ESKDALE ROAD, WINNERSH TRIANGLE, WOKINGHAM, BERKSHIRE RG41 5AQ
Tel: 0118 969 7700 Fax: 0118 944 6412

----- *PROTOCOL "A"* -----

TITLE	:	MEI 20 Ma Protocol A Specification
NUMBER	:	10102-000304001-PS
ISSUE	:	Y2
DATE	:	23 Aug 2001
FILE	:	000304001 - Protocol A - Y2.doc
AUTHOR	:	J Weston
COPY	:	Master

Change History

Issue	Date	Description
Y01	10/02/95	Created based on the references listed in section 1.5
Y2	23-Aug-2001	Converted from Framemaker to MS Word

Authors :	....		
	....		
Technical Manager :	....	Date	....
	....		....

Distribution List

Copy	**Recipient**

1 Introduction

1.1 Purpose

The purpose of this document is to specify the serial protocol based on the MEI 20 mA current loop link used to connect MEI and other third party devices in a transaction system. This document collates a number of previously published MEI specifications into a common format. A number of design guidelines have also been added to the basic specification which should aid development of a Protocol A based system.

1.2 Scope

The scope of this document is restricted to the specification of the current 20 mA serial protocol and the design guidelines mentioned above. Where required, examples or recommendations on the operation of the serial link will be given.

This document has been compiled from a number of MEI specifications which described various aspects of the protocol (see section 1.5). This document supersedes any references marked as such.

1.3 Overview

Chapter 2 gives an overview of the protocol and the various extensions which have been added.

Chapter 3 specifies the physical layer and the datalink layer of the protocol.

Chapter 4 describes in detail the functionality and data formats (i.e. the application layer) for the Executive to VMC.

Chapter 5 describes in detail the functionality and data formats for the Executive to Audit Storage Unit.

Chapter 6 describes in detail the functionality and data formats for the Executive to Cashless Payment Peripheral.

1.4 Terms & Abbreviations

ACK	-	The mnemonic for the serial datalink character Acknowledge (decimal value = 0). This reply is also used by any peripheral to a command that it does not understand but has not suffered a parity, framing or overrun error.
Application Layer	-	The "highest" level in the OSI protocol model. Application Layer messages are in the format, and contain the data applicable to the actual application software in each node.
Asynchronous	-	A form of serial communication which does not use a separate clock signal. The data signals supply the required synchronisation for the system. The proper term is Start/Stop Asynchronous, as the link nodes are synchronised between the Start and Stop bits.
ASU	-	Audit Storage Unit. A peripheral node on the network which stores detailed information of the transactions which have occurred within the system.
Baud	-	The unit of measure of the fundamental electrical signalling rate of a communications line. Although frequently used, the term "baud rate" is incorrect in the same way that it is incorrect to say "hertz rate". Note that baud is not a unit of measure of information transfer speed, and the baud of the link implies nothing of its information transfer rate. The Protocol A link operates at 9600 baud.
bps	-	Bits per second. This is the unit of measure of the rate of transfer of information of the link. In the simple case where one data bit is encoded during one signalling period (as is the case on the Protocol A link), then the signalling rate of the link (baud) is equal to the information transfer rate (bps).
BDV	-	Bundesverband der Dienstleistungsunternehmen fuer Verplfegungssysteme E.V. (German Vending Committee).
BDV 001 Protocol	-	A set of extensions made to Protocol A to allow a more flexible protocol for vending machines equipped with BDV devices.
Character	-	The smallest packet of information that can be transmitted on the link. It comprises of 8 data bits, and 1 parity bit. Each character is framed by 1 start bit and 1 stop bit.
CPP	-	Cashless Payment Peripheral. A peripheral node on the network which handles cashless payment media (e.g. magnetic cards).
Datafile	-	The term given to a "block" for data which is transferred between nodes on the network.
Datalink Layer	-	The purpose of the datalink layer is to provide secure, error-free data transfer across the physical link (layer).
Executive	-	The network node which sources the datalink current and performs the role of system controller. Other terms used for this function include "Link Master", and "Master".
Framing Error	-	Name given to an invalid stop bit error when receiving asynchronous serial data.

Hexadecimal	-	Base 16 numbering notation. The basic digits consist of 0-9 and A - F, representing the decimal values 0 to 15. Hexadecimal (or hex) numbers are indicated by preceding the number by 0x or following it by h. E.g. 18 decimal is written in hex as :12h or 0x12
Link Master	-	Another term for Executive.
Link Peripheral	-	A network node which performs a specific function in the vending environment under the direction of the Executive. Link Peripheral nodes are sometimes called "Slave" nodes.
Logical Node	-	A node or point on the network which has a unique address and command set. Within this document, unless otherwise stated, this will be shorten to just "node". There is often confusion between "logical" nodes and "physical" nodes. Logical nodes are functional entities defined by an address and command set. Physical nodes are items of hardware connected physically to the network. Each physical node on a functional network will contain at least 1 logical node. In many cases, it may contain more than 1 logical node.
Mark	-	A logic 1, indicated by the absence of current flow, on the serial link.
Master	-	Another term for Executive.
NAK	-	The mnemonic for the serial datalink character Negative Acknowledge. It is normally command code 15, sent by the Executive. This command is sent to a peripheral whose previous character was received by the Executive with a parity, framing or overrun error.
Network Layer	-	The purpose of this layer is to establish a communication channel which will allow the transfer of packets of data between nodes.
NRZ	-	Non-return to zero. A method of signalling data over a physical link, which only requires 2 signalling states.
OSI Model	-	A 7 layer model for good protocol design developed by the CCITT OSI (Open Systems Interconnect) committee. The layers are named : Physical, Datalink, Network, Transport, Session, Presentation, Application.
Overrun Error	-	Name given to a error, caused by receiving a character before the previous character was read and processed, when receiving asynchronous serial data.
Parity Bit	-	A single bit added to the "end" of an 8 bit character to enforce a system design rule that all characters must have an even (or odd) number of logic 1 bits. This allows the detection of single bit errors in the data character. On Protocol A, the rule is to use even parity, hence the parity bit is set or cleared to ensure each character always has an even number of 1's (not including the start or stop bits).
Peripheral	-	Another term for a Link Peripheral.
Physical Layer	-	The physical connection between network Physical Nodes. The purpose of this layer is to transfer electrical signals between nodes on the network. The layer defines of all aspects of the physical link (mechanical & electrical).

Physical Node	-	An item of hardware connected to the network's Physical Layer. Each physical node on a functional network will contain at least 1 Logical Node. In many cases, it may contain more than 1 logical node.
PNAK	-	The mnemonic for the serial datalink character Peripheral Negative Acknowledge (decimal value = 255). All peripherals must use this reply to any character addressed to them which has suffered a parity, framing or overrun error.
Polling	-	When the Executive sends STATUS commands to each peripheral node in sequence.
Protocol A	-	The MEI term for the original 20 mA serial protocol between a Coin Changer and a VMC.
Protocol A+	-	An extension made to Protocol A to allow enhanced operation with the MEI Multicard system.
Slave	-	Another term for a Link Peripheral.
Space	-	A logic 0, indicated by current flow, on the serial link.
Start Bit	-	A single logic 0 bit (space) which precedes the data character to be sent. This bit (or to be precise, the logic 1 to logic 0 transition) acts as a synchronising signal to each node, on a character by character basis.
Stop Bit(s)	-	One (or more) logic 1 (mark) bits which follow the data character (and parity if used). Note that the stop bit is the same as the link's idle state (mark) and is maintained until the next start bit.
Topology	-	The term used to describe the layout or structure of a network.
UART	-	Universal Asynchronous Receiver / Transmitter.
VMC	-	Vending Machine Controller - the network node which controls the vending machine.

1.5 References

BDV 001 Specification	-	S051AC5
Protocol A Specification	-	S063BC1 †
Executive to DCR Communications	-	PADCR.DOC V1.0 26/4/85 †
Multicard Peripheral Controller Communications	-	S063KC1 †
Recommendation X.200	-	CCITT Fascicle VIII.5 ‡
Data Communications, Computer Systems & Open Systems	-	ISBN 0-201-56506-4 ‡

† These MEI specifications are superseded by this document.

‡ These are available from any good technical bookshop (e.g. ILI in the UK on 01344 23377)

2 Protocol Overview

This chapter gives a brief overview of the protocol. The following are some key points on the principles behind the chosen design of the protocol :

- The protocol is based on the concept of a Executive node controlling a number of peripheral nodes.
- Overall operation of the system is controlled by the Executive node.
- All system level error handling is done by the Executive node.
- The design of the protocol allows each peripheral to have a very simple command driven structure.
- Peripherals do not need to track, or react to, the overall system state.
- Link timing is kept as relaxed as possible to minimise cost and complexity in nodes.
- The above leads to a system where extra nodes can be added easily, with changes limited to the Executive node only. Other peripheral nodes will not be affected.

2.1 History

A brief history of the development of the protocol follows to allow the reader to understand the present protocol and its extensions. Figure 2.1 below shows this development history. The thick line represents the core protocol present in MEI products using the serial link.

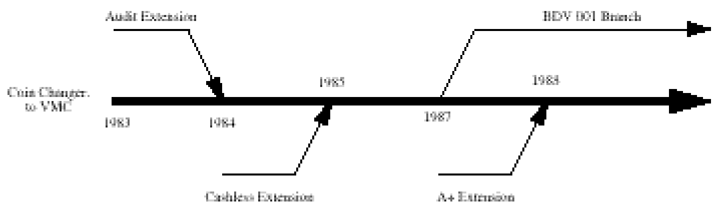


Fig 1 : Protocol Development History

As can be seen above, the initial purpose of the serial link and the protocol was to connect a Coin Changegiver to a Vending Machine Controller. This link / protocol was given the name "Protocol A", and the title is still used as a generic term.

Three main extensions have been added to the initial protocol to enhance its functionality, these are :

- An extension to allow the connection of an audit system,
- An extension to allow the addition of a cashless payment system.
- An extension to allow a number of extra features on second generation cashless systems (the A+ extension).

In addition to these extensions, a branch of the protocol was created by request of the BDV committee. The functionality of this branch is defined in the BDV 001 standard, and is outside of the scope of this document.

The following sections will give further details on the basic protocol and the extensions just described. It should be noted that these are based on a common hardware link, which is described first followed by the details of character transmission on the link.

2.2 Protocol Structure

The OSI model for protocol design defines the following layers :

- Application
- Presentation
- Session
- Transport
- Network
- Datalink
- Physical

Due to the simple nature of Protocol A, most of these layers are not used. The protocol structure can thus be simplified down to :

- Application
- Datalink
- Physical

The rest of this section will briefly describe each of the above 3 layers. Full details of the Physical and Datalink layers are in chapter 3. Chapters 4 to 6 define the Application layer for the various peripheral devices.

2.2.1 Physical Layer

The physical layer of the link is a full duplex, current loop connection. The principal of the current loop is that the logical state of the data bits transmitted are represented by either the flow, or lack of, current in the loop. The current loop provides a high level of noise protection for the link and also allows (by the use of opto-isolators) each node to be electrically isolated.

The current source for both loops of the link is provided by the Executive. If the Executive wishes to transmit a series of data bits, it will do so by sourcing a series of current pulses into the TX loop of the link, all peripheral nodes will see these current pulses and thus be able to re-create the data sent by the Executive.

If a peripheral node wishes to send data to the Executive, it will "short" the RX loop of the link, thus effecting a flow of current. By monitoring these current pulses, the Executive can determine the data sent by the peripheral. Figure 2.2 shows a schematic of the electrical connection between Executive and peripheral nodes.

Appendix A gives the recommended circuits for the Executive and peripheral nodes. If the recommended circuits are used, then up to 4 peripherals may be used on the network.

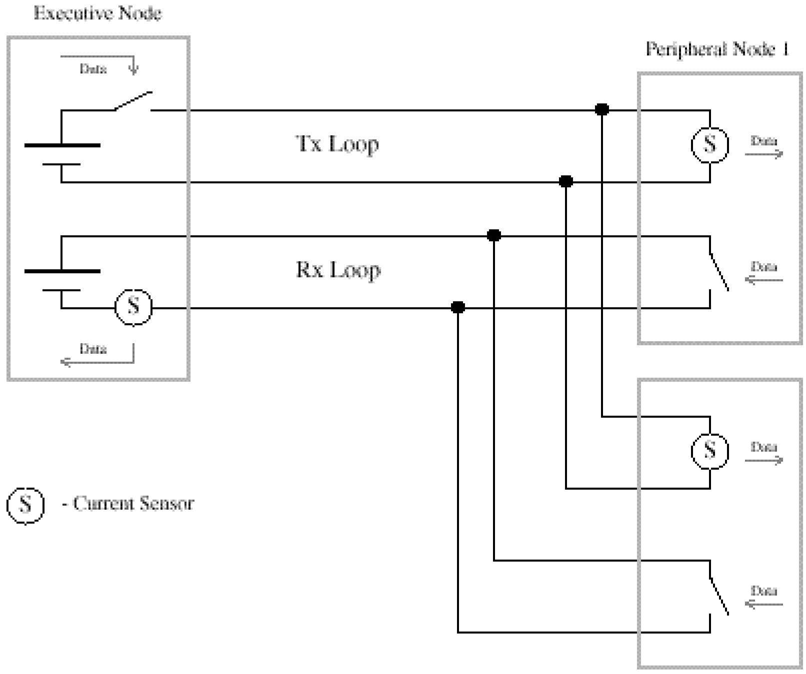


Fig 2.2 : *Current Loop Schematic*

Fig 2 : Current Loop Schematic

2.2.2 Datalink Layer

The protocol uses simple 8 bit character based data transfer. Characters are sent over the link at 9600 baud. A parity bit is used to allow the integrity of the data character to be checked.

Error detection and handling is very simple. The Executive will send out a single character message to a node. The addressed node must then respond within in a specified time with a single character reply. Based on the response (or lack of) the Executive can decide if the sequence was successful or not, and take the appropriate action.

2.2.3 Application Layer

Each (single character) message sent out by the Executive consists of a 3 bit address, a 1 bit control flag and a 4 bit field which contains either a command code or data nibble. Thus the Executive to peripheral application layer consists of 2 basic message types :

- Command
- Data

For any given node, up to 16 commands can be defined. Data messages can only transfer a nibble of data at a time to an addressed node.

The response from the peripheral to the Executive is an 8 bit binary reply code. Certain reply values are reserved for use as link control codes. These are fully defined in the following chapters.

2.3 The Protocol & Extensions

2.3.1 Coin Mechanism to VMC Protocol

The original protocol was designed to allow the connection of a Coin Mechanism to a VMC. The Coin Mechanism acted as the system Executive, with the VMC being a peripheral node. The protocol is structured to allow the Executive control over the peripheral. The amount of data sent across the link is minimal and the protocol could be though of as mainly a "control" protocol.

The Executive performs a poll of the peripheral node, which replies with its status. This status reply is analysed by the Executive to determine what action the peripheral should be directed to take.

