C120P Serial Communication Protocol
-
Interface & Command Specification
-
Issue 3.4
-
Andy Barson

Revision History

Issue	Date	Comments
1.0	29-04-96	Draft specification
2.0	20-05-96	Updated after customer meeting on 15/5/96 Commands returning ASCII strings have been removed from this document Build number request added (returns 1 byte only) Protocol Summary added - this document may be read *stand-alone* and will form part of the product specification Section 3.1 modified - no coin acceptance if CMM holds READY line active. Credit information is not lost whilst power remains on processor.
3.0	27-06-96	Change to processor state between coin insertions. Processor will be active but in a low power state rather than have the power cut off. Text revised to reflect this change - particularly Section 3 which has been expanded.
3.1	03-10-96	Header defined for downloading coin data (calibration)
3.2	01-07-97	Request build version - defined as database version Some minor text revisions - document made generic Addition of section 8 - connector pinout
3.3	15-04-03	Assigned TSP number.
3.4	30-06-04	Changed footer

Contents Page

1 Introduction

Please refer to 'cctalk Serial Communication Protocol / Generic Specification' document for an overview of the serial communication protocol. This document deals with a specific implementation of the protocol for payphone projects using the C120P coin validator.

The message structure is defined as follows.

Standard Message Packet (header byte, N data bytes)
[Destination Address]
[No. of Data Bytes]
[Source Address]
[Header]
[Data 1]
...
[Data N]
[Checksum]

The structure does not alter according to the direction of the message packet. The serial protocol structure *does not care* who originates the message and who responds to it.

1.1 Protocol Summary

The timing of the serial data bits conforms to the RS232 industry standard for low data rate NRZ asynchronous communication. RS232 has various parameters and these are configured as follows :

4800 baud, 1 start bit, 8 data bits, no parity bit, 1 stop bit

Timing between bytes is not important and may vary, but if there is a gap of more than 50 ms **within** a message packet then a timeout condition will occur.

A level-shifted, low voltage version of RS232 is used for convenience and to reduce cost. On the serial connector, the idle state = +5V nominal and the active state = 0V nominal.

[No. of Data Bytes] indicates the number of **data bytes** in the message, not the total number of bytes in the message packet. A value of '0' means there are no data bytes in the message and the total length of the message packet will be 5 bytes - the minimum allowed. A value of 252 means that another 255 bytes are to follow, 252 of which are data. The values 253 to 255 are illegal and will be treated as if the value was 252.

[Checksum] is a simple zero checksum such that the 8-bit addition of all the bytes in the message from the header to the checksum itself is zero. If a message is received and the addition of all the bytes is non-zero then an error has occurred.

An acknowledge message (ACK) is a 5 byte packet - there are no data bytes and the header byte is 0. Refer to section 6, Message Protocol, for more details.

The use of a not-acknowledge message (NAK) is avoided to reduce the chance of collisions in multi-drop applications. Any error condition results in no response by the mech. The master device must take whatever action is appropriate (retry or fail).

2 Notation

The master device initiates a communication sequence and sends a command or a request for data. The master device will be referred to as the **CMM** (Coin Management Module).

The slave device responds to a command with an acknowledge or a request with 1 or more data bytes. The slave device will be referred to as the **mech** (Coin Mechanism or Coin Validator).

All data byte values below are shown in decimal unless stated otherwise.

Values between square brackets, [XXXXXXXX], indicate message bytes, ie. 8 bits of data, the base unit of transfer in the RS232 configuration used here.

X indicates a bit in a don't care or undefined state
0 indicates a bit set to 0
1 indicates a bit set to 1

Message structure within bytes is shown with a vertical separator bar. MSB on the left, LSB on the right.

3 Low Power Considerations - Handshaking

The C120P payphone validator is a low power application and requires the processor to be in a low power (non-functional) state between coin insertions. Serial communications cannot take place with the processor in a low power state.

The mech makes use of a bi-directional READY line, active low, for handshaking. The mech uses it to signal to the CMM that a serial credit is available for reading. In addition, the CMM may use it to pull the processor out of the low power state for diagnostic purposes etc. These functions are mutually exclusive and so care must be taken when certain operations are performed.

The microcontroller has an EEPROM to store vital configuration data and the contents of this are preserved when the power is switched off. Information such as coin inhibit status is retained. Contents of the microcontroller RAM, however, are lost.

3.1 Coin Insertion Algorithm

1) Phone goes off hook, power applied to mech
2) Processor initialises and enters a low power state (INITIALISATION_TIME)

a) Coin detected entering mech, processor switches to normal run state
b) Coin validation sequence executed
c) Accept gate opened, coin detected on credit sensor, accept gate closed
d) Mech activates READY line
e) Mech starts credit timer for CREDIT_TIME
f) CMM requests credit code & mech replies immediately
g) Mech waits for credit timer to expire
h) Mech releases READY line and returns to low power state

3.1.1 Credit Generation

The CMM only knows a new coin has been entered by monitoring the READY line and checking that it goes from an inactive state to an active one. This hardware handshaking line is used to distinguish one serial credit from another and to prevent ambiguity when performing multiple serial reads.

