an un-programmed coin or a true coin just outside the normal population curve.

Inhibited coin

If a coin would have been accepted if it wasn't for the inhibit status, then this error is generated.

Multiple window (ambiguous coin type)

It is possible for a coin to fall within 2 coin windows, i.e. the validator can't determine whether the coin is type 1 or type 2. This occasionally occurs in currencies with a number of physically similar coins and slightly overlapping windows are programmed into memory. Coin specification issues such as this are routinely dealt with at the factory.

If this error is occurring all the time for a particular coin, check that the same coin has not been programmed twice into the validator or that for high / low security coin specifications, the bank select switches are in the correct state.

Credit sensor timeout

This error is generated if a coin is accepted (the accept gate opens) but does not reach the credit sensor. This could be a possible fraud attempt or a fault with the credit sensor. Note that a credit code is only generated for a coin which gives the correct signal on the credit sensor.

2nd close coin error (coin insertion rate too high)

If the coin insertion rate is too high, the validator switches to an 'inhibited' state and rejects coins. This message is generated for each coin that is seen separately but too close to the one in front. If touching or near-touching coins are put down, there may be less messages sent out than coins going in.

Credit sensor blocked

This message is issued if a blockage is seen at the credit sensor. In parallel mode, the error code is continuously present until the blockage is cleared. In serial mode, the error message is issued once on detection. While this error condition occurs, further coin acceptance is not possible.

Sorter opto blocked

This message is issued if a blockage is seen at the sorter optos. In parallel mode, the error code is continuously present until the blockage is cleared. In serial mode, the error message is issued once on detection. While this error condition occurs, further coin acceptance is not possible.

Coin going backwards

If a coin is detected going backwards through the coil area of the rundown (an obvious fraud attempt) then this message is generated.

C.O.S. mechanism activated (coin-on-string)

If a C.O.S. switch is fitted to the product (a build option), then this error is generated if a fraud attempt through stringing is seen.

DCE opto timeout

If the DCE (Dual Coin Entry rundown) is fitted and a coin or token is seen on the entry optos but not in the validator then this error is generated. There could be a blockage in the rundown somewhere.

DCE opto not seen

This is the converse of 'DCE opto timeout'. If the DCE is fitted and a coin or token is seen in the validator but not previously by the entry optos then this error is generated.

Credit sensor reached too early

A coin reaching the credit sensor too quickly is seen as suspicious and a credit code is not issued. This error is sent out instead.

Reject coin (repeated sequential trip)

This is a special message which is issued if there have been a number of sequential reject messages. The trip level has been set at 5 so if there have been 5 sequential rejects, this message replaces the normal reject error message.

5 sequential rejects is statistically very uncommon (1 in 10 million by chance on a fully working validator), and so there is either a fraud attempt or there is a fault with the validator.

This function is only available in serial mode.

Reject slug

This message can sometimes be issued instead of the more common 'Reject coin'. There are 2 ways in which it can be generated.

1. A lead or solder slug, very close to the true coin, is put down.
2. A factory pre-programmed fraud coin is detected.

Coin return mechanism activated (flight deck open)

This message is generated if the coin return lever is pressed.

25 Fault Code Table

Code	Fault
1	EEPROM checksum corrupted
2	Fault on inductive coils
3	Fault on credit sensor
4	Fault on piezo sensor
8	Fault on sorter exit sensors
20	Fault on C.O.S. mechanism
22	Fault on thermistor
34	Temperature outside operating limits
35	D.C.E. fault

26 Fault Codes Explained

EEPROM checksum corrupted

The EEPROM data is protected by a simple 8-bit checksum. This has been found to be incorrect so there is the possibility of data corruption - usually because power to the product was removed during an EEPROM write cycle.

See the 'Handheld function' command (sub-function 5) for a possible way of correcting this fault.

Fault on inductive coils

The coil number may be returned as extra information. If it isn't, there is a general frequency locking problem.