As mentioned in the above section, each character sent from the Executive contains the address of the destination node, and the Executive expects a reply to each of these bytes sent. This provides a simple command structure, with most operations requiring only two bytes of data to be sent across the link. As said above, the protocol is structured to be efficient for node control, but is less efficient when data transfer between nodes is required.

2.3.2 Audit Extension

The audit extension to the protocol allowed the connection of a Audit Storage Unit node to the network. The structure of the extension was very similar to that of the core protocol A.

If any node wishes to send data to the Audit Storage Unit, it indicates this fact in its status reply to the Executive's poll. The Executive will request the audit data from the node, and send it on to the audit unit. The data transfer method is inefficient and restrictive, requiring 14 link transfers to get one byte of audit data from the originating node to the audit unit.

2.3.3 Cashless Extension

The cashless extension to the protocol allowed the connection of a cashless payment unit to the network. As for the VMC, the link is mainly control based, with limited data transfer ability.

This extension has a very limited command set, allowing only the reading, decrementing and returning of a pre-programmed card, under the control of the Executive.

2.3.4 A+ Extension

Due to the limited functionality of the above cashless extension, a further extension was added to handle the 2nd generation cashless systems. This extension added the ability to perform card revalidation, subsidised vending and display of credit via the cashless system.

2.3.5 BDV 001 Branch

Prior to the addition of the A+ extension, the BDV committee requested that MEI developed a version of protocol A to support a lower cost vending system. The main feature required was a means of transferring larger data files around the network. These data files would hold configuration and audit data for designated nodes. The BDV 001 branch (see reference S051AC5) was developed to perform these tasks. In addition it extended the range of data which could be audited across the serial link.

A BDV 001 Executive will determine the status of each node in the same way as protocol A. If a node supports the BDV 001 extensions, it will signal this in its status reply. The Executive can then gain a picture of the type of nodes connected (BDV or non-BDV), and use the relevant commands for each node.

The BDV 001 protocol is more complex than the basic Protocol A and is not under MEI control. It is outside the scope of this document.

3 Physical & Datalink Layers

3.1 Physical Layer

The physical layer of the link is based on two 20mA current loops. The "transmit" (TX) loop transfers data from the Executive to the peripherals. The "receive" (RX) loop transfers data from the peripherals to the Executive. An earthed screen is also provided to improve noise immunity. Thus the link consists of 5 wires with the following signal names :

- Executive transmit +ve (Tx+)
- Executive transmit return (Tx-)
- Executive receive +ve (Rx+)
- Executive receive return (Rx-)
- Screen

Both current loops are housed in a single screened twisted pair cable (e.g. Belden 8723). It is important that the screen continuity is maintained throughout the length of the link, and that it is only connected to earth or 0V at the Executive end. Thus the screen must *not* be connected to any part of the circuitry of the peripheral.

3.1.1 Connector Specification

The Protocol A harness from the Executive must terminate in a 9 way Molex plug with the following pinout (based on Molex pin numbering). Recommended wire colours are shown in brackets :

1	Tx+	(Red)
2	Rx-	(Green)
3	Rx+	(White)
4	Tx-	(Black)
9	Screen	

The Molex series reference numbers for the connectors are as follows :

Plug on Executive harness - 1625-9P (part # 03-06-2091 or 03-06-2092)
Receptacle on peripheral harness - 1625-9R (part # 03-06-1091 or 03-06-1092)

Male crimp for plug - 1854 (part # 02-06-2131 or similar)
Female crimp for receptacle - 1855 (part # 02-06-1131 or similar)

3.1.2 Connection of Multiple Nodes

The harness from the Executive node will have a single plug termination. It is recommended that each peripheral is supplied with both a plug and receptacle harness to allow easy "daisy chaining" of nodes on the link (see figure 3.1). Alternatively, a "T" piece loom may be used.

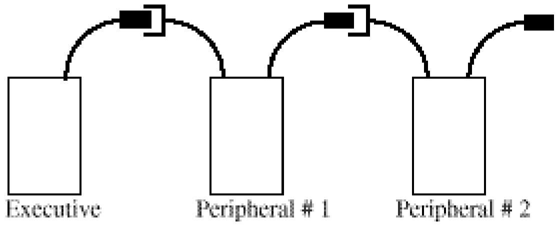


Fig 3 : Daisy chaining of nodes

Note that in the above, all 5 signal lines are "fed through" each peripheral. The peripheral will "tap off" from each current loop as shown in appendix A. When the recommended interface circuits are used, up to 4 peripherals may be fitted.

3.1.3 Current Loop Specification (Executive)

The Executive node must be able to drive the Tx and Rx current loops to the following minimum specification :

Transmit (Tx) Loop :

Minimum source current (active)	100 mA	(see notes 1, 2, 3)
Maximum leakage current (inactive)	100 uA	(see note 4)

Receive (Rx) Loop :

Minimum active current	15.0 mA	(see notes 5, 6, 8)
Maximum inactive current	1.0 mA	(see notes 7, 8)

Notes :

1) If the Executive is signalling a "space" (logic 0) on the link, then it MUST be able to source at least 100mA into the Tx loop.

2) The voltage between the Tx+ and Tx- lines on the Executive's harness MUST be at least 4.0 V when the current flowing through the Tx loop is 100 mA or less.

3) The Executive must be able to withstand a continual short across the Tx+ and Tx- pins on its harness, without damage.

4) If the Executive is signalling a "mark" (logic 1) on the link, the maximum permitted current flow in the Tx loop is 100uA, irrespective of the loading across the Tx+ and Tx- pins on the harness.

5) The Executive must register the link state as "space" (logic 0) when the current flow in the Rx loop is 15.0 mA or more.

6) The Executive must be able to withstand a continual short across the Rx+ and Rx- pins on its harness, without damage.

7) The Executive must register the link state as "mark" (logic 1) when the current flow in the Rx loop is 1.0 mA or less.

8) If the current flow in the Rx loop is between greater than 1.0 mA but less than 15.0 mA, then the link state registered by the Executive cannot be guaranteed.

3.1.4 Current Loop Specification (Peripheral)

Each peripheral node must be able to drive the Tx and Rx current loops to the following minimum specification. Note that the transmit (Tx) loop refers to the transmission from the Executive.

Transmit (Tx) Loop :

Maximum active current	20 mA	(see notes 1, 2)
Maximum inactive current	100 uA	(see note 3)

Receive (Rx) Loop :

Minimum sink current	15.0 mA	(see note 4)
Maximum leakage current	30.0 uA	(see note 5)

Notes :

1) With a voltage of 5.0 V across the Tx+ and Tx- pins to the peripheral, the current in the Tx loop should not exceed 20.0 mA.

2) With a voltage of 4.0 V (or greater) across the Tx+ and Tx- pins to the peripheral, the peripheral MUST register the link state as "space" (logic 0). Under this condition, it is recommended that the current flow in the Tx loop should be at least 10mA to ensure good noise immunity on the link.

3) With a current flow of 100uA or less in the Tx loop, the peripheral MUST register the link state as "mark" (logic 1).

4) If the peripheral is signalling a "space" (logic 0), then it MUST sink at least 15mA in the Rx loop, without the voltage across the Rx+ and Rx- pins exceeding 1.0 V.

5) If the peripheral is signalling a "mark" (logic 1), the current in the Rx loop MUST not exceed 30uA.

3.1.5 Network Topology

The network topology of the Protocol A link is somewhat of a hybrid. The transmit (Tx) current loop is effectively a "bus" topology, since when the Executive transmits, all nodes receive the data. However the receive (Rx) loop is more of a "star" topology, as data sent by a peripheral can only be received by the Executive. This dual current loop configuration places a number of restrictions on the operation of the serial link. The main ones are :

- There can be only one Executive (current sourcing) node. This node is specified at the system design stage, and cannot be dynamically changed during operation.
- The link does not allow direct peripheral to peripheral data transfer.
- There is no way to resolve contention on the Rx loop. This implies a strict message protocol must be observed.

3.2 Datalink Layer

3.2.1 Character Transmission Format

Character transmission over the link is done asynchronously using NRZ signalling at 9600 baud (+/- 3%). The following characteristics apply to each character transfer :

- Each data character consists of a start bit, 8 bits of data, a parity bit and a stop bit.
- The data bits are sent least significant bit first.
- The parity bit is set / cleared to give even parity.
- The start bit is logic 0.
- The stop bit and idle state of the link is logic 1.
- Logic 0 is represented by current flow in the loop (space).
- logic 1 is represented by no current flow in the loop (mark).

Figure 3.2 shows the character format :

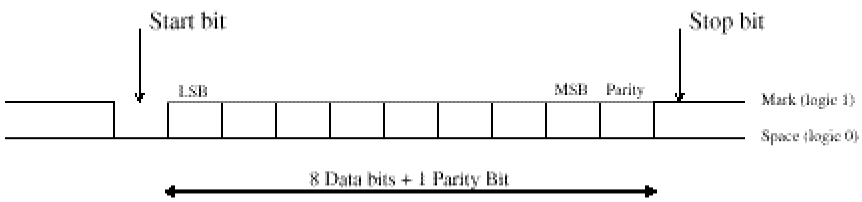


Fig 4 : Character Format

3.2.2 Executive to Peripheral Transfers

Section 2.2.3 briefly described the format of characters transmitted from the Executive to peripherals. The format is shown in figure 3.3 below :

Bit #

7 (MSB)	6	5	4	3	2	1	0 (LSB)
PERIPHERAL IDENTIFIER			MODE	COMMAND CODE OR DATA NIBBLE			

Fig 5 : Executive to Peripheral Character Format

The Peripheral Identifier (bits 5 to 7) signifies which peripheral the Executive is addressing. The currently defined Peripheral Identifiers are listed below (bit 7, bit 6, bit 5) :

- Vending Machine Controller 001
- Audit Unit 010
- Cashless Payment Unit 011

The Mode bit (bit 4) signifies whether the Executive is sending a command code or a data nibble to the peripheral. The bit is set ("1") for commands and cleared ("0") for data.

The lower 4 bits (bits 0 to 3) should be interpreted by the peripheral as a command if the mode bit is set. If the mode bit is clear, the lower 4 bits should be treated as data.

This specification and the protocol intentionally imposes no requirements on the timing of characters transmitted by the Executive. The response of the entire system is obviously affected by the frequency of transmissions from the Executive, but it is left to the designer of the Executive to judge what is acceptable for the application in question. This approach simplifies the design of a peripheral, as it need only respond when addressed, and does not have to be concerned with any other link timing. peripherals **MUST NOT** assume any message timings or sequences from the Executive.

For each character transmitted by the Executive to a fitted peripheral, a single character reply is expected. The following sections will detail the allowed replies.

3.2.3 Peripheral to Executive Transfers

The most important point to note is that a peripheral must only ever send a character in response to a correctly addressed message from the Executive. Apart from the case detailed in section 3.2.4, peripherals **MUST NEVER** generate unsolicited link traffic.

The second point to note is that unless otherwise specified, all peripherals should respond to a message from the Executive with minimal delay. Early specifications quoted a maximum delay of 10ms, but to ensure that coin acceptance is not affected, a maximum response time of 3ms is more appropriate (see figure 3.4). This is the only timing constraint specified in the protocol, and is required to ensure the system performance is not impacted by different peripheral designs.

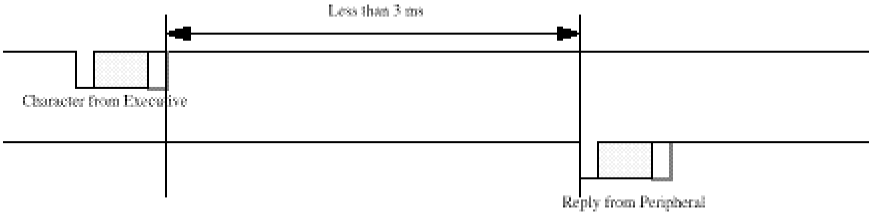


Fig 6 : Peripheral Response Timing

The reply character from the peripheral is an 8 bit binary number (range 0 to 255 decimal). The reply value of 0x00 (0 decimal) is generally used as an Acknowledge (or "OK") reply. The values 251 to 255 are reserved for special purposes as follows (see chapters 4 to 6 for full details) :

251	Failure of command (CPP)
252	Reserved
253	Reserved
254	Lost Sync, No vend request
255	Peripheral Negative Acknowledge (PNAK)

3.2.4 Peripheral Hailing on Power-up

To ensure the link is started as quickly as possible after power-up, each peripheral may issue a **SINGLE** unsolicited Peripheral Negative Acknowledge (PNAK) message after it powers up. This can be used by Executives with fairly long time-outs to detect a status change on the link, and take the appropriate action. This is the **ONLY** time a peripheral should issue an unsolicited reply, and is the only allowed alternate use of the PNAK (its main use is to signal corrupt data reception - see the following section).

3.2.5 Error Detection & Handling

Error detection at the Datalink Layer is done using the parity bit which is part of each character transferred on the link. The parity bit should be such that there are an even number of logic 1 bits in the 9 bit character (8 data bits and 1 parity bit). This method can detect (but not correct) a single bit error.

Peripheral :

- The error handling in a peripheral is probably one of the most misunderstood areas of the protocol. The reason for this is two-fold, firstly the initial specification was not clear on this point, and secondly the link performance was quite influenced by the method of error handling adopted.
- In an ideal system, the Executive should be able to quickly detect when a peripheral has received a corrupt character, and take the required corrective action. There are a number of possible ways the Executive can be "informed" of the reception of corrupt characters by the peripheral :
    - If on receiving a corrupt character the peripheral does not respond at all, then the Executive must be able to detect this lack of response fairly quickly and take corrective action. This approach is the most elegant for a network design such as Protocol A where all peripherals receive data from the Executive, but there is no way of resolving Rx loop contention. The downside to this approach is that each peripheral must meet fairly strict response times in normal (error free) operation, and the Executive has to have a fairly complex and accurate time-out detection system. This is because each character corruption leads to the link "stopping" for the time-out period. Hence if fairly relaxed time-outs are used with such an approach the link's performance can be significantly affected. When the system was initially developed the extra resources required to implement strict time-outs on each node were deemed excessive, and the alternate approach (below) was used.
    - On receiving a corrupt character the peripheral firstly checks if the character was addressed to it (see section 3.2.2), and if so responds with a reply indicating that the character was incorrect. This approach is not as "foolproof" as the above, but allows much less stringent time-outs to be used. Using this approach will mean that in at least 67% of single bit corruptions the peripheral will respond to the Executive indicating a corrupt character was received. This percentage will rise as the number of peripherals on the link increases. Hence the link will only "stop" if the corruption occurs in the peripheral Identifier section of the character, and the resulting address does not match any fitted peripheral. This is the approach which is recommended.

- Note that the above assumes that the corrupt character is received by all peripherals on the link.

