

C120S ccTalk Protocol Manual



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.

Contents

1.	Diary of Changes	4
2.	Documentation	5
3.	Introduction	5
4.	Product Capability	6
5.	Protocol Descriptor	7
6.	Device Address	7
7.	Electrical Connections	8
7.1	Electrical Interface Details	9
7.1.1	/ DATA	9
7.1.2	/ REQUEST POLL	9
7.1.3	/ RESET	9
7.1.4	/ INHIBIT ALL	9
8.	Quick Start	10
9.	Credit / Event Signalling	11
10.	Coin Inhibits	12
10.1	Individual Coin Inhibits	12
10.2	Master Inhibit	12
10.3	Inhibits versus Serial Credit Code versus Coin Position	13
11.	Command Sequence	14
12.	Typical Command Response Times	15
13.	Key Timing Information	16
13.1	Detailed Operation	16
14.	Command List	17
15.	Event Code Table	20
15.1	Event Code Notes	20
16.	Fault Code Table	20
17.	Remote Coin Programming	21
17.1	Introduction	21
17.2	Data Format	22
17.2.1	Sub-Header 255 - Begin packet upload	22
17.2.2	Sub-Header 254 - Upload packet data	22
17.2.3	Sub-Header 253 - End packet upload & program	22
17.2.4	Sub-Header 249 - Remove coin signature	22
17.2.5	Sub-Header 248 - Request extended coin id	22
17.2.6	Sub-Header 247 - Calculate ROM secure signature	22
17.2.7	Sub-Header 245 - Remove all coin signatures	22
17.3	Programming Procedure	23
17.3.1	The '.bin' file	23
17.3.2	Erasing Coins	24
17.3.2.1	Erased Coin Identifiers	24
17.3.3	Individual Coin Filenames	24
17.3.4	DOS Stub	25
17.4	Typical Reprogramming Times	25
17.5	Coin Identifiers	25
17.6	Auto-Credit Code Mode	25
17.6.1	Credit Code Examples	26
17.6.1.1	Simple GB Specification	26
17.6.1.2	Complex GB Specification	26
17.7	Remote Programming Error Code Table	27
18.	Appendix A : Messaging Examples	28
18.1	Remote Coin Programming	28
19.	Appendix B : C120S versus C120P	29

Tables

Table 1: Coin Dimensions.....	6
Table 2: Parallel Interface.....	8
Table 3: Inhibits v Serial Credit Code v Coin Position	13
Table 4: Typical Command Response Times.....	15
Table 5: Command List	17
Table 6: Event Code Table.....	20
Table 7: Fault Code Table	20
Table 8: Coin Filename Explained.....	24
Table 9: Remote Programming Error Code Table	27
Table 10: C120S v C120P Commands.....	29
Table 11: C120S v C120P Command Responses.....	29

Figures

Figure 1: C120S Parallel Connector.....	8
Figure 2: Credit / Event Signalling	11

1. Diary of Changes

- Issue 1.0..... 10th March 2004
 - 1st Issue

- Issue 1.1..... 30th June 2004
 - Changed footer

2. Documentation

In order to interface the C120S coin acceptor, the following document will also be needed.

'ccTalk Serial Communication Protocol - Generic Specification - Issue 4.2'

The latest issue of this document in pdf format is available from the ccTalk web site at...

www.ccTalk.org

The generic specification gives background information on the ccTalk protocol. This document details the specific command set implemented on C120S.

3. Introduction

C120S is a low power coin acceptor designed for payphone applications. Power is typically supplied through the telephone cable only.

The coin acceptor requires a single +5V regulated power supply. Each coin typically consumes no more than 36mJ of power with a peak current less than 80mA.

The coin acceptor communicates through the ccTalk serial interface to a management system which handles the coin escrow system, call duration / cost calculations and coin specification updating.

4. Product Capability

C120S has memory to store 16 different coin types.

Each coin has a serial credit code in the range 1 to 16.

Any combination of 16 coins may be enabled or disabled through software.

C120S has full download capability which means new currencies can be programmed in the payphone and updates to existing coin specifications made if new fraud coins appear.

The maximum coin insertion rate is 2 coins / s due to the power budget requirements. Coins inserted faster than this will be rejected.

The validator accepts the following range of coin or token dimensions which is believed to cover at least 95% of the world's coinage:

Table 1: Coin Dimensions

Diameter	15.0 - 33.0mm
Thickness	1.2 - 3.3mm
Weight	1.5g min

The coin or token material may be any of the normal alloys that are generally used to produce coins world wide. Coins may have holes or be non circular providing that they are capable of rolling uniformly down a ramp at an angle of 15° to the horizontal.

5. Protocol Descriptor

The C120S serial protocol conforms to ccTalk b48.p0.v5.a5.d0.c3.m0.x8.e0.i2.r4.

- 4800 baud (1 start bit, 1 stop bit, no parity)
- Open-collector interface
- Nominal supply voltage +5V
- Serial data pull-up voltage +5V
- Supply sink
- Connector type 3 (10-way pin header)
- Slave device only
- 8-bit addition checksum
- No encryption
- Minor release 2
- Major release 4

Note that the supply voltage is 5V and the lower ccTalk baud rate of 4800 is chosen to conserve power.