Fault on credit sensor

There is a fault on the credit sensor. Perhaps a coin jam.

Fault on piezo sensor

There is a fault on the piezo sensor. This should only affect the accept rate of coins which use the piezo sensor.

Fault on sorter exit sensors

There is a fault on the sorter optos. Perhaps a coin jam.

Fault on C.O.S. mechanism

If fitted, the C.O.S. switch is faulty. Perhaps jammed open.

Fault on thermistor

The temperature sensor is faulty. Perhaps not fitted.

Temperature outside operating limits

Based on the thermistor reading, the product seems to be outside the normal operating temperature range.

This is a fairly crude measurement, but it usually means the temperature is greater than 60°C or less than -10°C.

D.C.E. fault

The optos on the dual coin entry rundown are fault.

The opto at fault is returned as extra information :

- 1 - coin side
- 2 - token side

27 Fault Code Priority

Only one kind of fault can be reported by cctalk at any one time. Therefore, if a number of faults exist on a product, then the following table can be consulted to determine which fault is reported over and above the others. Once this fault is fixed, the next priority fault will reveal itself.

Priority	Fault	Details
1	EEPROM checksum corrupted	
2	Fault on inductive coils	Range error on sensor 1
3	Fault on inductive coils	Range error on sensor 2
4	Fault on inductive coils	Range error on sensor 3
5	Fault on inductive coils	Range error on sensor 4
6	Fault on inductive coils	Frequency locking problem
7	Fault on piezo sensor	
8	Fault on credit sensor	
9	Fault on sorter exit sensors	
10	Fault on C.O.S. mechanism	
11	D.C.E. fault	Coin opto
12	D.C.E. fault	Token opto
13	Fault on thermistor	
14	Temperature outside operating limits	

28 Status Code Table

Code	Status
1	Coin return mechanism activated (flight deck open)
2	C.O.S. mechanism activated (coin-on-string)

29 Coin Counters

The coin counters in C435S can operate in two different modes depending on the status of the EEPROM. If the EEPROM is judged to be operating correctly, the counters are non-volatile and the contents are retained after a power-down. However, if a problem is detected, the counters automatically switch to fail-safe mode and work in RAM only - after a power-up or reset the counter values are cleared to zero. In this case it is up to the host machine to store the counters in battery-backed RAM. Most applications only require the counter values for monitoring the accept rate of the validator and this can be done by looking at changes in counter values, i.e. using them in a relative rather than an absolute sense.

Although the counters are implemented in RAM for speed, they are copied automatically into EEPROM a fixed time after the last coin has been put down in a sequence of coins. This preserves the value after a power-down or reset, but reduces the duty cycle on a serial EEPROM which has a typical lifetime of 1 million erase / write cycles. The EEPROM built into the microcontroller is not used for counter storage.

At power-up or reset, if the EEPROM checksum for either the insertion counter or accept counter is incorrect, then both counters are reset to zero. The fraud counter does not have this check since it is relatively low usage and operates independently.

Prior to writing new counter values to EEPROM, the current values are read back together with the checksum. If this checksum is incorrect, it is assumed there is a problem with the EEPROM and the new values are not written - they are left unchanged. This safeguard applies to the insertion, accept and fraud counters.

If a 'Counters to EEPROM' command is issued, the insertion, accept and fraud counter values are written unconditionally to EEPROM along with their re-calculated and correct checksum values. This may be used to recover from a bad checksum situation which arose out of a power-down during a write cycle rather than an EEPROM nearing the end of its write cycle life.

Accuracy of the counters is limited by the following :

- The ability of the inductive coil system to resolve very close or touching coins. This may mean slightly more coins were put down than recorded.
- The removal of power very soon after a coin is entered. This may mean more coins were put down than recorded. The counter algorithm is not designed for applications which turn the power off and on between each coin insertion.