-
-
- The reply code 0xFF (255 decimal) is reserved for use as a Peripheral Negative Acknowledge (PNAK) code. This reply will be sent by a peripheral when it receives a corrupt character which is "addressed" to it (see above). Except for the case described in section 3.2.4, the PNAK reply code MUST NOT be used for any other purpose.
- If the Executive receives a PNAK reply, it will usually attempt to re-transmit the message which it last sent. Note that this is an not a Datalink Layer decision, and the exact response of the Executive will depend on a number of factors. The peripheral **MUST NOT** make any assumptions on the Executive's response to the PNAK.
- There are currently a number of unused command codes for some peripherals. To ensure future compatibility, if a peripheral receives such a code, it **MUST** reply with an ACK (0 decimal).

Executive :

- There is no error handling done at the Datalink Layer by the Executive, apart from signalling to the Application Layer that a Datalink Layer error (parity, framing, overrun) has occurred.
- The Application Layer will make the decision on what the response of the Executive should be, based on the current state of the system. This application response is **NOT** part of the Protocol A specification, and peripherals need not, and should not assume a specific response. The following chapters will give examples of typical responses.

3.2.6 Datalink Layer Summary

- The Executive controls all transmissions on the link by issuing commands or data to specific peripheral nodes. It is mandatory that all peripherals check that the received character was addressed to it before offering any response (i.e. only one peripheral may be addressed at any one time).

- A peripheral is forbidden to originate anything onto the link except in reply to a message addressed to it. The only allowed violation of this rule is on power-up where each peripheral may issue a **SINGLE** "hailing" PNAK message.

- Unless otherwise specified, the peripheral should respond to all messages addressed to it within 3 ms.

- Should a peripheral receive an incorrectly formatted character (e.g. parity or over- run error) addressed to it, then the peripheral should respond to the Executive with a PNAK. The Executive may re-issue the command, depending on the exact state of the system. The re-issuing of a command is **NOT** guaranteed, and the peripheral must make no assumption on the Executive's action.

- Should a peripheral receive a correctly formatted (address, parity and framing correct) character that it does not understand, then it **MUST** respond with an ACK (0 decimal).

The above rules are used to avoid contention problems on the Rx loop of the link. There are a number of other link rules relating to the Application Layer which are detailed in chapters 4 to 6.

4 Application Layer - VMC

4.1 General

The Vending Machine Controller (VMC) peripheral controls many of the functions in a typical vending machine. These include :

- Human interface (displays, selection buttons, etc.)
- Product dispensing
- Machine environment control (temperature, etc.).

The main area that the VMC does not control is the accumulation of credit (entered by the consumer) and the decision to allow the transaction to take place. These tasks are done by the Executive. The Application layer commands defined in Protocol A reflect the above split in tasks between VMC and Executive.

The VMC is assigned the peripheral identifier code of 001.

4.1.1 Storage of Prices

One of the areas of confusion on Protocol A is in the storage of the vend price. When the Protocol A system was originally designed, the idea was for prices to be held in the VMC. In such a system when the consumer made a selection, the VMC would match that selection to a stored price and report the price to the Executive. The Executive would then make the decision to vend based on the credit it holds and the price reported by the VMC.

The drawback to the above system is that the Executive has no information on which product was actually selected by the consumer, prior to vending. This is important in certain cases (special vend tokens, discounted vends, etc.). In addition, it makes auditing by selection simpler for the VMC. To resolve this issue, a special mode of operation was derived which is commonly termed "Price Holding".

In Price Holding, the "prices" held in the VMC are now used to represent the product line (e.g. product line 1 has its "price" set to 1, product line 2 to 2, etc.). When a selection is made, the VMC will thus report the produce line (1 based) rather than the actual product price. The Executive then uses this product line information to "look up" the actual vend price in its own price store. Using this approach allows the Executive to make (and audit) the vend by product line as well as price. Note that no change is required in the VMC's software.

This is a good example of where a system's operation can be modified without affecting the peripherals, if they all operate in the simple command driven manner which the Protocol was designed to support.

4.1.2 Monetary Units

Another area which has caused confusion on the Protocol A interface is the issue of units used for monetary values (credit, prices, etc.). To understand the chosen design approach, it is first necessary to define a method of representing monetary units.

Most currencies are decimal based, with at least two major currency units. These currency units are typically related to each other by a factor of 100 or 1000. Thus in the UK, the two major units are "pence" and "pounds" and 100 pence = 1 pound. In Tunisia 1000 "millims" is equal to 1 "dinar".

The handling of numbers with decimal points (floating point arithmetic) is very complex and memory intensive on a typical 8 bit micro-controller. Such devices are designed to handle integer numbers and can do this much more efficiently. This has lead to the definition of a term called "real money" which is an integer value of the smallest currency unit in a country. Thus in the UK a pound is 100 real money units and in Tunisia 1 dinar is 1000 real money units. This approach has been generally adopted in the Vending industry, and formalised by the BDV committee. Note that the BDV 001 standard defines the base unit for Italy as 10 Lira.

One problem with the above is that for many countries, the typical range of "real money" values for vending applications exceeds what can be stored in 8 bits. Thus for the systems that Protocol A would be used in, real money values would need to be stored as 16 bit integers.

When the initial protocol was developed microcontroller memory was an important and scarce resource. This lead to the decision to generally store and transfer monetary units in the form of an 8 bit "base unit" and an 8 bit "scaling factor" (sometimes called the coin scaling factor). The scaling factor would be a constant in a given system. The relationship to "real money" is :

real money = base units * scaling factor.

The major drawback to the above, is that the resolution of a stored value is now only as fine as the scaling factor used. Thus in the UK using a scaling factor of 1 would give a resolution to 1 penny, but a maximum range of only £2.55. Using a scaling factor of 5 would increase the range to £10.20 but the resolution is now only in steps of 5 pence.

Within the rest of this document, the units used will be specified to ensure that no confusion arises.

4.2 VMC Command Codes

The Executive can issue up to 16 different commands to the VMC peripheral. The following table shows the complete command set which is currently defined for a VMC.

Command #	Code	Attribute
0	0000	Not Used †
1	0001	STATUS
2	0010	CREDIT
3	0011	VEND
4	0100	AUDIT
5	0101	SEND AUDIT ADDRESS
6	0110	SEND AUDIT DATA
7	0111	IDENTIFY
8	1000	ACCEPT DATA
9	1001	DATA SYNC
10	1010	Not Used †
11	1011	Not Used †
12	1100	Not Used †
13	1101	Not Used †
14	1110	Not Used †
15	1111	NEGATIVE ACKNOWLEDGE

† : Although these commands are not currently used, the VMC **MUST** reply to them with an ACK (0x00) to ensure compatibility in the future.

4.3 Detail of VMC Commands / Replies

The following sections will detail each specified command, and the expected reply from the VMC to that command. Note that peripherals must only ever reply to commands which are addressed to them.

4.3.1 PNAK Reply

As was stated in section 3.2.5, if the peripheral receives a message addressed to it with incorrect parity (or other datalink layer error), then it should respond with a PNAK (0xFF). Thus a PNAK is a valid response to all of the following commands (if they were received with an error).

4.3.2 Command 1 - Status

The STATUS command is issued by the Executive during each poll of the VMC. It may be interleaved between other commands. The format of the reply from a standard (non-BDV) Protocol A VMC is shown in fig 4.1.

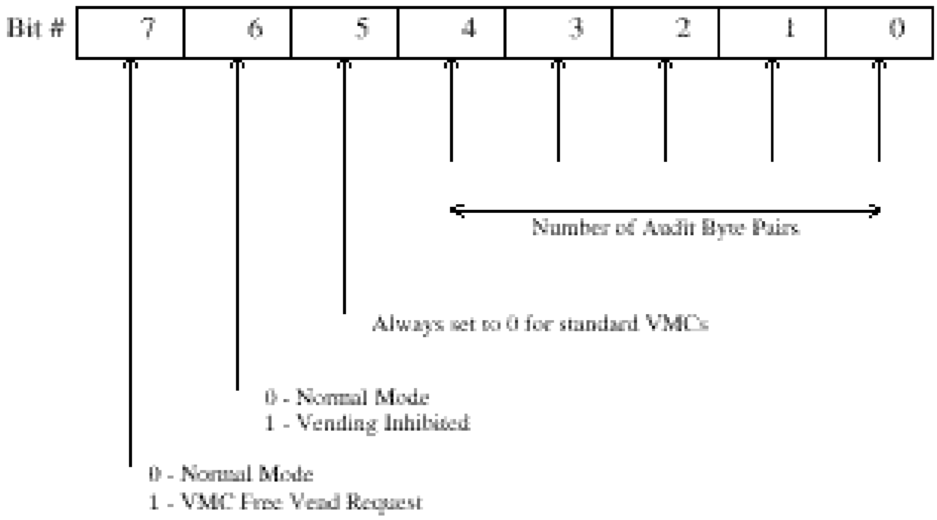


Fig 7 : Reply to status

The Free Vend Request flag must be set by the VMC when the consumer requests a vend, and the price programmed in the VMC for that vend is zero. The flag should **NOT** be set until such a vend is selected. The normal effect of this flag being set is to cause the Executive to inhibit payment acceptance and to issue a CREDIT command to the VMC, even if there is no credit in the system. Note that the Executive will **NOT** automatically issue a VEND command in response to this flag being set.

Free vend operation will work even if the system is operating in "price holding" (see section 4.1.1), as the price line data held in the VMC is 1 based, but it is still possible to set a VMC "price" to zero, and hence effect the above operation.

The Vending Inhibited flag must be set if the VMC is unable to vend products (e.g. there is a fault present in the VMC, or the products are all sold out). The effect of this flag being set is to disable all credit acceptance into the system. The Executive will still continue to poll the VMC with STATUS commands, but will not issue any CREDIT commands. The Vending Inhibited flag will take precedence over the Free Vend Request flag.

Bit 5 indicates the type of VMC. If this bit is 0, then it indicates that a "standard" VMC is fitted. A BDV001 VMC will set this bit to 1. Verification of the VMC type should be done on power-up by the Executive. Note that the BDV001 protocol is outside the scope of this document.

A non-zero value in the Number of Audit Byte Pairs field indicates to the Executive that the VMC has audit data available. See section 4.3.6 for details of transfer of audit data from the VMC to the Executive.

4.3.3 Command 2 - Credit

If the system state is such that vending is allowed, then the Executive will issue the CREDIT command on each poll of the VMC. Note that this is **NOT** the only time that the Executive can issue the CREDIT command.

The reply from the VMC will indicate to the Executive if a vend has been requested by the Consumer or not. The reply is a binary number as follows :

- Vend price in base units (0 - 250 decimal) - Vend requested
- 254 decimal - No vend requested

If the reply sent by the VMC indicates a vend is requested, the Executive will then verify that the credit covers the price of the vend, and if so may then issue a VEND command. Note that on many systems the decision to vend may be based on more than just the credit covering the price of the selection. Hence there is no guarantee that a VEND command will follow the CREDIT command.

If the system is operating in Price Holding, then the reply from the VMC will represent the product line if it is in the range 1 to 250. The Executive then uses this product line information to "look up" the actual vend price in its own price store. It will then make the decision to vend or not. As above, the decision to vend may depend on more than just the vend price, hence there is no guarantee that a VEND command will follow the CREDIT command.

4.3.4 Command 3 - Vend

The VEND command is issued by the Executive if the price of the vend requested by the VMC is less than or equal to the value of credit held by the Executive, and any other conditions which apply are met.

The VMC will take this command as a signal to start the vend (on the product which was selected and reported in the reply to the CREDIT command). If the VMC does not correctly receive the VEND command (e.g. parity or over-run error), it must immediately (within 3 ms) respond with a PNAK.

The VMC's reply to a valid VEND command is one of the few cases where it is permissible for a peripheral to not response immediately to the Executive. This is to allow the VMC time to carry out the vend and then report its outcome in the reply to the Executive. The Executive will typically allow at least 60 seconds for a reply to the VEND command (this does however depend on the implementation of the Executive). The reply from a Protocol A VMC will take the following form :

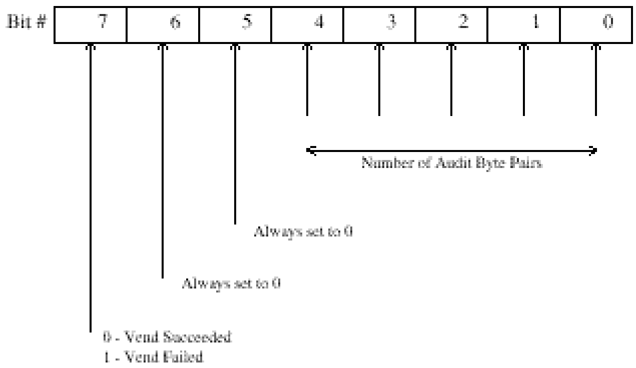


Fig 8 : Reply to Vend

A non-zero value in the Number of Audit Byte Pairs field indicates to the Executive that the VMC has audit data available. See section 4.3.6 for details of how this data is transferred to the Executive.

If the VMC reports that the vend has failed, then the Executive will take the appropriate action, which may include refunding the cost of the vend. Note that the actions of the Executive is not part of this specification, and does not influence the design of a VMC or its responses to commands.

4.3.5 Command 4 - Audit

The AUDIT command instructs the VMC to report how much (if any) audit data it currently has stored. The Executive will normally issue the AUDIT command when there is an audit request on the system. This audit request may come from the Audit Storage Unit peripheral (see chapter 5) or from another source. The AUDIT command allows the Executive to get (and possibly pass on) the most up-to-date audit data from the VMC.

The reply from the VMC is a binary number in the range 0 to 250 indicating the number of Audit Byte Pairs of audit data which the VMC has to report. See section 4.3.6 for details of how this audit information is transferred to the Executive. Note that most VMCs will report audit data "as it happens", for example after a vend, and in such cases the reply to the AUDIT command may well be 0x00 (no data byte pairs pending).

4.3.6 Command 5 - Send Audit Address

The SEND AUDIT ADDRESS command is issued by the Executive if the VMC indicates it has audit data byte pairs to report (see sections 4.3.2, 4.3.4, 4.3.5, 4.3.8). The Executive will send a SEND AUDIT ADDRESS command followed by a SEND AUDIT DATA for each Audit Byte Pair the VMC indicates it has.

All audit data transfer on the Protocol A link is via "Audit Byte Pairs". An Audit Byte Pair consists of 2 characters (each in the range 0 to 250). The first character specifies an address of an audit data register in the Audit Storage Unit peripheral. The second character specifies the data that is to be "loaded" in the specified register. Note that the term "loaded" does not imply a simple overwrite of any data already stored in the Audit Storage Unit. See chapter 5 for details of how the Audit Storage Unit operates, and how data sent to its registers is handled.

The allowed replies to the SEND AUDIT ADDRESS command are :

- 0 - 250 decimal - Address to audit the data to
- 254 decimal - Sync lost (SEND AUDIT DATA expected)

The "sync lost" reply is sent if the VMC was not expecting the audit command just received. This mis-match could occur due to message loss due to noise on the link, or VMC power loss.