When the mech receives a credit code request, it re-starts the credit timer. This gives time for the CMM to re-send the command in the event of an error. It is up to the CMM whether a checksum integrity test and immediate request retry is performed.

The mech will not accept any more coins until the credit timer has expired and the READY line goes inactive.

3.1.2 Reject Coin

The same sequence is gone through as outlined above but the accept gate does not open (step 'c'). The credit code is replaced by an error code, *reject coin*. See Section 7.5 for more information.

3.1.3 Mech Diagnostics

If the CMM wants to communicate with the mech for any reason other than reading a credit code (fetching mech serial number, executing self-check etc.) it must activate the READY line. This forces the processor to switch from low power state to normal run state. There will then be a time STATE_RECOVERY_TIME before a serial command can be sent.

With the READY line active, the mech cannot enter a low power state and will always respond to serial messages. Note that the last credit code to be generated since the mech powered up will still be available for reading.

If the CMM holds the READY line active then the mech will not accept any coins.

Note that because the READY line is bi-directional the CMM must not activate this line when coins are passing through the mech or there will be the possibility of missed credits. Any serial commands that need to be executed after the phone goes off hook should be performed immediately before coins can physically reach the mech. The mech will read the READY line just prior to the accept gate opening - this is the *cut-off point*. If the READY line is active, the coin will not be accepted.

To read the serial number after the phone goes off-hook, the CMM must do the following.

a) Apply power to mech (phone goes off-hook)
b) Wait for INITIALISATION_TIME
c) Activate READY line
d) Wait for STATE_RECOVERY_TIME
e) Perform serial commands
f) Release READY line

3.1.4 Inhibit All

The INHIBIT ALL line puts the processor in a low power state and prevents further coins from accepting.

The INHIBIT ALL line can be used up to the *cut-off point* (see above) in the validation cycle. After that, a pending coin will be accepted and credited before further coins are inhibited.

3.2 Nominal Timer Values

Description	Label	Value
Coin insertion rate	-	< 2 coins / s
Time between coins	-	> 500 ms
Typical serial message exchange	-	30 ms
CMM max. response time	CREDIT_TIME	50 ms
Time to switch from low power mode to normal run mode	STATE_RECOVERY_TIME	< 100us
Mech initialisation time after power-up	INITIALISATION_TIME	< 20 ms

4 Example Message

To get an idea of the level of complexity of the serial communication protocol, here is an example showing all the bytes transferred between the CMM and mech in order to determine the type of coin just accepted.

CMM : [2] [0] [1] [235] [18]
Mech : [1] [1] [2] [0] [1] [251] (eg. £1 credit)

At 4800 baud and 10 bits per RS232 byte, each byte takes 2.083 ms.

The mech can send transmit a byte every 3 ms.

Assuming the CMM fires the bytes out as fast as possible, the total communication time will be about **30 ms**.

Total transfer of 11 bytes.

The only real redundancy is 4 bytes of source and destination address information which is retained for compatibility with future multi-drop applications.

5 Message Implementation

11 core messages are implemented in the C120P coin validator. Contact CCL for details of other cctalk commands which may be available.

- Simple poll
- Request status
- Request build version
- Request electronic serial number
- Read last credit or error code
- Perform self-check
- Modify inhibit status
- Request inhibit status
- Modify master inhibit status
- Request master inhibit status
- Download coin data

6 Message Protocol

Since the C120P application is a simple 2-point connection network, the following conditions will always apply to the CMM when it issues a command or request.

[Destination Address] = 2
[Source Address] = 1

When the mech returns data, the following conditions will apply.

[Destination Address] = 1
[Source Address] = 2

The acknowledge message from the mech would therefore be

[Destination Address] = 1
[No. of Data Bytes] = 0
[Source Address] = 2
[Header] = 0
[Checksum] = 253

In hex, the following bytes would be received by the CMM :
01 00 02 00 FD

7 Command Set

7.1 Header 254 - Simple poll

An acknowledge message is returned.

No action is taken by the mech when this command is received. It just returns an acknowledge to verify the communication link is working correctly. If there is no response then check the following...

a) Is the mech fitted ? Is it a CCL C120P mech ?
b) Is there power on the mech ?
c) Is the READY line active (processor powered up) ?
d) Is the cable harness OK ?
e) Have the destination & source addresses been set correctly ?
f) Is the mech faulty ? Try another unit.

7.2 Header 248 - Request status

The return data is 1 byte.

[STATUS_CODE]

STATUS_CODE is defined in the following table :

Value	Meaning
0	OK (operating)
1	Flight deck open (return lever pressed)
2 to 255	Future expansion

7.3 Header 243 - Request build version

The return data is 1 byte.

0 indicates a prototype model.

1 to 255 refers to a specific production build.

Production build forms part of the coin data downloading process (calibration) and is identical in this product to the 'coin database version'.

7.4 Header 242 - Request electronic serial number in binary

The return data is 3 bytes.