6. Device Address

All C120S's leave the factory set to **address 2**.

The address is stored in EEPROM and can be subsequently changed with serial commands. It is not recommended that this address is changed from the default value.

7. Electrical Connections

Figure 1: C120S Parallel Connector

Industry standard interface.
Connector type: 10 pin DIL

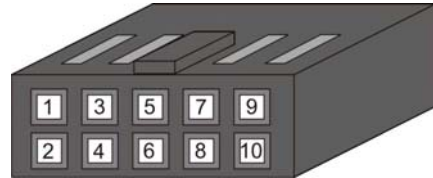
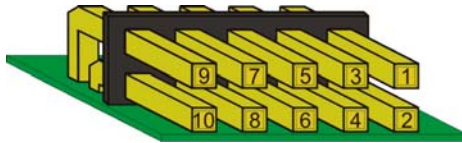


Table 2: Parallel Interface

PIN	ccTalk	Flash Reprogramming
1	/ DATA	
2	-	RxD1
3	/ REQUEST POLL	
4	-	TxD1
5	/ RESET	
6	-	/ CNVSS
7	/ INHIBIT ALL	
8	0V	
9	+5V	
10	(Solenoid 0V supply return)	

The payphone should connect through pins 1, 3, 5, 8 and 9. The reset line should be used to enable the coin acceptor only when the +5V supply is stable. Pin 7 can be used to inhibit the coin acceptor from accepting coins regardless of the serial inhibit status. Pin 10 is an optional 0V return line for the solenoid - it normally returns through pin 8.

Pins 2, 3, and 6 are used for flash programming of the microcontroller firmware and should not be connected during normal operation.

The data pin on C120S is pulled up to +5V via a **10K resistor** and should be driven with an open-collector transistor.

The /Request Poll line is bi-directional. To communicate with C120S with C120S as the slave device, pull this line to 0V. To accept coins, just monitor the voltage on this line and when a coin is inserted the C120S will pull the line to 0V for a brief period.

7.1 Electrical Interface Details

7.1.1 /DATA

The standard ccTalk data line. Open collector drive.

There is an internal pull-up of 100K rather than 10K on this line to reduce power.

7.1.2 /REQUEST POLL

Active low. Open collector drive. 100K pull-up.

The Request Poll line is bi-directional. It is used to indicate to the host controller that there is a message waiting from the validator. The validator will stay powered up until the host controller polls the message or a timeout occurs. It is also used to wake up the validator when the host controller wishes to send it a message.

7.1.3 /RESET

Active low. 100K pull-up.

Reset is an active low signal which forces the processor into a reset state for as long as the reset line is active. Minimum pulse width is 10ms.

7.1.4 /INHIBIT ALL

Active low. 1M pull-up.

Inhibit all can be used to slow down coin acceptance 'on the fly' giving the host controller time to deal with the preceding coin. The inhibit line status is read immediately prior to operation of the accept gate solenoid. The decision to accept a true coin is made at this point. A subsequent change in the inhibit line has no effect on the coin just accepted, which will continue through the defined accept process. Following coins detected by the validator while the inhibit status is active will be rejected.

Most applications do not need to use the Inhibit all line and it can be left unconnected.

8. Quick Start

If you have just received a C120S then you are probably keen to see if you can get it to work. The first step is sending and receiving ccTalk messages and the second step is accepting coins.

First of all is the electrical interface. If you are using a PC-based system with a RS232 port then you will need to construct a 'RS232 to ccTalk' converter which basically level-shifts the $\pm 12V$ of RS232 into the +5V/0V voltages of ccTalk as well as combine separate TX and RX lines into a single bi-directional data line. The circuit needed is listed as 'Circuit 4' in the generic ccTalk specification. Alternatively a suitable interface box can be ordered from Money Controls - P/N APC INT XX 00001

The coin acceptor needs to be supplied on the ccTalk connector with a regulated +5V DC supply.

Connect the PC to the coin acceptor on the 10-way ccTalk connector. Make sure that pin 3 is tied to 0V to wake-up the coin acceptor and allow serial communication to take place.

Using your chosen programming language, try sending a simple poll command to the UART port at 4800 baud with no parity and 8 data bits.

```
TX = 002 000 001 254 255
```

Note these are the actual byte values sent in decimal. Each value is in the range 0 to 255. They should be sent out in quick succession - no more than 50ms between bytes.

There should be a reply from the C120S. If not, something is seriously wrong with the hardware. This can be best looked at by connecting an oscilloscope to the ccTalk data line and actually observing the data sent back and to.

As ccTalk connects the TX line to the RX line, an identical copy of the TX message should be seen first. If not, the 'local loop-back' is not working.

```
RX = 002 000 001 254 255 001 000 002 000 253
```

The reply from the coin acceptor should be seen in the receive buffer after the transmit message.

```
RX = 002 000 001 254 255 001 000 002 000 253
```

If the reply part is missing then the coin acceptor is not working or configured incorrectly. Make sure you wait long enough for the reply to come back - in this case at least 10ms.