Note than a VMC does not need to report audit data if the application in which it is used does not require this.

4.3.7 Command 6 - Send Audit Data

The SEND AUDIT DATA command is issued by the Executive (after the SEND AUDIT ADDRESS) if the VMC indicates it has audit data to report. The Executive sends a SEND AUDIT ADDRESS command followed by a SEND AUDIT DATA for each Audit Byte Pair the VMC indicates it has.

The allowed replies to the SEND AUDIT DATA command are :

- 0 - 250 decimal - Audit data byte
- 254 decimal - Sync lost (SEND AUDIT ADD. expected)

4.3.8 Command 7 - Identify

The IDENTIFY command can be issued by the Executive after power-up to request the VMC's identification data.

The reply from the VMC is a binary number in the range 0 to 250 indicating the number of Audit Byte Pairs of identification data which the VMC has to report. See section 4.3.6 for details of how this audit information is transferred to the Executive.

4.3.9 Command 8 - Accept Data

The ACCEPT DATA command allows the Executive to send a block of data (8 nibbles) over the link to the VMC. This data is for display purposes only. The reply to the ACCEPT DATA command is a simple ACK (0x00).

The data sent over will normally be the credit within the system, and hence is updated on any change in the credit accumulator. However, credit information is **NOT** the only display information that the Executive may send over. Thus it is vital that the VMC **NEVER** assumes that the data received represents actual system credit (or any other item), and never uses the data for any of its internal control logic. The VMC must simply treat the data as a value to be displayed.

To transfer data to the VMC, the Executive will issue the ACCEPT DATA command, and if the VMC reply is OK (i.e. an ACK), it will send a number of data nibbles to the VMC (bit 4 clear for data nibble transfer). The VMC is expected to acknowledge each of these data nibbles with an ACK (0x00) if it was received with no datalink layer errors, or a PNAK if a datalink layer error (e.g. parity) occurred.

At the end of the transfer, the Executive will issue a DATA SYNC command to signal the end of transfer to the VMC.

The 8 data nibbles will always be sent in the following order :

Least significant 4 bits (bits 0 to 3) of a 16 bit base unit value
Next 4 bits (bits 4 to 7) of a 16 bit base value
Next 4 bits (bits 8 to 11) of a 16 bit base value
Most significant 4 bits (bits 12 to 15) of a 16 bit base value
Least significant 4 bits (bits 0 to 3) of a 8 bit scaling factor value
Most significant 4 bits (bits 4 to 7) of a 8 bit scaling factor value
Decimal point position indicator
Exact change status flag

Fig 4.3 details the messages sent from the Executive during a data transfer sequence. Note that the VMC is expected to ACK each message as it is received.

Bit 7							Bit 0	
0	0	1	1	1	0	0	0	ACCEPT DATA cmd
0	0	1	0	Base Unit Value (bit 3 .. bit 0)				Data nibble 1
0	0	1	0	Base Unit Value (bit7 .. bit 4)				Data nibble 2
0	0	1	0	Base Unit Value (bit11 .. bit 8)				Data nibble 3
0	0	1	0	Base Unit Value (bit15 .. bit 12)				Data nibble 4
0	0	1	0	Scaling Factor (bit3 .. bit 0)				Data nibble 5
0	0	1	0	Scaling Factor (bit 7.. bit 4)				Data nibble 6
0	0	1	0	Decimal Point Indicator				Data nibble 7
0	0	1	0	0	0	0	EC Flag	Data nibble 8
0	0	1	1	1	0	0	1	DATA SYNC cmd

Fig 9 : Data transfer messages form executive to VMC

The integer "real money" value to be displayed should be computed by the VMC by taking the 16 bit base unit value in data nibbles 1 to 4 and multiplying it by the 8 bit scaling factor in nibbles 5 to 6. This (in theory) results in a 24 bit "real money" integer. In practice it should be safe for the VMC to truncate the 16 bit * 8 bit multiplication to a 16 bit result.

To display this to the user in a normal currency format, a decimal point is normally required. The Decimal Point Indicator is used to indicate to the VMC where it should place the decimal point. The allowed values for the Decimal Point Indicator are :

0001	Display no digits to right of the decimal point (e.g 1234.)
0010	Display 1 digit to right of decimal point (e.g. 123.4)
0100	Display 2 digits to right of decimal point (e.g. 12.34)
1000	Display 3 digits to right of decimal point (e.g. 1.234)

If the Exact Change (EC) Flag is set to 1, then the VMC that should display its "Use Exact Change" notice to the consumer. The exact form of this notice is left for the VMC designer to define. This flag will normally be set to 1 if the Executive detects that the change available in the system may not be sufficient to ensure that the consumer always gets a correct refund. The VMC should remove its "Use Exact Change" notice if the EC Flag is 0.

Note that this specification does not impose any timing restrictions on the commands from the Executive when transferring data to the VMC. Thus it is important that the VMC does not assume any timings concerning this function. Also note that the Executive may abort a data transfer sequence at any point.

4.3.10 Command 9 - Data Sync

The DATA SYNC command is issued by the Executive to terminate a data transfer to the VMC. The reply to a the DATA SYNC command is a simple ACK (0x00). See section 4.3.9 for details of a data transfer session.

4.3.11 Command 15 - Negative Acknowledge

The NEGATIVE ACKNOWLEDGE (NAK) command is issued by the Executive to the VMC when it detects a parity or over-run error on the VMC's reply character. On receiving this command, the VMC **MUST** repeat its last transmission. The Executive will wait with all functions inhibited until a reply is received.

4.4 Example Message Sequences

This section contains a number of example message sequences. Note that these are typical sequences for information only and do not form part of the formal protocol specification. Note also that the time between messages sent from the Executive is not part of this specification, and must **NOT** be implied from these examples.

4.4.1 Polling of VMC (No Activity)

The following sequence illustrates the commands sent to the VMC and its expected reply when there is no activity in the system, and no credit is present.

Executive -> VMC	Meaning	Reply	Meaning
00110001b (31h)	STATUS	00000000b	ACK (Nothing to report)
00110001b (31h)	STATUS	00000000b	ACK (Nothing to report)
00110001b (31h)	STATUS	00000000b	ACK (Nothing to report)

4.4.2 Polling of VMC (Vending Inhibited)

The following sequence illustrates the commands sent to the VMC and its expected reply when vending becomes inhibited (between the 2nd & 3rd poll) and there is no credit in the system.

Executive -> VMC	Meaning	Reply	Meaning
00110001b (31h)	STATUS	00000000b	ACK (Nothing to report)
00110001b (31h)	STATUS	00000000b	ACK (Nothing to report)
00110001b (31h)	STATUS	01000000b	Vending Inhibited
00110001b (31h)	STATUS	01000000b	Vending Inhibited

4.4.3 Polling of VMC (Vending Enabled)

The following sequence illustrates the commands sent to the VMC and its expected reply when vending becomes enabled (between the 2nd & 3rd poll) and there is no credit in the system.

Executive -> VMC	Meaning	Reply	Meaning
00110001b (31h)	STATUS	01000000b	Vending Inhibited
00110001b (31h)	STATUS	01000000b	Vending Inhibited
00110001b (31h)	STATUS	00000000b	ACK (Nothing to report)
00110001b (31h)	STATUS	00000000b	ACK (Nothing to report)

4.4.4 Display Data Transfer to VMC

The following sequence illustrates the commands sent to the VMC and its expected reply when credit (10 base units, scaling factor of 1, decimal point of 2 places, exact change indication off) is entered in the system (after the 2nd poll).

Executive -> VMC	Meaning	Reply	Meaning
00110001b (31h)	STATUS	00000000b	ACK (Nothing to report)
00110001b (31h)	STATUS	00000000b	ACK (Nothing to report)
00111000b (38h)	ACCEPT DATA	00000000b	ACK
00101010b (2Ah)	credit (00.10)	00000000b	ACK
00100000b (20h)	credit	00000000b	ACK
00100000b (20h)	credit	00000000b	ACK
00100000b (20h)	credit	00000000b	ACK
00100001b (21h)	scaling factor	00000000b	ACK
00100000b (20h)	scaling factor	00000000b	ACK
00100100b (24h)	decimal point	00000000b	ACK
00100000b (20h)	exact change flag	00000000b	ACK
00111001b (39h)	DATA SYNC	00000000b	ACK
00110001b (31h)	STATUS	00000000b	ACK (Nothing to report)
00110010b (32h)	CREDIT	11111110b	No vend request

4.4.5 Polling of VMC (Credit in System)

The following sequence illustrates the commands sent to the VMC and its expected reply when credit is present in the Executive.

Executive -> VMC	Meaning	Reply	Meaning
00110001b (31h)	STATUS	00000000b	ACK (Nothing to report)
00110010b (32h)	CREDIT	11111110b	No vend request
00110001b (31h)	STATUS	00000000b	ACK (Nothing to report)
00110010b (32h)	CREDIT	11111110b	No vend request
00110001b (31h)	STATUS	00000000b	ACK (Nothing to report)
00110010b (32h)	CREDIT	11111110b	No vend request

4.4.6 Single Vend Sequence

The following sequence illustrates the commands sent to the VMC and its expected reply when credit (10 units) is present in the Executive and a vend request of 5 base units is made (after the 2nd poll). In this example the Executive is only granting a single vend, and will pay change after the vend has been completed.

Executive -> VMC	Meaning	Reply	Meaning
00110001b (31h)	STATUS	00000000b	ACK (Nothing to report)
00110010b (32h)	CREDIT	11111110b	No vend request
00110001b (31h)	STATUS	00000000b	ACK (Nothing to report)
00110010b (32h)	CREDIT	00000101b	Vend request - 5 units
00111000b (38h)	ACCEPT DATA	00000000b	ACK
00100101b (25h)	credit (00.05)	00000000b	ACK
00100000b (20h)	credit	00000000b	ACK
00100000b (20h)	credit	00000000b	ACK
00100000b (20h)	credit	00000000b	ACK
00100001b (21h)	scaling factor	00000000b	ACK
00100000b (20h)	scaling factor	00000000b	ACK
00100100b (24h)	decimal point	00000000b	ACK
00100000b (20h)	exact change flag	00000000b	ACK
00111001b (39h)	DATA SYNC	00000000b	ACK
00110011b (33h)	VEND ... vending ...		
		00000000b	Vend OK
00111000b (38h)	ACCEPT DATA	00000000b	ACK
00100000b (20h)	credit (00.00)	00000000b	ACK
00100000b (20h)	credit	00000000b	ACK
00100000b (20h)	credit	00000000b	ACK
00100000b (20h)	credit	00000000b	ACK
00100001b (21h)	scaling factor	00000000b	ACK
00100000b (20h)	scaling factor	00000000b	ACK
00100100b (24h)	decimal point	00000000b	ACK
00100000b (20h)	exact change flag	00000000b	ACK
00111001b (39h)	DATA SYNC	00000000b	ACK
00110001b (31h)	STATUS	00000000b	ACK (Nothing to report)
00110001b (31h)	STATUS	00000000b	ACK (Nothing to report)
00110001b (31h)	STATUS	00000000b	ACK (Nothing to report)

4.4.7 Single Vend Sequence - Vend Failed

The following sequence illustrates the commands sent to the VMC and its expected reply when credit (10 units) is present in the Executive and a vend request of 5 base units is made (after the 2nd poll). In this example the vend fails.

Executive -> VMC	Meaning	Reply	Meaning
00110001b (31h)	STATUS	00000000b	ACK (Nothing to report)
00110010b (32h)	CREDIT	11111110b	No vend request
00110001b (31h)	STATUS	00000000b	ACK (Nothing to report)
00110010b (32h)	CREDIT	00000101b	Vend request - 5 units
00111000b (38h)	ACCEPT DATA	00000000b	ACK
00100101b (25h)	credit (00.05)	00000000b	ACK
00100000b (20h)	credit	00000000b	ACK
00100000b (20h)	credit	00000000b	ACK
00100000b (20h)	credit	00000000b	ACK
00100001b (21h)	scaling factor	00000000b	ACK
00100000b (20h)	scaling factor	00000000b	ACK
00100100b (24h)	decimal point	00000000b	ACK
00100000b (20h)	exact change flag	00000000b	ACK
00111001b (39h)	DATA SYNC	00000000b	ACK
00110011b (33h)	VEND ... vending ...		
		10000000b	Vend Failed
00111000b (38h)	ACCEPT DATA	00000000b	ACK
00101010b (2Ah)	credit (00.10)	00000000b	ACK
00100000b (20h)	credit	00000000b	ACK
00100000b (20h)	credit	00000000b	ACK
00100000b (20h)	credit	00000000b	ACK
00100001b (21h)	scaling factor	00000000b	ACK
00100000b (20h)	scaling factor	00000000b	ACK
00100100b (24h)	decimal point	00000000b	ACK
00100000b (20h)	exact change flag	00000000b	ACK
00111001b (39h)	DATA SYNC	00000000b	ACK
00110001b (31h)	STATUS	00000000b	ACK (Nothing to report)
00110010b (32h)	CREDIT	11111110b	No vend request
00110001b (31h)	STATUS	00000000b	ACK (Nothing to report)

4.4.8 Multiple Vend Sequence

The following sequence illustrates the commands sent to the VMC and its expected reply when credit (10 units) is present in the Executive and a vend request of 5 base units is made (after the 2nd poll). In this example the Executive allows multiple vends in a single session and a 2nd vend request of 2 base units follows.

Executive -> VMC	Meaning	Reply	Meaning
00110001b (31h)	STATUS	00000000b	ACK (Nothing to report)
00110010b (32h)	CREDIT	11111110b	No vend request
00110001b (31h)	STATUS	00000000b	ACK (Nothing to report)
00110010b (32h)	CREDIT	00000101b	Vend request - 5 units
00111000b (38h)	ACCEPT DATA	00000000b	ACK
00100101b (25h)	credit (00.05)	00000000b	ACK
00100000b (20h)	credit	00000000b	ACK
00100000b (20h)	credit	00000000b	ACK
00100000b (20h)	credit	00000000b	ACK
00100001b (21h)	scaling factor	00000000b	ACK
00100000b (20h)	scaling factor	00000000b	ACK
00100100b (24h)	decimal point	00000000b	ACK
00100000b (20h)	exact change flag	00000000b	ACK
00111001b (39h)	DATA SYNC	00000000b	ACK
00110011b (33h)	VEND ... vending ...		
		00000000b	Vend OK
00110001b (31h)	STATUS	00000000b	ACK (Nothing to report)
00110010b (32h)	CREDIT	11111110b	No vend request
00110001b (31h)	STATUS	00000000b	ACK (Nothing to report)
00110010b (32h)	CREDIT	00000010b	Vend request - 2 units
00111000b (38h)	ACCEPT DATA	00000000b	ACK
00100011b (23h)	credit (00.03)	00000000b	ACK
00100000b (20h)	credit	00000000b	ACK
00100000b (20h)	credit	00000000b	ACK
00100000b (20h)	credit	00000000b	ACK
00100001b (21h)	scaling factor	00000000b	ACK
00100000b (20h)	scaling factor	00000000b	ACK
00100100b (24h)	decimal point	00000000b	ACK
00100000b (20h)	exact change flag	00000000b	ACK
00111001b (39h)	DATA SYNC	00000000b	ACK
00110011b (33h)	VEND ... vending ...		
		00000000b	Vend OK
00110001b (31h)	STATUS	00000000b	ACK (Nothing to report)
00110010b (32h)	CREDIT	11111110b	No vend request
00110001b (31h)	STATUS	00000000b	ACK (Nothing to report)

4.4.9 Credit Return

The following sequence illustrates the commands sent to the VMC and its expected reply when credit in the system is returned to the User.