[SERIAL_LOWER]
[SERIAL_MIDDLE]
[SERIAL_UPPER]

The electronic serial number is unique to each mech and cannot be modified for security reasons. The 3 bytes may be combined into a 24-bit binary number to produce a serial number in the range 0 to 16,777,215.

7.5 Header 235 - Read last credit or error code

The return data is 1 or 2 bytes depending on whether the last coin credited correctly.

[WINDOW]
or
[0]
[ERROR CODE]

WINDOW is 1 to 12 and indicates the type of coin accepted. If it is '0' then an additional byte is returned which is the error code.

ERROR is defined in the following table :

Value	Meaning
0	Null event (no events logged)
1	Reject coin
2	Inhibited coin
3	Multiple window error (coin type is ambiguous)
4 to 255	Contact CCL for details

The ERROR byte will also be able to indicate possible coin jam and fraud conditions that were evident after the last coin was entered.

In the C120P implementation, repeated use of this request will return the same information. The READY line (part of the hardware handshaking) must be monitored to determine when another coin has been entered and another credit is available for reading.

7.6 Header 232 - Perform self-check

The mech immediately executes a self-test sequence.

The return data is 1 byte.

[FAULT_CODE]

FAULT_CODE is defined in the following table :

Value	Meaning
0	OK (no fault detected)
1	EEPROM checksum corrupted
2	Fault on inductive coils
3	Fault on credit sensor
4 to 255	Contact CCL for details

7.7 Header 231 - Modify inhibit status

2 bytes of data are sent to the mech.

[INHIBIT_1_8]
[XXXX | INHIBIT_9_12]

0 = coin inhibited
1 = coin enabled

The 1st byte consists of inhibit status bits for coin windows 1 to 8. The LSB is window 1, the MSB is window 8.
The 2nd byte consists of inhibit status bits for coin windows 9 to 12. The LSB is window 9. The upper nibble of this byte is not used.

The inhibit status is modified accordingly and stored in EEPROM.

An acknowledge message is returned.

7.8 Header 230 - Request inhibit status

The return data is 2 bytes.

[INHIBIT_1_8]
[0000 | INHIBIT_9_12]

0 = coin inhibited
1 = coin enabled

The 1st byte consists of inhibit status bits for coin windows 1 to 8. The LSB is window 1, the MSB is window 8.
The 2nd byte consists of inhibit status bits for coin windows 9 to 12. The LSB is window 9. The upper nibble of this byte is not used.

7.9 Header 228 - Modify master inhibit status

1 byte of data is sent to the mech.

[XXXXXXX | MASTER_INHIBIT]

0 = all coins disabled
1 = normal operation

The master inhibit status is modified accordingly and stored in EEPROM.

An acknowledge message is returned.

Both the MASTER_INHIBIT flag in EEPROM and the external INHIBIT ALL line must be in the right state for a coin to accept.

7.10 Header 227 - Request master inhibit status

The return data is 1 byte.

[0000000 | MASTER_INHIBIT]

0 = all coins disabled
1 = normal operation

The master inhibit status stored in EEPROM is returned, not the status of the external INHIBIT ALL line.

Both the MASTER_INHIBIT flag in EEPROM and the external INHIBIT ALL line must be in the right state for a coin to accept.

7.11 Header 200 - Download coin data

A block of calibration data is sent to the mech.

[data 1] [data 2] [data 3] ...

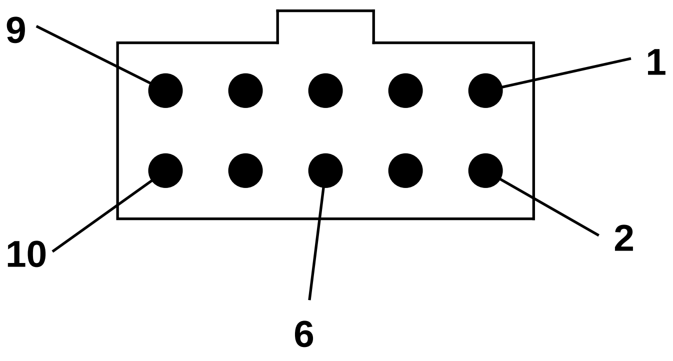
An acknowledge message is returned.

This command allows remote re-programming of coin acceptors by transferring up to 252 bytes of calibration data. Further details are given in a separate document on calibration.

8 Connector Pinout

The C120P serial connector is a 10-way, dual pin header on a 0.1inch spacing with a mechanical key.

The pinout is as follows :



Connector-side View of C120P

Pin	I/O	Label	Polarity	Description
1	I/O	/DATA	Active low	Bi-directional serial data line
2	-	GNDSHLD	-	Ground shield (may be left unconnected)
3	I/O	/READY	Active low	Bi-directional handshaking line
4	-	GNDSHLD	-	Ground shield (may be left unconnected)
5	I	/RESET	Active low	Reset line
6	-	-	-	N/C
7	I	/INHIBIT	Active low	Master inhibit line
8	P	GND	-	Common 0V line
9	P	VBOARD	-	+5V power line
10	P	GNDP	-	Common 0V line for accept gate solenoid (on some models only)