How do you find out where the receive message is in the receive buffer ? Easy :-

Receive message index = $5 + 2^{\text{nd}}$ byte value, assuming index starts at zero.

The data length of any ccTalk message is contained in the 2^{nd} byte and the total length is this value + 5.

If the coin acceptor returns data rather than an ACK, this can be found as well.

Receive data index = $9 + 2^{\text{nd}}$ byte value

Receive data length = contents[$6 + 2^{\text{nd}}$ byte value]

To accept coins see the next section.

9. Credit / Event Signalling

C120S signals to the host controller that a coin has been inserted or another event has occurred by pulling the Request Poll line down to 0V.

This line should be monitored by the controller and as soon as it goes low, ccTalk command 235, 'Read last credit or error code' sent to C120S to find out the event code.

The data will consist of either a single credit code, which indicates the coin type accepted, or a 2 byte event code, the first byte of which will be zero. For a list of error or event codes see the Event Code Table in this document.

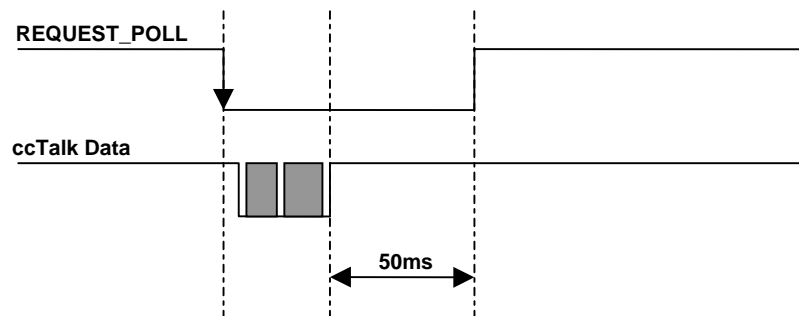
e.g.

```
TX = 002 000 001 235 018
```

```
RX = 001 001 002 000 001 251 = Credit [ 001 ]
```

```
RX = 001 002 002 000 000 001 250 = Event [ 001 ] = Reject coin
```

Figure 2: Credit / Event Signalling



The Request Poll line returns to 5V around 50ms after the coin acceptor replies to the host with the credit or error code. This allows the host to request the data again in the event of a receive checksum error or other comms problem. After the Request Poll line goes high, the host cannot ask for the data again so it is effectively lost. So when the Request Poll line goes low, the host should issue a header 235 command within a few ms and certainly no more than 50ms to prevent coins being 'swallowed'. If the host does not send a ccTalk command within 50ms, the Request Poll line will go high and the coin acceptor will be ready to accept the next coin.

10. Coin Inhibits

10.1 Individual Coin Inhibits

Each of the 16 coin positions is controlled by a bit flag in a 16-bit register. If the bit is set then the coin is enabled for acceptance but if the bit is clear it will always reject. The ccTalk header 231, Modify inhibit status, sends the 16-bit inhibit register to the coin acceptor. The LSB is sent first, controlling coins 1 to 8, followed by the MSB controlling coins 9 to 16.

The inhibit status is stored in EEPROM and is retained after power is removed.

All coins are enabled when C120S leaves the factory.

As an example, this is how to enable coins 2, 7 and 13.

```
Modify inhibit status
TX = 002 002 001 231 066 016 194
RX = 001 000 002 000 253

Coin position      87654321
LSB = 066 decimal = 01000010 binary
Coin position      87654321 + 8
MSB = 016 decimal = 00010000 binary
```

Coins 2, 7 and 13 are enabled. All other coins are disabled.

10.2 Master Inhibit

There is a 'Modify master inhibit status' command, header 228. This command acts in conjunction with the 'Modify inhibit status' command described above.

The master inhibit status is stored in EEPROM and is retained after power is removed.

If master inhibit is active, all coins are disabled regardless of the status of the individual coin inhibits. If master inhibit is not active, the enable / disable state of each coin will be given by the 'Modify inhibit status' command.

Examples of master inhibit use.

Disable coin acceptor...

Modify master inhibit status

```
TX = 002 001 001 228 000 024
RX = 001 000 002 000 253
```

Enable coin acceptor...

Modify master inhibit status

```
TX = 002 001 001 228 001 023
RX = 001 000 002 000 253
```

10.3 Inhibits versus Serial Credit Code versus Coin Position

On C120S there is a **one-to-one correspondence** between inhibit position, serial credit code and coin position for remote programming and reading coin identifiers.

Table 3: Inhibits v Serial Credit Code v Coin Position

Inhibit Position	Serial Credit Code	Coin Position
1	1	1
2	2	2
3	3	3
4	4	4
5	5	5
6	6	6
7	7	7
8	8	8
9	9	9
10	10	10
11	11	11
12	12	12
13	13	13
14	14	14
15	15	15
16	16	16

Note: Some coin specifications (e.g. with old and new coins) allocate the same serial credit code to a number of different physical coin types. This is only done when the coins have the same monetary value.

11. Command Sequence

A typical sequence of commands from the payphone is as follows...