Executive -> VMC	Meaning	Reply	Meaning
00110001b (31h)	STATUS	00000000b	ACK (Nothing to report)
00110010b (32h)	CREDIT	11111110b	No vend request
00110001b (31h)	STATUS	00000000b	ACK (Nothing to report)
00110010b (32h)	CREDIT	11111110b	No vend request
00111000b (38h)	ACCEPT DATA	00000000b	ACK
00100000b (20h)	credit (00.00)	00000000b	ACK
00100000b (20h)	credit	00000000b	ACK
00100000b (20h)	credit	00000000b	ACK
00100000b (20h)	credit	00000000b	ACK
00100001b (21h)	scaling factor	00000000b	ACK
00100000b (20h)	scaling factor	00000000b	ACK
00100100b (24h)	decimal point	00000000b	ACK
00100000b (20h)	exact change flag	00000000b	ACK
00111001b (39h)	DATA SYNC	00000000b	ACK
00110001b (31h)	STATUS	00000000b	ACK (Nothing to report)
00110001b (31h)	STATUS	00000000b	ACK (Nothing to report)
00110001b (31h)	STATUS	00000000b	ACK (Nothing to report)

4.4.10 Free Vend Request

The following sequence illustrates the commands sent to the VMC and its expected reply when there is no credit in the system, but the VMC signals there is a free vend request (after the 2nd poll).

Executive -> VMC	Meaning	Reply	Meaning
00110001b (31h)	STATUS	00000000b	ACK (Nothing to report)
00110001b (31h)	STATUS	00000000b	ACK (Nothing to report)
00110001b (31h)	STATUS	10000000b	Free vend request
00110010b (32h)	CREDIT	00000000b	Vend price = 0
00110011b (33h)	VEND ... vending ...		
		00000000b	Vend OK
00110001b (31h)	STATUS	00000000b	ACK (Nothing to report)
00110001b (31h)	STATUS	00000000b	ACK (Nothing to report)

4.4.11 Single Vend Sequence (Price Holding)

The following sequence illustrates the commands sent to the VMC and its expected reply when credit (10 units) is present in the Executive and a vend request is made on price selection #1. In this example the prices are held by the Executive (Price Holding mode), and the price of selection #1 is 5 base units. The Executive is in single vend mode , and will pay change after the vend has been completed.

Executive -> VMC	Meaning	Reply	Meaning
00110001b (31h)	STATUS	00000000b	ACK (Nothing to report)
00110010b (32h)	CREDIT	11111110b	No vend request
00110001b (31h)	STATUS	00000000b	ACK (Nothing to report)
00110010b (32h)	CREDIT	00000001b	Vend request (selection 1)
00111000b (38h)	ACCEPT DATA	00000000b	ACK
00100101b (25h)	credit (00.05)	00000000b	ACK
00100000b (20h)	credit	00000000b	ACK
00100000b (20h)	credit	00000000b	ACK
00100000b (20h)	credit	00000000b	ACK
00100001b (21h)	scaling factor	00000000b	ACK
00100000b (20h)	scaling factor	00000000b	ACK
00100100b (24h)	decimal point	00000000b	ACK
00100000b (20h)	exact change flag	00000000b	ACK
00111001b (39h)	DATA SYNC	00000000b	ACK
00110011b (33h)	VEND ... vending ...		
		00000000b	Vend OK
00111000b (38h)	ACCEPT DATA	00000000b	ACK
00100000b (20h)	credit (00.00)	00000000b	ACK
00100000b (20h)	credit	00000000b	ACK
00100000b (20h)	credit	00000000b	ACK
00100000b (20h)	credit	00000000b	ACK
00100001b (21h)	scaling factor	00000000b	ACK
00100000b (20h)	scaling factor	00000000b	ACK
00100100b (24h)	decimal point	00000000b	ACK
00100000b (20h)	exact change flag	00000000b	ACK
00111001b (39h)	DATA SYNC	00000000b	ACK
00110001b (31h)	STATUS	00000000b	ACK (Nothing to report)
00110001b (31h)	STATUS	00000000b	ACK (Nothing to report)
00110001b (31h)	STATUS	00000000b	ACK (Nothing to report)

4.4.12 Free Vend Sequence (Price Holding)

The following sequence illustrates the commands sent to the VMC and its expected reply when the Executive is in Price Holding and selection # 2 has a price of 0. There is no credit in the system. Note that the first selection made (#1) is ignored (no credit to cover it).

Executive -> VMC	Meaning	Reply	Meaning
00110001b (31h)	STATUS	00000000b	ACK (Nothing to report)
00110010b (32h)	CREDIT	11111110b	No vend request
00110001b (31h)	STATUS	00000000b	ACK (Nothing to report)
00110010b (32h)	CREDIT	00000001b	Vend request (selection #1)
00110001b (31h)	STATUS	00000000b	ACK (Nothing to report)
00110010b (32h)	CREDIT	11111110b	No vend request
00110001b (31h)	STATUS	00000000b	ACK (Nothing to report)
00110010b (32h)	CREDIT	00000010b	Vend request (selection #2)
00111000b (38h)	ACCEPT DATA	00000000b	ACK
00100000b (20h)	credit (00.00)	00000000b	ACK
00100000b (20h)	credit	00000000b	ACK
00100000b (20h)	credit	00000000b	ACK
00100000b (20h)	credit	00000000b	ACK
00100001b (21h)	scaling factor	00000000b	ACK
00100000b (20h)	scaling factor	00000000b	ACK
00100100b (24h)	decimal point	00000000b	ACK
00100000b (20h)	exact change flag	00000000b	ACK
00111001b (39h)	DATA SYNC	00000000b	ACK
00110011b (33h)	VEND ... vending ...		
		00000000b	Vend OK
00111000b (38h)	ACCEPT DATA	00000000b	ACK
00100000b (20h)	credit (00.00)	00000000b	ACK
00100000b (20h)	credit	00000000b	ACK
00100000b (20h)	credit	00000000b	ACK
00100000b (20h)	credit	00000000b	ACK
00100001b (21h)	scaling factor	00000000b	ACK
00100000b (20h)	scaling factor	00000000b	ACK
00100100b (24h)	decimal point	00000000b	ACK
00100000b (20h)	exact change flag	00000000b	ACK
00111001b (39h)	DATA SYNC	00000000b	ACK
00110001b (31h)	STATUS	00000000b	ACK (Nothing to report)
00110010b (32h)	CREDIT	11111110b	No vend request
00110001b (31h)	STATUS	00000000b	ACK (Nothing to report)
00110010b (32h)	CREDIT	11111110b	No vend request
00110001b (31h)	STATUS	00000000b	ACK (Nothing to report)

4.4.13 Price Display Operation

The following shows an example of what is termed "Price Display" operation. This is a special mode of operation available on some Executives which allows the display of a selection price. Note that the VMC's operation is completely unchanged. The example shows a system with no credit, and the Consumer makes a selection which has a vend price of 20 units. This is displayed to him for 2 seconds.

Executive -> VMC	**Meaning**	**Reply**	**Meaning**
00110001b (31h)	STATUS	00000000b	ACK (Nothing to report)
00110010b (32h)	CREDIT	11111110b	No vend request
00110001b (31h)	STATUS	00000000b	ACK (Nothing to report)
00110010b (32h)	CREDIT	00010100b	Vend request (20 units)
00111000b (38h)	ACCEPT DATA	00000000b	ACK
00100100b (24h)	credit (00.20)	00000000b	ACK
00100001b (21h)	credit	00000000b	ACK
00100000b (20h)	credit	00000000b	ACK
00100000b (20h)	credit	00000000b	ACK
00100001b (21h)	scaling factor	00000000b	ACK
00100000b (20h)	scaling factor	00000000b	ACK
00100100b (24h)	decimal point	00000000b	ACK
00100000b (20h)	exact change flag	00000000b	ACK
00111001b (39h)	DATA SYNC	00000000b	ACK
00110001b (31h)	STATUS	00000000b	ACK (Nothing to report)
.			
<2 seconds later >			
.			
00110001b (31h)	STATUS	00000000b	ACK (Nothing to report)
00111000b (38h)	ACCEPT DATA	00000000b	ACK
00100000b (20h)	credit (00.00)	00000000b	ACK
00100000b (20h)	credit	00000000b	ACK
00100000b (20h)	credit	00000000b	ACK
00100000b (20h)	credit	00000000b	ACK
00100001b (21h)	scaling factor	00000000b	ACK
00100000b (20h)	scaling factor	00000000b	ACK
00100100b (24h)	decimal point	00000000b	ACK
00100000b (20h)	exact change flag	00000000b	ACK
00111001b (39h)	DATA SYNC	00000000b	ACK
00110001b (31h)	STATUS	00000000b	ACK (Nothing to report)
00110010b (32h)	CREDIT	11111110b	No vend request
00110001b (31h)	STATUS	00000000b	ACK (Nothing to report)

4.4.14 Price Display Operation (with Price Holding)

The following shows an example of "Price Display" operation combined with "Price Holding". The example shows a system with no credit, and the Consumer makes a selection (line 3) which has a vend price of 20 units. This is displayed to him for 2 seconds.

Executive -> VMC	Meaning	Reply	Meaning
00110001b (31h)	STATUS	00000000b	ACK (Nothing to report)
00110010b (32h)	CREDIT	11111110b	No vend request
00110001b (31h)	STATUS	00000000b	ACK (Nothing to report)
00110010b (32h)	CREDIT	00000011b	Vend request (selection #3)
00111000b (38h)	ACCEPT DATA	00000000b	ACK
00100100b (24h)	credit (00.20)	00000000b	ACK
00100001b (21h)	credit	00000000b	ACK
00100000b (20h)	credit	00000000b	ACK
00100000b (20h)	credit	00000000b	ACK
00100001b (21h)	scaling factor	00000000b	ACK
00100000b (20h)	scaling factor	00000000b	ACK
00100100b (24h)	decimal point	00000000b	ACK
00100000b (20h)	exact change flag	00000000b	ACK
00111001b (39h)	DATA SYNC	00000000b	ACK
00110001b (31h)	STATUS	00000000b	ACK (Nothing to report)
.			
<2 seconds later >			
.			
00110001b (31h)	STATUS	00000000b	ACK (Nothing to report)
00111000b (38h)	ACCEPT DATA	00000000b	ACK
00100000b (20h)	credit (00.00)	00000000b	ACK
00100000b (20h)	credit	00000000b	ACK
00100000b (20h)	credit	00000000b	ACK
00100000b (20h)	credit	00000000b	ACK
00100001b (21h)	scaling factor	00000000b	ACK
00100000b (20h)	scaling factor	00000000b	ACK
00100100b (24h)	decimal point	00000000b	ACK
00100000b (20h)	exact change flag	00000000b	ACK
00111001b (39h)	DATA SYNC	00000000b	ACK
00110001b (31h)	STATUS	00000000b	ACK (Nothing to report)
00110010b (32h)	CREDIT	11111110b	No vend request
00110001b (31h)	STATUS	00000000b	ACK (Nothing to report)

4.4.15 Audit Request - No Data

The following sequence illustrates the AUDIT command when the VMC has no static audit data to report.

Executive -> VMC	Meaning	Reply	Meaning
00110001b (31h)	STATUS	00000000b	ACK (Nothing to report)
00110001b (31h)	STATUS	00000000b	ACK (Nothing to report)
00110100b (34h)	AUDIT	00000000b	ACK (No audit data byte pairs)
00110001b (31h)	STATUS	00000000b	ACK (Nothing to report)
00110001b (31h)	STATUS	00000000b	ACK (Nothing to report)

4.4.16 Audit Request - Data

The following sequence illustrates the AUDIT command when the VMC has 2 audit byte pairs to report.

Executive -> VMC	Meaning	Reply	Meaning
00110100b (34h)	AUDIT	00000010b	2 audit byte pairs
00110101b (35h)	SEND AUDIT ADDRESS	00011001b	1st address = 25
00110110b (36h)	SEND AUDIT DATA	00000010b	Data for loc 25 = 2
00110101b (35h)	SEND AUDIT ADDRESS	01000110b	2nd address = 70
00110110b (36h)	SEND AUDIT DATA	00000010b	Data for loc 70 = 2
00110001b (31h)	STATUS	00000000b	ACK (Nothing to report)
00110001b (31h)	STATUS	00000000b	ACK (Nothing to report)

4.4.17 Identify Request - Data

The following sequence illustrates the IDENTIFY command when the VMC has 1 audit byte pair to report.

Executive -> VMC	Meaning	Reply	Meaning
00110111b (37h)	IDENTIFY	00000001b	1 audit byte pair
00110101b (35h)	SEND AUDIT ADDRESS	01111001b	1st address = 121
00110110b (36h)	SEND AUDIT DATA	00101010b	Data for loc 121 = 42
00110001b (31h)	STATUS	00000000b	ACK (Nothing to report)
00110001b (31h)	STATUS	00000000b	ACK (Nothing to report)

4.4.18 Negative Acknowledge

The following sequence illustrates the NEGATIVE ACKNOWLEDGE command when the Executive detects an error in the VMC's reply (to the 2nd STATUS command).

Executive -> VMC	Meaning	Reply	Meaning
00110001b (31h)	STATUS	00000000b	ACK (Nothing to report)
00110001b (31h)	STATUS	00000000b *	corrupt ACK
00111111b (3Fh)	NEGATIVE ACKNOWLEDGE	00000000b	Re-send of last reply
00110001b (31h)	STATUS	00000000b	ACK (Nothing to report)

5 Audit Storage Unit Application Layer

5.1 General

The Audit Storage Unit (peripheral identifier : 010) provides a non-volatile store for transaction information within the system. This includes :

- Number & value of vends
- Value of cash / cashless payment deposited into system
- Value of cash paid as change

On many early systems the Audit Storage Unit (ASU) was a physically separate device on the Protocol A link. On later systems, the functionality of this (logical) node was incorporated into another peripheral (e.g. the VMC). The physical location of the ASU does not affect any of the Protocol A commands to it.