The insertion counter can be used in conjunction with the accept counter to monitor acceptor performance. A fall-off in accept rate over any given time period may be indicative of a fault or fraud attempt. The 'fault' may just be a need for some basic servicing or cleaning.

Specification Sheets

30 Miscellaneous Timings

Power-up time :	< 75ms
Credit code length :	80ms ± 2ms
Alarm code length :	Variable, depending on type. Usually at least 80ms.
A4 strobe delay timer :	5us
Max. coin insertion rate :	3 coins / s typical (coin size & sorter path dependent)

It is recommended that parallel credit codes are debounced for at least 30ms.

30.1 Parallel Inhibit Timing

As soon as a coin is detected coming into the validator, the current inhibit settings are 'frozen'. This means that once a coin is in the validator, an inhibit signal cannot be applied in order to not accept that coin - it is too late by then. However, the next coin entering the validator can be inhibited.

30.2 Parallel Override Timing

The same timing restriction applies to the overrides as the inhibits in EEPROM routing mode. However, in routing plug mode, the sampling of the overrides is done much later - just prior to the accept gate opening. This allows more time for an on-the-fly divert to be successful.

30.3 Serial Inhibit / Override Timing

Serial inhibits and overrides can be sent to the validator at any time, and will be successful on a particular coin if done prior to the accept gate opening.

31 Coin Sizes

Coin diameters :	15mm to 30mm
Coin thicknesses :	1.0mm to 3.3mm

Not all combinations of diameter and thickness may be possible - there are some corners knocked off the space envelope. Contact Money Controls if you have a specific coin requirement and we can confirm the suitability of the product for you.

32 Mechanical Data

The mechanical specification for the product is the same as C435. Contact Money Controls for more information.

Width : 128mm (5 inches)
Height, body only : 159mm
Depth, with sorter : 73mm

Mounting angle : $\pm 2^\circ$ from vertical in any plane

33 Electrical Data

33.1 Power Supply

Nominal voltage : +12V to +24V d.c.
Minimum voltage : +10V d.c. (+9.6V absolute minimum)
Maximum voltage : +27V d.c.
Min / Max rise time : 5ms / 500ms
Min / Max fall time : 5ms / 500ms
Ripple voltage < 120Hz : < 1V
Ripple voltage > 120Hz : < 100mV
Ripple voltage > 1KHz : < 20mV

33.2 Current Requirements

Quiescent current : < 35mA
Validation current, prior to sorting : < 500mA
Max with sorter : < 2A
Max with active manifold : < 3A

33.3 Output Drive Capability

Accept pin voltage : Within 1V of COMA
Accept pin current : < 50mA source or sink

34 Flammability Data

Main body rating : UL94-V0
Miscellaneous part rating : UL94-HB

35 Environmental Data

35.1 Temperature

Operating temperature range : 0°C to +55°C
Storage temperature range : -20°C to +70°C
Recovery time after step change : 1 hour per 20°C

35.2 Humidity

Operating humidity range : 10% to 75% RH non-condensing
Storage humidity range : 0% to 95% RH non-condensing

36 EMC Performance

36.1 Emissions

It is intended that the product will be compliant with EMC test specification EN 55022 Class B (CISPR22) for residential, commercial & light industrial use.

36.2 Immunity

It is intended that the product will be compliant with EMC test specification BS EN 50082-1 (CISPR14) for industrial use.

37 Shock / Vibration

This product is compliant with BS 2011 part 2.1 (IEC 68-2-27)

38 Cleaning

It is recommended that the rundown is cleaned with a non-abrasive cloth and mild-detergent every 100,000 coin insertions to remove any build-up of dirt.

In serial mode, the product accept rate can be monitored closely so that an early warning of service requirements can be obtained. This will minimise lost revenue.

39 Lifetime

Target : 1 million coin insertions or 10 years

Q.A. data on MTBF is not yet available. Contact the Technical Services department for the latest information.