- Phone goes off hook, power applied to mech, reset lifted
- Processor initialises and enters a low power state
- Coin detected at wake-up sensor, processor enters run state
- Coin validated and accept gate activated if true
- Coin accepts or rejects
- REQUEST_POLL line goes low for 50ms
- Host requests an event code using ccTalk command 235, 'Request last credit or error code'
- REQUEST_POLL line goes high
- Processor enters low power state

If there is a communication error when the event code is polled then the host can retry and the 50ms polling window will be restarted.

If the host polls for an event when the REQUEST_POLL line is high, there is no response.

No other commands should be sent during the polling window.

If the host needs to send other commands as soon as the phone goes off hook and before coins are entered, it can do.

As an example...

- Phone goes off hook, power applied to mech, reset lifted
- Processor initialises and enters a low power state
- Host pulls REQUEST_POLL line low
- Host sends ccTalk command 254, 'Simple poll'
- Host sends ccTalk command 248, 'Request status'
- Host sends ccTalk command 245, 'Request equipment category id'
- Host sends ccTalk command 244, 'Request product code'
- Host sends ccTalk command 192, 'Request build code'
- Host sends ccTalk command 246, 'Request manufacturer id'
- Host sends ccTalk command 242, 'Request serial number'
- Host sends ccTalk command 241, 'Request software revision'
- Host sends ccTalk command 4, 'Request comms revision'
- Host sends ccTalk command 243, 'Request database version'
- Host sends ccTalk command 231, 'Modify inhibit status'
- Host sends ccTalk command 228, 'Modify master inhibit status'
- Host releases REQUEST_POLL line
- Processor enters low power state

Note: The 'Modify inhibit status' and 'Modify master inhibit status' commands store data in non-volatile memory and so there is no need to send these commands every time the phone goes off hook. Factory units are sent out with all coins enabled.

12. Typical Command Response Times

Examples of ccTalk response times are shown in the table below. They are meant to be indicative only.

Table 4: Typical Command Response Times

Header	Description	Total TX + RX Packet Length	Typical Response Time
96:247	Calculate ROM secure signature	19	1.26 s
197	Calculate ROM checksum	14	460 ms
231	Modify inhibit status	12	13 ms
228	Modify master inhibit status	11	13 ms
232	Perform self-check	11	12 ms
246	Request manufacturer id	24	3 ms
242	Request serial number	13	3 ms
226	Request insertion counter	13	3 ms
235	Read last credit or error code	11(12)	3 ms
254	Simple poll	10	3 ms
001	Reset device	10	3 ms

The typical response time is most easily measured from the start of the final stop bit of the transmitted message to the start bit of the reply.

13. Key Timing Information

Description	Value
Power-up to Comms Ready	< 100 ms

13.1 Detailed Operation

When the phone is lifted off-hook, power is applied to C120S and the processor runs through its initialisation routine. After external memory is shadowed in RAM and the front-end electronics compensated for temperature, the processor enters a very low power stop mode. If the REQUEST_POLL line is brought low by the host then the processor starts immediately and can reply to ccTalk commands and requests. Otherwise an inserted coin will wake-up the processor and after coin acceptance or rejection the REQUEST_POLL line will be brought low to indicate to the host that it can poll for credits. 50ms after the credit poll the REQUEST_POLL line goes high and the processor re-enters stop mode. When the phone is placed back on-hook, power is removed from C120S.

14. Command List

The following table shows all the ccTalk commands currently available for customer use on C120S. **More details can be found in the generic specification.**

Any [data bytes] are shown in decimal.

MDCES = Multi-Drop Command Extension Set. These commands are only used when peripheral addresses are unknown.

Header numbers are shown in descending order by convention.

Table 5: Command List

Header	Function	Returned Data / Comments
254	Simple poll	ACK returned.
253	Address poll	MDCES support
252	Address clash	MDCES support
251	Address change	MDCES support
250	Address random	MDCES support
249	Request polling priority	[0] [255] = Hardware request poll line
248	Request status	[0] = 'OK' [1] = 'Flight deck open'
247	Request variable set	9 bytes of Money Controls internal data
246	Request manufacturer id	'Money Controls'
245	Request equipment category id	'Coin Acceptor'
244	Request product code	'C120S'
243	Request database version	[0] if no calibration support, otherwise new coin sets can be remotely programmed.
242	Request serial number	Range 0 to 16,777,215
241	Request software revision	e.g. 'GTO-F1-V2.xx'
240	Test solenoids	Bit 0 = Accept gate Pulsed for 500ms. The ACK is returned after the output pulse has finished.
237	Read input lines	For low level diagnostics only. 4 bytes returned... [Byte 1] Thermistor reading [Byte 2] Credit sensor reading [Byte 3] Zero [Byte 4] Bit 0 = Request poll line (0 = talk) Bit 1 = Wake-up sensor (0 = coin detected) Bit 2 = Not used Bit 3 = Flight deck (0 = reject pressed) Bit 4 = Master inhibit (0 = inhibited) Bits 5:7 = Not used
Header	Function	Returned Data / Comments