The Audit Storage Unit for a Protocol A system is a very simple device. As far as the Protocol A interface is concerned, the unit consists of a number of pre-defined "addresses" to which the Executive "writes" a single byte of data to. The Audit unit converts these single byte "writes" to an increment or overwrite of its internal accumulator for the specified "address". The actual action depends on the type of accumulator.

The retrieval of data from the ASU is done via a separate data port on the unit. Details of this port are outside of the scope of this document.

5.2 Audit Addresses

The ASU unit allows up to 256 "addresses". On early systems each "address" actually corresponded to a physical RAM byte location in the ASU. As most accumulators were 2 bytes wide, this meant that the "addresses" corresponding to the high order byte of each accumulator were reserved, and should **NEVER** be written to. This is the reason for the gaps in the following table of addresses. Most modern ASU's use larger accumulators, and the "address" is now used only as a pointer into a much larger memory array. However to ensure backward compatibility, the reserved addresses are not used.

The Audit Storage Unit addresses are grouped into the following classes :

- System identification data
- Totals data
- Interims data
- VMC specific data

ADDRESS	CONTENTS	COMMENTS
0	Audit Unit Serial number	Reserved for internal Audit Unit use
1	Audit Unit Serial number	Reserved for internal Audit Unit use
2	Reserved	Reserved for internal Audit Unit use
3	Coin Mechanism ID	Reserved for internal Audit Unit use
4	Coin Scaling Factor	Reserved for internal Audit Unit use
5	Transaction Number	Reserved for internal Audit Unit use
6	Power Interruption Count	Reserved for internal Audit Unit use
7	Total - Money to tubes	In base units
8	Reserved	
9	Total - Money to cashbox	In base units
10	Reserved	
11	Total - Money as change	In base units
12	Reserved	
13	Total - Money manually dispensed	In base units
14	Reserved	
15	Total - Overpay value	In base units
16	Reserved	
17	Total - Pay vend value	In base units
18	Reserved	
19	Total - Pay vend value (in exact change)	In base units
20	Reserved	
21	Total - Discount value	In base units
22	Reserved	
23	Total - Misc. vend value	In base units
24	Reserved	
25	Total - Price line 1 vend value	In base units
26	Reserved	
27	Total - Price line 2 vend value	In base units
28	Reserved	
29	Total - Price line 3 vend value	In base units
30	Reserved	
31	Total - Price line 4 vend value	In base units
32	Reserved	
33	Total - Price line 5 vend value	In base units
34	Reserved	
35	Total - Price line 6 vend value	In base units
36	Reserved	
37	Total - Price line 7 vend value	In base units
38	Reserved	
39	Total - Price line 8 vend value	In base units
40	Reserved	
41	Total - Price line 9 vend value	In base units
42	Reserved	
43	Total - Price line 10 vend value	In base units
44	Reserved	

45	Total - Event counter 1	
46	Reserved	
47	Total - Event counter 2	
48	Reserved	
49	Total - Event counter 3	
50	Reserved	
51	Total - Event counter 4	
52	Interim - Money to tubes	In base units
53	Reserved	
54	Interim - Money to cashbox	In base units
55	Reserved	
56	Interim - Money as change	In base units
57	Reserved	
58	Interim - Money manually dispensed	In base units
59	Reserved	
60	Interim - Overpay value	In base units
61	Reserved	
62	Interim - Pay vend value	In base units
63	Reserved	
64	Interim - Pay vend value (in exact change)	In base units
65	Reserved	
66	Interim - Discount value	In base units
67	Reserved	
68	Interim - Misc. vend value	In base units
69	Reserved	
70	Interim - Price line 1 vend value	In base units
71	Reserved	
72	Interim - Price line 2 vend value	In base units
73	Reserved	
74	Interim - Price line 3 vend value	In base units
75	Reserved	
76	Interim - Price line 4 vend value	In base units
77	Reserved	
78	Interim - Price line 5 vend value	In base units
79	Reserved	
80	Interim - Price line 6 vend value	In base units
81	Reserved	
82	Interim - Price line 7 vend value	In base units
83	Reserved	
84	Interim - Price line 8 vend value	In base units
85	Reserved	
86	Interim - Price line 9 vend value	In base units
87	Reserved	
88	Interim - Price line 10 vend value	In base units
89	Reserved	
90	Interim - Event counter 1	
91	Reserved	
92	Interim - Event counter 2	
93	Reserved	
94	Interim - Event counter 3	
95	Reserved	
96	Interim - Event counter 4	
97 - 120	Spare	
121	VMC ID	
122	Coin Mechanism ID	
123	Coin Scaling Factor	

124	Security code	Reserved for internal Audit Unit use
125	Security code	Reserved for internal Audit Unit use
126	Reserved	Reserved for internal Audit Unit use
127	Reserved	Reserved for internal Audit Unit use
128 - 239	User defined addresses †	
240 - 255	Reserved	Reserved for internal Audit Unit use

† : The user defined addresses (128 to 239) are available for application specific audit items. Their exact content will depend on each application, and are outside the scope of this specification.

5.3 Audit Storage Unit Command Codes

The Executive can issue up to 16 different commands to the Audit Storage Unit. The following table shows the complete command set for the ASU peripheral.

Command #	Code	Attribute
0	0000	Not Used †
1	0001	STATUS
2	0010	DATA SYNC
3	0011	Not Used †
4	0100	Not Used †
5	0101	Not Used †
6	0110	Not Used †
7	0111	Not Used †
8	1000	Not Used †
9	1001	Not Used †
10	1010	Not Used †
11	1011	Not Used †
12	1100	Not Used †
13	1101	Not Used †
14	1110	Not Used †
15	1111	NEGATIVE ACKNOWLEDGE

† : Although these commands are not currently used, the ASU MUST reply to them with an ACK (0x00).

5.4 Detail of ASU Commands / Replies

The following sections will detail each specified command, and the expected reply from the ASU to that command. Note that peripherals must only ever reply to commands which are addressed to them.

5.4.1 PNAK Reply

As was stated in section 3.2.5, if the peripheral receives a message addressed to it with incorrect parity (or other datalink layer error), then it should respond with a PNAK (0xFF). Thus a PNAK is a valid response to all of the following commands (if they were received with an error).

5.4.2 Command 1 - Status

The STATUS command is issued by the Executive during each poll of the ASU. It may be interleaved between other commands. The format of the reply from the ASU is shown in figure 5.1.

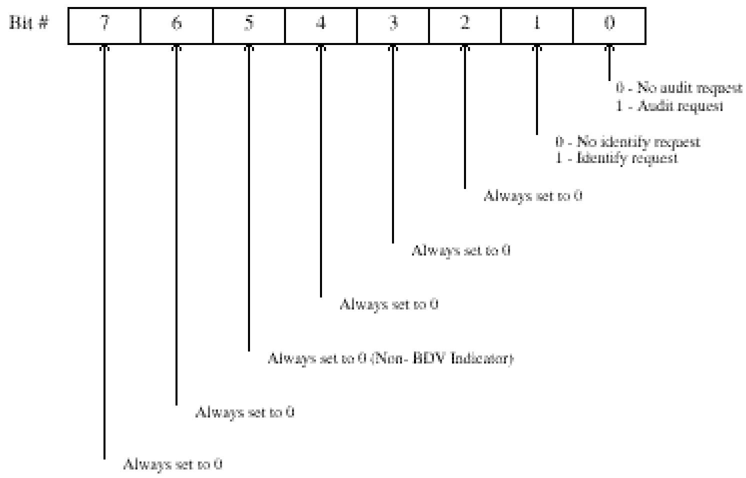


Fig 10 : ASU Reply to Status

The Audit Request flag (bit 0) will be set if the ASU requires an update of the audit data present in the system. This could occur when the ASU has been "interrogated" by a data collector (external to the serial link). If the Audit Request bit is set, the Executive will normally then send an AUDIT command to all other peripherals, and handle any audit data thus reported by the use of the SEND AUDIT ADDRESS and SEND AUDIT DATA commands. The ASU should not keep the Audit Request flag set after it has been reported to the Executive.

The Identify flag (bit 1) will be set if the ASU has just been switched on. The Executive will then issue an IDENTIFY command to each peripheral, and handle any identity data thus reported by the use of the SEND AUDIT ADDRESS and SEND AUDIT DATA commands. The ASU should not keep the Identify flag set after it has been reported to the Executive.

Note that the Executive will send over the coin scaling factor (address 123) whenever it is switched on, or if either of the above flags signal a request for an audit data update.

5.4.3 Command 2 - Data Sync

The DATA SYNC command is issued by the Executive to terminate a datablock transfer to the ASU. To transfer data to the ASU, the Executive will send 4 data nibbles (bit 4 clear for data nibble transfer) followed by the DATA SYNC command. The ASU is expected to acknowledge each of these data nibbles with an ACK (0x00) if it was received with no Datalink Layer errors, or a PNAK if a datalink Layer error (e.g. parity) occurred.

At the end of the transfer, the Executive will issue a DATA SYNC command to signal the end of transfer to the ASU.

The 4 data nibbles will always be sent in the following order :

Least significant 4 bits (bits 0 to 3) of a 8 bit address
Most significant (bits 4 to 7) of a 8 bit address
Least significant 4 bits (bits 0 to 3) of a 8 bit data value
Most significant (bits 4 to 7) of a 8 bit data value

Fig 11 details the messages sent from the Executive during a data transfer sequence. Note that the ASU is expected to ACK each message as it is received.

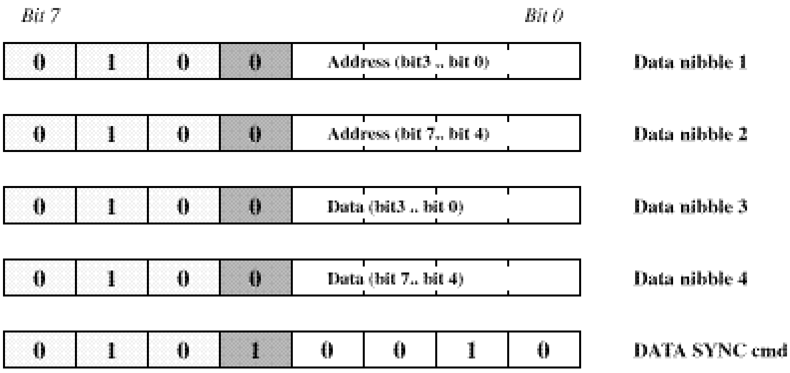


Fig 11 : Data Transfer Messages from Executive to ASU

5.4.4 Command 15 - Negative Acknowledge

The NEGATIVE ACKNOWLEDGE (NAK) command is issued by the Executive to the ASU when it detects a parity or over-run error on the ASU's reply character. On receiving this command, the ASU **MUST** repeat its last transmission. The Executive will wait with all functions inhibited until a reply is received.

5.5 Example Message Sequences

This section contains a number of example message sequences. Note that these are typical sequences for information only and do not form part of the specification. Note also that the time between messages sent from the Executive is not part of this specification, and must **NOT** be implied from these examples.

5.5.1 Polling of ASU (No Activity)

The following sequence illustrates the commands sent to the ASU and its expected reply when there is no activity in the system, and no credit is present.

Executive -> ASU	**Meaning**	**Reply**	**Meaning**
01010001b (51h)	STATUS	00000000b	ACK (Nothing to report)
01010001b (51h)	STATUS	00000000b	ACK (Nothing to report)

5.5.2 Polling of ASU (Audit Request)

The following sequence illustrates the commands sent to the ASU and its expected reply when it wishes to get updated audit information (between the 2nd & 3rd poll). Note that once the request is made, the ASU should clear the flag. In this example the Executive does not send any audit data.

Executive -> ASU	Meaning	Reply	Meaning
01010001b (51h)	STATUS	00000000b	ACK (Nothing to report)
01010001b (51h)	STATUS	00000000b	ACK (Nothing to report)
01010001b (51h)	STATUS	00000001b	Audit Request
01010001b (51h)	STATUS	00000000b	ACK (Nothing to report)

5.5.3 Polling of ASU (Identify Request)

The following sequence illustrates the commands sent to the ASU and its expected reply when it wishes to get updated identity information (between the 2nd & 3rd poll). Note that once the request is made, the ASU should clear the flag. In this example the Executive does not send any identity data.

Executive -> ASU	Meaning	Reply	Meaning
01010001b (51h)	STATUS	00000000b	ACK (Nothing to report)
01010001b (51h)	STATUS	00000000b	ACK (Nothing to report)
01010010b (51h)	STATUS	00000001b	Identify Request
01010001b (51h)	STATUS	00000000b	ACK (Nothing to report)

5.5.4 Polling of ASU (Identify Request)

The following sequence illustrates the commands sent to the ASU and its expected reply when it wishes to get updated identity information (between the 2nd & 3rd poll). Note that once the request is made, the ASU should clear the flag. In this example the Executive sends the coin scaling factor data (the value of which is 10).

Executive -> ASU	Meaning	Reply	Meaning
01010001b (51h)	STATUS	00000000b	ACK (Nothing to report)
01010001b (51h)	STATUS	00000000b	ACK (Nothing to report)
01010001b (51h)	STATUS	00000001b	Identify Request
01001011b (4Bh)	address (bits 0 - 3)	00000000b	ACK
01000111b (47h)	address (bits 4 - 7)	00000000b	ACK
01001010b (4Ah)	data (bits 0 - 3)	00000000b	ACK

01000000b (40h)	data (bits 4 - 7)	00000000b	ACK
01010010b (52h)	DATA SYNC	00000000b	ACK
01010001b (51h)	STATUS	00000000b	ACK (Nothing to report)

5.5.5 Audit Data Transfer

The following sequence illustrates the commands sent to the ASU and its expected reply when the Executive increments the "money to tubes" accumulators by 5 base units (after the 2nd poll). Note that the "money to tubes" accumulators are located at address 7 and 52.

Executive -> ASU	Meaning	Reply	Meaning
01010001b (51h)	STATUS	00000000b	ACK (Nothing to report)
01010001b (51h)	STATUS	00000000b	ACK (Nothing to report)
01000111b (47h)	address (bits 0 - 3)	00000000b	ACK
01000000b (40h)	address (bits 4 - 7)	00000000b	ACK
01000101b (45h)	data (bits 0 - 3)	00000000b	ACK
01000000b (40h)	data (bits 4 - 7)	00000000b	ACK
01010010b (52h)	DATA SYNC	00000000b	ACK
01000100b (44h)	address (bits 0 - 3)	00000000b	ACK
01000011b (43h)	address (bits 4 - 7)	00000000b	ACK
01000101b (45h)	data (bits 0 - 3)	00000000b	ACK
01000000b (40h)	data (bits 4 - 7)	00000000b	ACK
01010010b (52h)	DATA SYNC	00000000b	ACK
01010001b (51h)	STATUS	00000000b	ACK (Nothing to report)
01010001b (51h)	STATUS	00000000b	ACK (Nothing to report)

5.5.6 Negative Acknowledge

The following sequence illustrates the NEGATIVE ACKNOWLEDGE command when the Executive detects an error in the ASU's reply (to the 2nd STATUS command).