236	Read opto states	Bit 0 = Credit sensor (1 = blocked)
232	Perform self-check	Supported. Refer to table 3 in the generic specification for possible fault codes.
231	Modify inhibit status	Send [inhibit 1] [inhibit 2] [inhibit 1] for coins 1 to 8 [inhibit 2] for coins 9 to 16 0 = coin inhibited, 1 = coin enabled Inhibits are stored in EEPROM.
230	Request inhibit status	[inhibit 1] [inhibit 2] The inhibit status of 16 coin positions is returned.
228	Modify master inhibit status	0 = device inhibited, 1 = device enabled The master inhibit flag is stored in EEPROM. The master inhibit flag is not normally used and all C120S's which leave the factory are 'enabled'.
227	Request master inhibit status	Supported
226	Request insertion counter	< 3 byte counter > This counter records the total number of inserted coins. The counter is stored in RAM and is cleared at power-up or reset.
225	Request accept counter	< 3 byte counter > This counter records the total number of accepted coins. The counter is stored in RAM and is cleared at power-up or reset.
219	Enter NEW PIN number	ACK returned for compatibility with older products but the PIN number mechanism is not used to protect any commands.
218	Enter PIN number	ACK returned for compatibility with older products but the PIN number mechanism is not used to protect any commands.
216	Request data storage availability	[2] [2] [8] [2] [8] This indicates a small amount of EEPROM storage is available for customer use... 2 blocks of 8 bytes for reading 2 blocks of 8 bytes for writing
215	Read data block	Block number = 0 or 1
214	Write data block	Block number = 0 or 1
213	Request option flags	Bit 0 = credit code format [0] = coin position, default [1] = CVF, not supported
212	Request coin position	Supported Returns position of coins with specified ccTalk credit codes.
197	Calculate ROM checksum	A 4 byte ROM checksum of the firmware is returned to the host machine. [XX] [XX] [XX] [XX]
196	Request creation date	Supported
195	Request last modification date	Supported
194	Request reject counter	< 3 byte counter > This counter records the total number of rejected coins. The counter is stored in RAM and is cleared at power-up or reset.

Header	Function	Returned Data / Comments
193	Request fraud counter	< 3 byte counter > This counter records the total number of fraud coins. The counter is stored in RAM and is cleared at power-up or reset. Note : Fraud coins can only be counted if they are factory pre-programmed into the coin acceptor and marked as 'bad'
192	Request build code	Fixed width 8 character ASCII string e.g. 'STD01 '
185	Modify coin id	Supported 6 x ASCII characters e.g. EU200A It is not recommended that factory settings are changed. When remote programming new coins, the coin id is automatically updated.
184	Request coin id	Supported 6 x ASCII characters e.g. EU200A
182	Download calibration info	Supported.
181	Modify security setting	Supported.
180	Request security setting	Supported.
179	Modify bank select	[bank no.] 0 = default (both banks enabled) 1 = bank 1 only enabled (coins 1 to 8) 2 = bank 2 only enabled (coins 9 to 16) The action of this command is the same as using the 'Modify inhibit status' command.
173	Request thermistor reading	Supported
170	Request base year	'2003'
169	Request address mode	[132] Address is stored in EEPROM and may be changed serially (non-volatile).
162	Modify inhibit and override registers	Supported
96:255	Begin packet upload	Start remote programming
96:254	Upload packet data	Send data packets
96:253	End packet upload & program	Finish remote programming
96:249	Remove coin signature	Delete a coin signature from memory. This permanently stops a coin from being accepted rather than modifying the inhibits.
96:248	Request extended coin id	The extended coin id contains 2 extra characters at the end which reflect window security. 8 x ASCII characters e.g. EU100A-2
96:247	Calculate ROM secure signature	The ROM checksum is calculated with a 32-bit seed and uses a more secure algorithm.
96:245	Remove all coin signatures	Deletes all coin signatures from memory. Be careful - there is no 'undo' command !
4	Request comms revision	[1][4][2] ccTalk specification = 4.2 minor revision = 1
3	Clear comms status variables	Supported
2	Request comms status variables	Supported
1	Reset device	Supported This command performs a 'software reset'.

15. Event Code Table

Implemented on C120S

Table 6: Event Code Table

Code	Error
1	Reject coin
2	Inhibited coin
3	Multiple window
4	Wake-up timeout
5	Validation timeout
6	Credit sensor timeout
8	2 nd close coin error
10	Credit sensor not ready
18	Coin too fast (over credit sensor)
23	Credit sensor reached too early
24	Reject coin (repeated sequential trip)
25	Reject slug
128	Inhibited coin (Type 1)
...	Inhibited coin (Type n)
254	Coin return mechanism activated (Flight deck open)

15.1 Event Code Notes

The 'Reject coin (repeated sequential trip)' indicates 5 sequential reject coins which may possibly be a fault condition.

16. Fault Code Table

Implemented on C120S

Table 7: Fault Code Table

Code	Fault	Optional Extra Info
1	EEPROM checksum corrupted	-
2	Fault on inductive coils	Coil number
3	Fault on credit sensor	-
19	Fault on coin return mechanism	-
22	Fault on thermistor	-

17. Remote Coin Programming

17.1 Introduction

It is possible to update or completely replace the programmed coins in C120S with a different currency remotely. This means through the ccTalk serial interface and without the manual insertion of any coins. So from a central office with a server it would be possible to update thousands of pay-phones overnight.