Executive -> VMC	Meaning	Reply	Meaning
01010001b (51h)	STATUS	00000000b	ACK (Nothing to report)
01010001b (51h)	STATUS	00000001b *	corrupt Audit Request
01011111b (5Fh)	NEGATIVE ACKNOWLEDGE	00000001b	Re-send of last reply
01010001b (51h)	STATUS	00000000b	ACK (Nothing to report)

6 Cashless Payment Peripheral Application Layer

6.1 General

The Cashless Payment Peripheral allows the use of cashless payment (e.g. magnetic cards) in the transaction system. This peripheral handles all the low level reading and writing the actual cashless media, while allowing the Executive control over the system level aspects.

6.2 Card Payment System Command Codes

The Executive can issue up to 16 different commands to the Cashless Payment Peripheral (CPP). The following table shows the complete command set for a card payment peripheral.

Command #	Code	Attribute
0	0000	Not Used †
1	0001	STATUS
2	0010	READ CARD
3	0011	SEND DATA
4	0100	ACCEPT DATA
5	0101	DECREMENT
6	0110	RE-INSTATE
7	0111	RETURN CARD
8	1000	DATA SYNC
9	1001	AUDIT
10	1010	SEND ADDRESS (Audit data)
11	1011	SEND DATA (Audit data)
12	1100	IDENTIFY
13	1101	ACCEPT DISPLAY DATA
14	1110	Not Used †
15	1111	NEGATIVE ACKNOWLEDGE

† : Although these commands are not currently used, the CPP MUST reply to them with an ACK (0x00).

6.3 Detail of CPP Commands / Replies

The following sections will detail each specified command, and the expected reply from the CPP to that command. Note that peripherals must only ever reply to commands which are addressed to them.

6.3.1 PNAK Reply

As was stated in section 3.2.5, if the peripheral receives a message addressed to it with incorrect parity (or other datalink layer error), then it should respond with a PNAK (0xFF). Thus a PNAK is a valid response to all of the following commands (if they were received with an error).

6.3.2 Command 1 - Status

The STATUS command is issued by the Executive during each poll of the CPP. It may be interleaved between other commands. The format of the reply is shown in figure 12 :

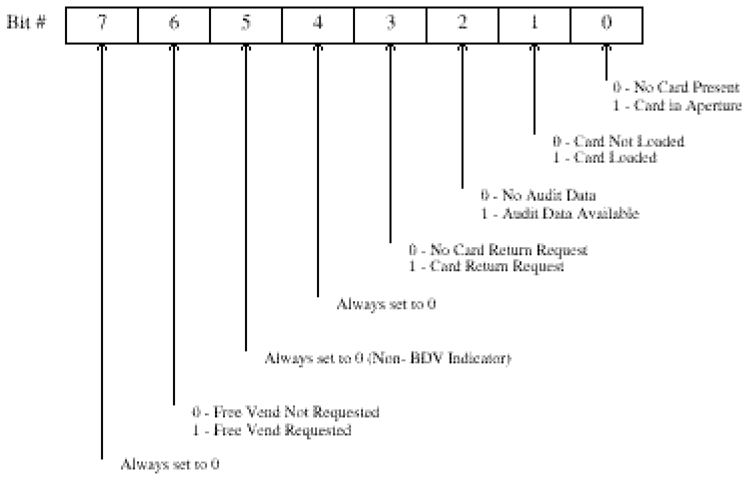


Fig 12 : Reply to Status

The Free Vend Request bit (bit 6) will be set if the CPP is allowing a free vend, and is determined when the card is first loaded. The Executive can then authorise a vend to be made without performing any credit deduction.

The Card Return Request bit is set if the user presses the card return switch on the CPP. If a card is present, the Executive would normally issue a CARD RETURN command.

The Audit Data Available bit is set if the CPP has audit information to report. The Executive will then usually request the data (via the SEND AUDIT ADDRESS, SEND AUDIT DATA commands). Note that the Executive will recheck the status of this flag after each data pair transfer.

The Card Loaded bit is set if a card has been loaded by the card system.

The Card in Aperture bit is set is a card is present in the CPP but has not yet been loaded. The Executive would normally send a READ CARD command to effect the loading.

6.3.3 Command 2 - Read Card

The READ CARD command is issued by the Executive to instruct the CPP to read and lock the card. It is issued when the reply to the STATUS command indicates that a card is present in the aperture (bit 0 set).

The reply from the CPP will indicate to the Executive if the loading was successful or not. The allowed replies are :

- 0 decimal - Load Successful
- 251 decimal - Load Failed

6.3.4 Command 3 - Send Data

This command is used to transfer the card credit data from the CPP to the Executive. This will usually be done after the card has been successfully loaded. The card credit is a 16 bit binary number and is sent over the link by 4 data nibble transfers. A checksum byte is also sent, making a total of 6 nibble transfers.

The CPP reply to each SEND DATA command is shown below :

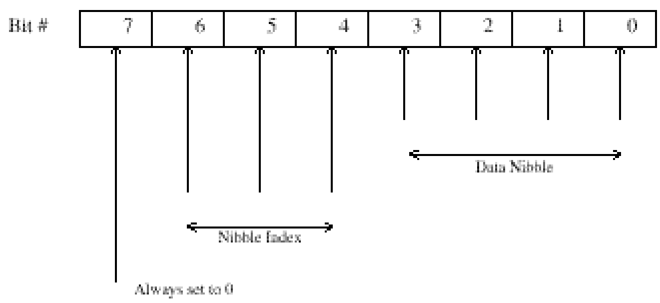


Fig 13 : Reply to Send Data

The CPP increments the Nibble Index modulo 6 for each SEND DATA command. The Executive may request the data block more than once if errors are detected.

The data block is made up as follows :

Nibble Index	Data
0	Binary real-money credit nibble 0 (lsn)
1	Binary real-money credit nibble 1
2	Binary real-money credit nibble 2
3	Binary real-moeny credit nibble 3 (msn)
4	Checksum nibble 0
5	Checksum nibble 1

The checksum for the data block is calculated such that the binary sum of the two credit bytes and the checksum byte is zero (see figure 6.3)

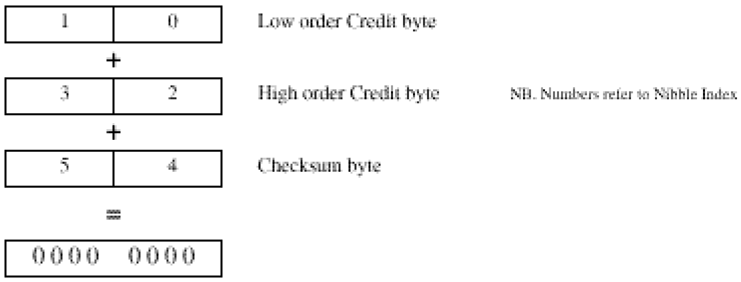


Fig 14 : Checksum Calculation

Note that if the card credit exceeds two bytes (i.e. 65,535 real money units), the CPP will send over a credit value of 65,535 to the Executive. The CPP will store the full credit value.

6.3.5 Command 4 - Accept Data

The ACCEPT DATA command allows the Executive to send a block of data (6 nibbles) over the link to the CPP. This data is a 16 bit real-money value. The reply to a the ACCEPT DATA command is a simple ACK (0x00).

To transfer data to the CPP, the Executive will issue the ACCEPT DATA command, and if the CPP reply is OK, it will send a number of data nibbles to the CPP (bit 4 clear for data nibble transfer). The CPP is expected to acknowledge each of these data nibbles with an ACK (0x00) if it was received with no datalink layer errors, or a PNAK if a datalink layer error (e.g. parity) occurred.

At the end of the transfer, the Executive will issue a DATA SYNC command to signal the end of transfer to the CPP.

The 6 data nibbles will always be sent in the following order :

Least significant 4 bits (bits 0 to 3) of a 16 bit real-money value
Next 4 bits (bits 4 to 7) of a 16 bit real-money value
Next 4 bits (bits 8 to 11) of a 16 bit real-money value
Most significant (bits 12 to 15) of a 16 bit real-money value
Least significant 4 bits (bits 0 to 3) of a 8 bit checksum
Most significant (bits 4 to 7) of a 8 bit checksum

Fig 6.4 details the messages sent from the Executive during a data transfer sequence. Note that the CPP is expected to ACK each message as it is received.

Bit 7							*Bit 0*	
0	1	1	1	0	1	0	0	ACCEPT DATA cmd
0	1	1	0	Real money value (bit 3 .. bit 0)				Data nibble 1
0	1	1	0	Real moneyalue (bit7 .. bit 4)				Data nibble 2
0	1	1	0	Real money value (bit11 .. bit 8)				Data nibble 3
0	1	1	0	Real money value (bit15 .. bit 12)				Data nibble 4
0	1	1	0	Checksum (bit3 .. bit 0)				Data nibble 5
0	1	1	0	Checksum (bit 7.. bit 4)				Data nibble 6
0	1	1	1	1	0	0	0	DATA SYNC cmd

Fig 15 : Data Transfer Messages from Executive to CCP

This 16 bit real-money value will be held by the CPP until it is "told" what to do with it. The action taken by the CPP will depend on which of the following commands is received after the data block is received :

- DECREMENT - The transferred value will be decremented from the card credit, and the data block cleared.
- RETURN CARD - The transferred value will be added to the card credit and the data block cleared. The card will be returned with the new credit value.
- ACCEPT DATA - The transferred value is cleared in preparation for receiving a new data block. The card credit remains unchanged.

6.3.6 Command 5 - Decrement

The DECREMENT command causes the CPP to deduct the 16 bit real-money value sent by the Executive (see above) from the card credit. The value data block is cleared. The reply to the DECREMENT command is either :

- 0 decimal - Deduction OK
- 251 decimal - Deduction failed

6.3.7 Command 6 - Reinstate

The REINSTATE command instructs the CPP to re-instate the card data to the state before the last DECREMENT command was received. This would normally be used if the vend failed due to some vending machine malfunction. The reply to the REINSTATE command is either :

- 0 decimal - Reinstate OK
- 251 decimal - Reinstate failed

6.3.8 Command 7 - Return Card

The RETURN CARD command instructs the CPP to re-write and return the card. The value data block value previously received should be added to the card credit before the card is written. The data block value is then cleared. This allows the Executive to re-value cards.

If the addition of the datablock to the card credit causes the card credit to exceed the maximum, a Return Failed reply (251 decimal) must be sent to the Executive, the card credit is left unchanged and the card is **NOT** returned. The Executive should ensure this case does not occur.

If the CPP does not allow re-valuation, the card may be returned, and a reply of 0 decimal sent to the Executive.

The Executive should assume the CPP has accepted the data value to be summed to the card credit providing the response to the RETURN CARD command indicates Re-valuation successful, regardless of whether the card was returned or not. If the response indicates that re-valuation failed, the Executive should retain the re-valuation credit value.

The reply to the RETURN CARD command is either :

- 0 decimal - Re-valuation failed, Card return OK
- 1 decimal - Re-valuation OK, Card return OK
- 251 decimal - Re-valuation failed, Card return failed
- 252 decimal - Re-valuation OK, Card return failed

6.3.9 Command 8 - Data Sync

The DATA SYNC command is issued by the Executive to terminate a data transfer to the CPP. Note that there are two types of data transfer to the CPP :

- Card credit via ACCEPT DATA
- Display information via ACCEPT DISPLAY DATA

In both cases the CPP will verify the checksum of the data block received, and send back a reply indicating the outcome of this check :

- 0 decimal - Checksum OK
- 251 decimal - Checksum failed
- 254 decimal - Lost Sync

The Checksum Failed reply will be returned if the checksum of the datablock sent to the CPP was incorrect. The Executive will repeat the transfer attempt.

The Lost Sync reply will be returned if the CPP was still expecting further data (i.e. synchronization error). The Executive will repeat the transfer attempt.

6.3.10 Command 9 - Audit

The AUDIT command instructs the CPP to report how much (if any) audit data it currently has stored. The Executive will normally issue the AUDIT command when there is an audit request on the system. This audit request may come from the Audit Storage Unit peripheral (see chapter 5) or from another source. The AUDIT command allows the Executive to get the most up-to-date audit data from the CPP.

The reply from the CPP is a binary number in the range 0 to 250 indicating the number of Audit Byte Pairs of static data which the CPP has to report. See section 6.3.11 for details of how this audit information is transferred to the Executive. Note that most CPPs will report audit data "as it happens", and in such cases the reply to the AUDIT command may well be 0x00 (no data byte pairs).

6.3.11 Command 10 - Send Audit Address

The SEND AUDIT ADDRESS command is issued by the Executive if the CPP indicates it has audit data byte pairs to report (see sections 6.3.2, 6.3.10, 6.3.13). The Executive will send a SEND AUDIT ADDRESS command followed by a SEND AUDIT DATA for each Audit Byte Pair the CPP indicates it has.

All audit data transfer on the Protocol A link is via "Audit Byte Pairs". An Audit Byte Pair consists of 2 characters (each in the range 0 to 250). The first character specifies an address of an audit data register in the Audit Storage Unit peripheral. The second character specifies the data that is to be "loaded" in the specified register. Note that the term "loaded" does not imply a simple overwrite of any data already stored in the Audit Storage Unit. See chapter 5 for details of how the Audit Storage Unit operates, and how data sent to its registers is handled.

The allowed replies to the SEND AUDIT ADDRESS command are :

- 0 - 250 decimal - Address to audit the data to
- 254 decimal - Sync Lost (SEND AUDIT DATA expected)

The "sync lost" reply is sent if the CPP was not expecting the audit command just received. This mis-match could occur due to message loss due to noise on the link, or CPP power loss.

Note than a CPP does not need to report audit data if the application in which it is used does not require this.

6.3.12 Command 11 - Send Audit Data

The SEND AUDIT DATA command is issued by the Executive (after the SEND AUDIT ADDRESS) if the CPP indicates it has audit data to report. The Executive sends a SEND AUDIT ADDRESS command followed by a SEND AUDIT DATA for each Audit Byte Pair the CPP indicates it has.

The allowed replies to the SEND AUDIT DATA command are :

- 0 - 250 decimal - Audit data byte
- 254 decimal - Sync Lost (SEND AUDIT ADD. expected)

6.3.13 Command 12 - Identify

The IDENTIFY command can be issued by the Executive after power-up to request the CPP’s identification data.

The reply from the CPP is a binary number in the range 0 to 250 indicating the number of Audit Byte Pairs of identification data which the CPP has to report. See section 6.3.11 for details of how this audit information is transferred to the Executive.

6.3.14 Command 13 - Accept Display Data

The ACCEPT DISPLAY DATA command is issued by the Executive to transfer the current non-card credit data block to the CPP for display purposes only. The non-card credit is transferred in the same way as the credit value (see section 6.3.5).