The principal mechanism is to take a data file and send it to the C120S in ccTalk data packets. As ccTalk cannot send more than 252 data bytes per packet, it must be split into sequential chunks and sent one at a time. Each chunk is acknowledged by the coin acceptor before the next one can be sent.

Each coin in a coin specification or currency has its own data file. This can be sent to any of the 16 coin positions within C120S. A blank position can be programmed with a new coin or an existing coin overwritten with a new or updated one. Coin positions can also be erased when removing coins from a currency or replacing a currency with another containing less coins.

6 ccTalk commands are associated with remote coin programming. They are currently in the application-specific section of the ccTalk header table.

96:255 means ccTalk header 96 with a sub-header of 255. The sub-header is defined as the first ccTalk data byte. Other data can follow the sub-header where necessary.

- Header 96:255, Begin packet upload
- Header 96:254, Upload packet data
- Header 96:253, End packet upload & program
- Header 96:249, Remove coin signature
- Header 96:248, Request extended coin id
- Header 96:245, Remove all coin signatures

17.2 Data Format

Only data fields are shown here for clarity. The rest of the packet is formatted as a normal ccTalk message. The ccTalk header is 96 with a sub-header as shown (the sub-header is the first ccTalk data byte).

17.2.1 SUB-HEADER 255 - BEGIN PACKET UPLOAD

Transmitted data : <none>
Received data : ACK

17.2.2 SUB-HEADER 254 - UPLOAD PACKET DATA

Transmitted data : <variable>
Received data : ACK

17.2.3 SUB-HEADER 253 - END PACKET UPLOAD & PROGRAM

Transmitted data : [coin position]
Received data : ACK or [error code]
[coin position] = 1 to 16

It is also possible to program multiple coin positions with the same information.

Transmitted data : [coin position 1] [coin position 2] ...
Received data : ACK or [error code]

17.2.4 SUB-HEADER 249 - REMOVE COIN SIGNATURE

Transmitted data : [coin position]
Received data : ACK or [error code]
[coin position] = 1 to 16

17.2.5 SUB-HEADER 248 - REQUEST EXTENDED COIN ID

Transmitted data : [coin position]
Received data : [char 1] [char 2] [char 3] ... [char 8]
[coin position] = 1 to 16

17.2.6 SUB-HEADER 247 - CALCULATE ROM SECURE SIGNATURE

Transmitted data : [seed 1] [seed 2] [seed 3] [seed 4]
Received data : [signature 1] [signature 2] [signature 3] [signature 4]

17.2.7 SUB-HEADER 245 - REMOVE ALL COIN SIGNATURES

Transmitted data : <none>
Received data : ACK or [error code]

17.3 Programming Procedure

17.3.1 THE '.BIN' FILE

Each coin to be programmed has its data in a file with a '.bin' extension. A typical filename for a coin is **EU200A-0.bin**. The size of each coin file varies in length but will typically be between 900 bytes and 2.5K bytes.

The data inside each file is in binary format. The entire file must be transferred to C120S. No modifications to the data must be made.

Step 1. Send ccTalk command 'Begin packet upload'

TX data : < none >

RX data : ACK

Wait up to 1s for reply

Step 2. Send ccTalk command 'Upload packet data'

For line = 0 to last_line

TX data : [data 1]... [data N]

RX data : ACK

Wait up to 1s for reply

Next

The number of data bytes sent with each upload command, N, can be anywhere between 1 and 251. Loop until all the file data is sent. The last line may contain less than N bytes of data.

Step 3. Send ccTalk command 'End packet upload & program'

a) To program a single coin position

TX data : [coin position]

RX data : ACK or [error code]

Wait up to 1s for reply

[coin position]

Specify 1 to 16. The coin position is the same as the ccTalk credit code (with the odd exception - see the chapter on 'Duplicated name coin programming').

[error code]

Refer to the Remote Programming Error Code Table.

b) To program multiple coin positions with the same coin

TX data : [coin position 1] [coin position 2]...

RX data : ACK or [error code]

Wait up to 1s for reply per coin position

Step 4. Wait 200ms before sending next ccTalk command

Wait 200ms

17.3.2 ERASING COINS

To erase a coin permanently from memory on C120S then two ccTalk commands are provided. The first erases a single coin, the second erases all 16 coins.

Single Coin : Send ccTalk command 'Remove coin signature'

TX data : [coin position]
RX data : ACK or [error code]

Wait up to 1s for reply

All Coins : Send ccTalk command 'Remove all coin signatures'

TX data : < none >
RX data : ACK or [error code]

Wait up to 4s for reply

After sending an erase, wait 200ms before sending the next ccTalk command.

17.3.2.1 Erased Coin Identifiers

The corresponding coin identifier strings will be cleared to '.....' i.e. ASCII 46 dot characters.

17.3.3 INDIVIDUAL COIN FILENAMES

By example...

EU200A-0.bin

The filename uses a 8.3 convention.

Table 8: Coin Filename Explained

EU	2 character country code as listed in ISO 3166-1-A2
200	3 digit monetary value in decimal
A	Mint issue letter
0	Security code
bin	Indicates coin binary file

The security code is usually 0 but there may be some local conditions which require special treatment e.g. a very close fraud coin. Coins with a filename suffix of -1 and -2 etc. can be programmed in exactly the same way as described previously. Contact Money Controls for the latest information.

17.3.4 DOS STUB

The first 23 characters of the coin files (.bin) are in ASCII and may be viewed under DOS by typing the filename to the screen (or in Windows by opening them with Notepad).

e.g. Type EU200A-0.bin

C C120S STD 00001

= C(oin file) C120S, STD build, spec. revision 00001

17.4 Typical Reprogramming Times

Approximately 3s per coin.

All 16 coins would take less than 1 minute.

17.5 Coin Identifiers

There are two ccTalk commands which can be used to read coin identifiers.

ccTalk header 184, 'Request coin id'

ccTalk header 96:248, 'Request extended coin id'

Header 184 returns the classic ccTalk 6 character coin identifier e.g. EU200A.

Header 96, sub-header 248 returns an extended 8 character coin identifier e.g. EU200A-0 which includes the security code.

17.6 Auto-Credit Code Mode

When remote programming a coin with header 96:253, the coin position is specified. This is a value from 1 to 16. When requesting an event with header 235, an accepted coin returns a credit code equal to the coin position.

For compatibility with older products, it is possible to program the same coin into more than one coin position. If this is done then the same coin would have different credit codes which means the coin could not be accepted unless just one of those coins is enabled and the rest are inhibited. The credit code would change depending on which coin is enabled. To resolve this issue, an auto-credit code mode is implemented such that coins with the same name are allocated a credit code which is the same as the position of the lowest coin. The credit codes are updated whenever a coin is added or deleted from the coin table.

17.6.1 CREDIT CODE EXAMPLES

17.6.1,1 Simple GB Specification

Coin 1	GB0.10	Credit 1
Coin 2	GB0.20	Credit 2
Coin 3	GB0.50	Credit 3
Coin 4	GB1.00	Credit 4
Coin 5	-	Credit 5
Coin 6	-	Credit 4
Coin 7	-	Credit 7
Coin 8	-	Credit 8
Coin 9	-	Credit 9
Coin 10	-	Credit 10
Coin 11	-	Credit 11
Coin 12	-	Credit 12
Coin 13	-	Credit 13
Coin 14	-	Credit 14
Coin 15	-	Credit 15
Coin 16	-	Credit 16

17.6.1,2 Complex GB Specification

In this example, a GB £1 has been programmed into positions 4, 5 and 6. The coin acceptor software automatically allocates all these coins a credit code of 4 (the lowest coin position of this type). When GB £2 is programmed into position 7, it is allocated a credit code of 7 not 5. Credit codes 5 & 6 are no longer obtainable with this specification.

Coin 1	GB0.10	Credit 1
Coin 2	GB0.20	Credit 2
Coin 3	GB0.50	Credit 3
Coin 4	GB1.00	Credit 4
Coin 5	GB1.00	Credit 4
Coin 6	GB1.00	Credit 4
Coin 7	GB2.00	Credit 7
Coin 8	EU0.10	Credit 8
Coin 9	EU0.20	Credit 9
Coin 10	EU0.50	Credit 10
Coin 11	EU1.00	Credit 11
Coin 12	EU2.00	Credit 12
Coin 13	-	Credit 13
Coin 14	-	Credit 14
Coin 15	-	Credit 15
Coin 16	-	Credit 16

Deleting the coin in position 4 would change the 2 remaining GB £1s in positions 5 and 6 to credit code 5 - the new lowest coin position.

17.7 Remote Programming Error Code Table

Implemented on C120S

Table 9: Remote Programming Error Code Table

Error Code	Description	What Went Wrong?
255	Packet too short	Less data than expected was sent to the device during the packet upload process
254	Packet too long	More data than necessary was sent to the device during the packet upload process
253	Too much data	The device ran out of memory for storing the coin packet data! Should never happen.
252	File verification error type 1	Data corruption was detected in the coin packet.
251	File verification error type 2	Data corruption was detected in the coin packet.
250	File verification error type 3	Data corruption was detected in the coin packet or a coin signature was transferred to another mech.
249	File verification error type 4	Data corruption was detected in the coin packet.
240	Unsupported packet header	The packet upload file is not recognised.
239	Unsupported file format	The packet upload file is not supported.
238	Non-matching product name	The wrong upload file is being used. Contact MCL for the correct version for your application.
237	Non-matching variant name	The wrong upload file is being used. Contact MCL for the correct version for your application.
236	Non-matching database version	The wrong upload file is being used. Contact MCL for the correct version for your application.
235	Mech does not support calibration	Calibration was factory-disabled in this coin acceptor. No coins can be programmed remotely.
234	Mech does not allow calibration	Calibration has been disabled in this coin position. It may be possible to program other coin positions.
232	Key mis-match	The wrong upload file is being used. Contact MCL for the correct version for your application.
231	Sensor mis-match	The wrong upload file is being used. Contact MCL for the correct version for your application.
100	Illegal coin position	The coin position requested is outside the legal range for the device.

18. Appendix A : Messaging Examples

18.1 Remote Coin Programming

To replace all coins with new ones...

Remove all coin signatures = ACK

TX = 002 001 001 096 245 167

RX = 001 000 002 000 253

Begin {

Upload packet data = ACK

TX = 002 001 001 096 255 157

RX = 001 000 002 000 253

Upload packet data = ACK

TX = 002 250 001 096 254

```
067 032 083 082 053 105 032 032 032 032 083 084 068 032 032 032 032 032 048 048
048 048 049 026 000 003 225 168 034 086 217 246 119 082 026 062 068 010 084 033
166 140 249 214 105 116 129 222 248 042 033 003 111 156 037 002 053 096 251 239
097 187 233 046 037 043 160 236 231 170 085 206 072 158 165 049 021 228 011 071
193 139 072 141 190 109 096 199 227 005 159 033 095 077 136 049 212 065 049 128
036 059 191 036 118 162 199 113 046 221 018 034 238 154 170 250 023 086 066 050
005 059 091 074 128 069 049 051 146 253 202 096 223 019 075 156 085 158 032 147
220 175 124 184 075 078 208 218 165 152 047 084 221 060 093 032 033 128 012 240
053 117 105 023 050 236 193 102 106 227 166 168 017 186 088 064 179 125 008 089
145 051 224 167 204 163 030 246 054 121 020 249 027 126 181 008 161 238 083 246
199 042 058 067 036 203 158 014 170 207 177 207 234 224 134 106 181 113 232 031
211 121 031 104 009 186 219 230 056 142 210 044 186 197 214 255 067 212 000 092
032 206 002 020 148 096 086 233 122 (249 bytes of data in this example)
163
```

RX = 001 000 002 000 253

n x Upload packet data = ACK (send next data blocks in sequence)

Upload packet data = ACK

TX = 002 233 001 096 254

```
029 158 060 155 161 009 248 021 038 198 082 229 102 066 132 003 164 143 041 066
187 035 247 245 072 148 089 140 208 052 065 196 131 045 224 253 175 067 243 107
217 229 014 215 011 009 053 099 157 187 068 185 246 204 232 117 231 164 013 144
008 243 070 208 144 098 027 228 184 108 170 002 175 175 155 213 204 141 015 047
087 040 213 003 218 236 119 208 092 009 017 146 234 090 050 131 211 149 084 056
205 106 234 018 255 024 115 248 216 150 000 106 181 160 031 242 212 127 063 028
041 122 057 117 046 113 219 102 079 139 016 116 250 103 229 230 148 178 167 194
075 025 005 122 056 122 017 086 252 195 160 040 243 099 151 231 176 191 010 052
148 253 141 224 073 019 234 147 001 226 115 166 186 166 220 083 182 096 164 132
014 198 193 204 159 065 049 115 129 058 145 103 179 234 014 171 117 117 161 068
164 018 192 126 151 120 097 047 040 115 204 057 230 007 093 237 034 014 104 177
176 137 084 025 022 013 021 158 139 165 029 004 (finally 232 bytes of data in
this example)
```

165

RX = 001 000 002 000 253

End packet upload & program = ACK

TX = 002 002 001 096 253 001 157 (program coin position 1)

RX = 001 000 002 000 253

Wait 200ms

} Loop for all coins

19. Appendix B : C120S versus C120P

C120S is the replacement model for C120P with superior coin discrimination. It uses an enhanced ccTalk command set. C120P operated with the very first issue of the ccTalk specification issued into the public domain and many enhancements have been made since then.

The following commands are taken from the C120P Serial Communication Protocol - Interface & Command Specification - Issue 3.2 document.

Table 10: C120S v C120P Commands

C120P Command	C120S Comments
254 : Simple poll	No difference
248 : Request status	No difference
243 : Request build version	Renamed 'Request database version'. Same reply.
242 : Request electronic serial number	Renamed 'Request serial number'. Same reply.
235 : Read last credit or error code	No difference but 16 coins and more error codes.
232 : Perform self-check	No difference
231 : Modify inhibit status	No difference but an additional 4 coins can be inhibited in the upper nibble of the 2 nd byte.
230 : Request inhibit status	No difference but support for 16 coins - see above.
228 : Modify master inhibit status	No difference
227 : Request master inhibit status	No difference
200 : Download coin data	Obsolete. There is a new method of remote coin programming using application specific headers.

There are some differences in the command responses, mainly to do with product identification, which software engineers need to be aware of.

Table 11: C120S v C120P Command Responses

C120P Command	C120S Comments
246 : Request manufacturer id	'Coin Controls Ltd' → 'Money Controls'
244 : Request product code	'C120P' → 'C120S'
241 : Request software revision	'P-0.17' → 'GTO-F1-V2.xx'
004 : Request comms revision	'4.2.0' → '1.4.2' (ccTalk spec. is now issue 4.2 rather than 2.0)

Due to the addition of flash reprogramming, there have been changes to some of the unused pins on C120P which were originally tied to 0V. Pins 2, 4 and 6 should be left floating.

This manual is intended only to assist the reader in the use of this product and therefore Money Controls shall not be liable for any loss or damage whatsoever arising from the use of any information or particulars in, or any incorrect use of the product. Money Controls reserve the right to change product specifications on any item without prior notice