To transfer data to the CPP, the Executive will issue the ACCEPT DISPLAY DATA command, and if the CPP reply is OK, it will send a number of data nibbles to the CPP (bit 4 clear for data nibble transfer). The CPP is expected to acknowledge each of these data nibbles with an ACK (0x00) if it was received with no datalink layer errors, or a PNAK if a datalink layer error (e.g. parity) occurred.

At the end of the transfer, the Executive will issue a DATA SYNC command to signal the end of transfer to the CPP.

The 6 data nibbles will always be sent in the following order :

Least significant 4 bits (bits 0 to 3) of a 16 bit base unit value
Next 4 bits (bits 4 to 7) of a 16 bit base unit value
Next 4 bits (bits 8 to 11) of a 16 bit base unit value
Most significant (bits 12 to 15) of a 16 bit base unit value
Least significant 4 bits (bits 0 to 3) of a 8 bit scaling factor
Most significant (bits 4 to 7) of a 8 bit scaling factor

On receipt of the DATA SYNC command the CPP should convert the non-card base unit value received into "real-money" (by multiplying it by the scaling factor). The CPP should then add the card credit value it is currently holding to the non-card credit value calculated, and display the resulting total (true system credit). It is not necessary for the decimal point position to be transferred from the Executive as the CPP will have this programmed into it.

The reply to the ACCEPT DISPLAY DATA command is either :

- 0 decimal - Failed - will not accept data
- 128 decimal - OK - waiting for data to follow

6.3.15 Command 15 - Negative Acknowledge

The NEGATIVE ACKNOWLEDGE (NAK) command is issued by the Executive to the CPP when it detects a parity or over-run error on the CPP's reply character. On receiving this command, the CPP **MUST** repeat its last transmission. The Executive will wait with all functions inhibited until a reply is received.

6.4 Example Message Sequences

This section contains a number of example message sequences. Note that these are typical sequences for information only and do not form part of the specification. Note also that the time between messages sent from the Executive is not part of this specification, and must **NOT** be implied from these examples.

6.4.1 Polling of CPP (No Activity)

The following sequence illustrates the commands sent to the CPP and its expected reply when there is no activity in the system, and no card is present.

Executive -> CPP	Meaning	Reply	Meaning
01110001b (71h)	STATUS	00000000b	ACK (Nothing to report)
01110001b (71h)	STATUS	00000000b	ACK (Nothing to report)

6.4.2 Reading of Card

The following sequence illustrates the commands sent to the CPP and its expected reply when a card is inserted into the aperture (after the 2nd poll) and is read.

Executive -> CPP	Meaning	Reply	Meaning
01110001b (71h)	STATUS	00000000b	ACK (Nothing to report)
01110001b (71h)	STATUS	00000000b	ACK (Nothing to report)
01110001b (71h)	STATUS	00000001b	Card in Aperture
01110010b (72h)	READ CARD	00000000b	Read OK
01110001b (71h)	STATUS	00000010b	Card Loaded
01110001b (71h)	STATUS	00000010b	Card Loaded

6.4.3 Reading of Card (free vend)

The following sequence illustrates the commands sent to the CPP and its expected reply when a free vend card is inserted into the aperture (after the 2nd poll) and is read.

Executive -> CPP	Meaning	Reply	Meaning
01110001b (71h)	STATUS	00000000b	ACK (Nothing to report)
01110001b (71h)	STATUS	00000000b	ACK (Nothing to report)
01110001b (71h)	STATUS	00000001b	Card in Aperture

01110010b (72h)	READ CARD	00000000b	Read OK
01110001b (71h)	STATUS	01000010b	Card Loaded - Free vend req

6.4.4 Reading of Card Data

The following sequence illustrates the commands sent to the CPP and its expected reply when a card is loaded, and its value of 10 real money units is read by the Executive.

Executive -> CPP	Meaning	Reply	Meaning
01110001b (71h)	STATUS	00000000b	ACK (Nothing to report)
01110001b (71h)	STATUS	00000000b	ACK (Nothing to report)
01110001b (71h)	STATUS	00000001b	Card in Aperture
01110010b (72h)	READ CARD	00000000b	Read OK
01110001b (71h)	STATUS	00000010b	Card Loaded
01110011b (73h)	SEND DATA	00001010b	nibble 0
01110011b (73h)	SEND DATA	00010000b	nibble 1
01110011b (73h)	SEND DATA	00100000b	nibble 2
01110011b (73h)	SEND DATA	00110000b	nibble 3
01110011b (73h)	SEND DATA	01000110b	nibble 4
01110011b (73h)	SEND DATA	01011111b	nibble 5
01110001b (71h)	STATUS	00000010b	Card Loaded

6.4.5 Reading of Card Data (with parity error)

The following sequence illustrates the commands sent to the CPP and its expected reply when a card is loaded, and its value of 10 real money units is read by the Executive. In this example, the Executive detects nibble 0 as corrupt.

Executive -> CPP	Meaning	Reply	Meaning
01110001b (71h)	STATUS	00000000b	ACK (Nothing to report)
01110001b (71h)	STATUS	00000000b	ACK (Nothing to report)
01110001b (71h)	STATUS	00000001b	Card in Aperture
01110010b (72h)	READ CARD	00000000b	Read OK
01110001b (71h)	STATUS	00000010b	Card Loaded
01110011b (73h)	SEND DATA	00001010b	nibble 0 * Rx'ed corrupt*
01111111b (7Fh)	NAK	00001010b	Repeat of nibble 0
01110011b (73h)	SEND DATA	00010000b	nibble 1
01110011b (73h)	SEND DATA	00100000b	nibble 2
01110011b (73h)	SEND DATA	00110000b	nibble 3
01110011b (73h)	SEND DATA	01000110b	nibble 4
01110011b (73h)	SEND DATA	01011111b	nibble 5
01110001b (71h)	STATUS	00000010b	Card Loaded

6.4.6 Reading of Card Data (with checksum error)

The following sequence illustrates the commands sent to the CPP and its expected reply when a card is loaded, and its value of 10 real money units is read by the Executive. In this example, the Executive detects a checksum error on the block, due to corruption in nibble 4.

Executive -> CPP	Meaning	Reply	Meaning
01110001b (71h)	STATUS	00000000b	ACK (Nothing to report)
01110001b (71h)	STATUS	00000000b	ACK (Nothing to report)
01110001b (71h)	STATUS	00000001b	Card in Aperture
01110010b (72h)	READ CARD	00000000b	Read OK
01110001b (71h)	STATUS	00000010b	Card Loaded
01110011b (73h)	SEND DATA	00001010b	nibble 0
01110011b (73h)	SEND DATA	00010000b	nibble 1
01110011b (73h)	SEND DATA	00100000b	nibble 2
01110011b (73h)	SEND DATA	00110000b	nibble 3
01110011b (73h)	SEND DATA	01000011b	nibble 4 *corrupt *
01110011b (73h)	SEND DATA	01011111b	nibble 5
01110011b (73h)	SEND DATA	00001010b	nibble 0
01110011b (73h)	SEND DATA	00010000b	nibble 1
01110011b (73h)	SEND DATA	00100000b	nibble 2
01110011b (73h)	SEND DATA	00110000b	nibble 3
01110011b (73h)	SEND DATA	01000110b	nibble 4
01110011b (73h)	SEND DATA	01011111b	nibble 5
01110001b (71h)	STATUS	00000010b	Card Loaded

6.4.7 Transfer of Display Data

The following sequence illustrates the commands sent to the CPP and its expected reply when the Executive wishes to display a total credit of (card credit + 50 real money units)

Executive -> CPP	Meaning	Reply	Meaning
01110001b (71h)	STATUS	00000010b	Card Loaded
01111101b (7Dh)	ACCEPT DISPLAY DATA	10000000b	Ready to Receive
01101010b (6Ah)	value (10 base units)	00000000b	ACK
01100000b (60h)	value	00000000b	ACK
01100000b (60h)	value	00000000b	ACK
01100000b (60h)	value	00000000b	ACK
01100101b (65h)	scaling factor	00000000b	ACK
01100000b (60h)	scaling factor	00000000b	ACK
01111000b (78h)	DATA SYNC	00000000b	ACK
01110001b (71h)	STATUS	00000010b	Card Loaded

6.4.8 Card Return - No Revalidation

The following sequence illustrates the commands sent to the CPP and its expected reply when there is a card loaded and the user requests its return (after the 2nd poll). Note that in this example, the Executive does not wish to add any value to the card, and the CPP does not support card revaluation.

Executive -> CPP	Meaning	Reply	Meaning
01110001b (71h)	STATUS	00000010b	Card Loaded
01110001b (71h)	STATUS	00000010b	Card Loaded
01110001b (71h)	STATUS	00001010b	Card Return Request
01110100b (74h)	ACCEPT DATA	00000000b	ACK
01100000b (60h)	value	00000000b	ACK
01100000b (60h)	value	00000000b	ACK
01100000b (60h)	value	00000000b	ACK
01100000b (60h)	value	00000000b	ACK
01100000b (60h)	checksum LSN	00000000b	ACK
01100000b (60h)	checksum MSN	00000000b	ACK
01111000b (78h)	DATA SYNC	00000000b	ACK
01110111b (77h)	RETURN CARD	00000000b	Card Returned - no reval.
01110001b (71h)	STATUS	00000000b	ACK (Nothing to report)

6.4.9 Card Return - Revalidation

The following sequence illustrates the commands sent to the CPP and its expected reply when there is a card loaded and the user requests its return (after the 2nd poll). The example shows the card being revalued by 10 real money units.

Executive -> CPP	Meaning	Reply	Meaning
01110001b (71h)	STATUS	00000010b	Card Loaded
01110001b (71h)	STATUS	00000010b	Card Loaded
01110001b (71h)	STATUS	00001010b	Card Return Request
01110100b (74h)	ACCEPT DATA	00000000b	ACK
01101010b (6Ah)	value	00000000b	ACK
01100000b (60h)	value	00000000b	ACK
01100000b (60h)	value	00000000b	ACK
01100000b (60h)	value	00000000b	ACK
01100110b (66h)	checksum LSN	00000000b	ACK
01101111b (6Fh)	checksum MSN	00000000b	ACK
01111000b (78h)	DATA SYNC	00000000b	ACK
01110111b (77h)	RETURN CARD	00000001b	Card Returned OK
01110001b (71h)	STATUS	00000000b	ACK (Nothing to report)

6.4.10 Card Decrement & Return

The following sequence illustrates the commands sent to the CPP and its expected reply when there is a card loaded and the user requests its return (after the 2nd poll). The example shows the card being decremented by 10 real money units.

Executive -> CPP	Meaning	Reply	Meaning
01110001b (71h)	STATUS	00000010b	Card Loaded
01110001b (71h)	STATUS	00000010b	Card Loaded
01110001b (71h)	STATUS	00001010b	Card Return Request
01110100b (74h)	ACCEPT DATA	00000000b	ACK
01101010b (6Ah)	value	00000000b	ACK
01100000b (60h)	value	00000000b	ACK
01100000b (60h)	value	00000000b	ACK
01100000b (60h)	value	00000000b	ACK
01100110b (66h)	checksum LSN	00000000b	ACK
01101111b (6Fh)	checksum MSN	00000000b	ACK
01111000b (78h)	DATA SYNC	00000000b	ACK
01110101b (75h)	DECREMENT	00000000b	ACK
01110100b (74h)	ACCEPT DATA	00000000b	ACK
01100000b (60h)	value	00000000b	ACK
01100000b (60h)	value	00000000b	ACK
01100000b (60h)	value	00000000b	ACK
01100000b (60h)	value	00000000b	ACK
01100000b (60h)	checksum LSN	00000000b	ACK
01100000b (60h)	checksum MSN	00000000b	ACK
01111000b (78h)	DATA SYNC	00000000b	ACK
01110111b (77h)	RETURN CARD	00000001b	Card Returned OK
01110001b (71h)	STATUS	00000000b	ACK (Nothing to report)

6.4.11 Card Decrement & Reinstate

The following sequence illustrates the commands sent to the CPP and its expected reply when a card is decremented by 10 real money units, then this is re-instated before the card's return

Executive -> CPP	Meaning	Reply	Meaning
01110001b (71h)	STATUS	00000010b	Card Loaded
01110001b (71h)	STATUS	00000010b	Card Loaded
01110100b (74h)	ACCEPT DATA	00000000b	ACK
01101010b (6Ah)	value	00000000b	ACK
01100000b (60h)	value	00000000b	ACK
01100000b (60h)	value	00000000b	ACK
01100000b (60h)	value	00000000b	ACK
01100110b (66h)	checksum LSN	00000000b	ACK
01101111b (6Fh)	checksum MSN	00000000b	ACK
01111000b (78h)	DATA SYNC	00000000b	ACK
01110101b (75h)	DECREMENT	00000000b	ACK
01110001b (71h)	STATUS	00000010b	Card Loaded
01110001b (71h)	STATUS	00000010b	Card Loaded
01110110b (76h)	REINSTATE	00000000b	Reinstate OK
01110001b (71h)	STATUS	00000010b	Card Loaded
01110100b (74h)	ACCEPT DATA	00000000b	ACK
01100000b (60h)	value	00000000b	ACK
01100000b (60h)	value	00000000b	ACK
01100000b (60h)	value	00000000b	ACK
01100000b (60h)	value	00000000b	ACK
01100000b (60h)	checksum LSN	00000000b	ACK
01100000b (60h)	checksum MSN	00000000b	ACK
01111000b (78h)	DATA SYNC	00000000b	ACK
01110111b (77h)	RETURN CARD	00000001b	Card Returned OK
01110001b (71h)	STATUS	00000000b	ACK (Nothing to report)

6.4.12 Audit Request - No Data

The following sequence illustrates the AUDIT command when the CPP has no static audit data to report.

Executive -> CPP	Meaning	Reply	Meaning
01110001b (71h)	STATUS	00000000b	ACK (Nothing to report)
01110001b (71h)	STATUS	00000000b	ACK (Nothing to report)
01111001b (79h)	AUDIT	00000000b	ACK (No audit data byte pairs)
01110001b (71h)	STATUS	00000000b	ACK (Nothing to report)
01110001b (71h)	STATUS	00000000b	ACK (Nothing to report)

6.4.13 Audit Request - Data

The following sequence illustrates the AUDIT command when the CPP has 2 audit byte pairs to report.

Executive -> CPP	Meaning	Reply	Meaning
01111001b (79h)	AUDIT	00000010b	2 audit byte pairs
01111010b (7Ah)	SEND AUDIT ADDRESS	00010111b	1st address = 23
01111011b (7Bh)	SEND AUDIT DATA	00000010b	Data for loc 23 = 2
01111010b (7Ah)	SEND AUDIT ADDRESS	01000100b	2nd address = 68
01111011b (7Bh)	SEND AUDIT DATA	00000010b	Data for loc 68 = 2
01110001b (71h)	STATUS	00000000b	ACK (Nothing to report)
01110001b (71h)	STATUS	00000000b	ACK (Nothing to report)

6.4.14 Identify Request - Data

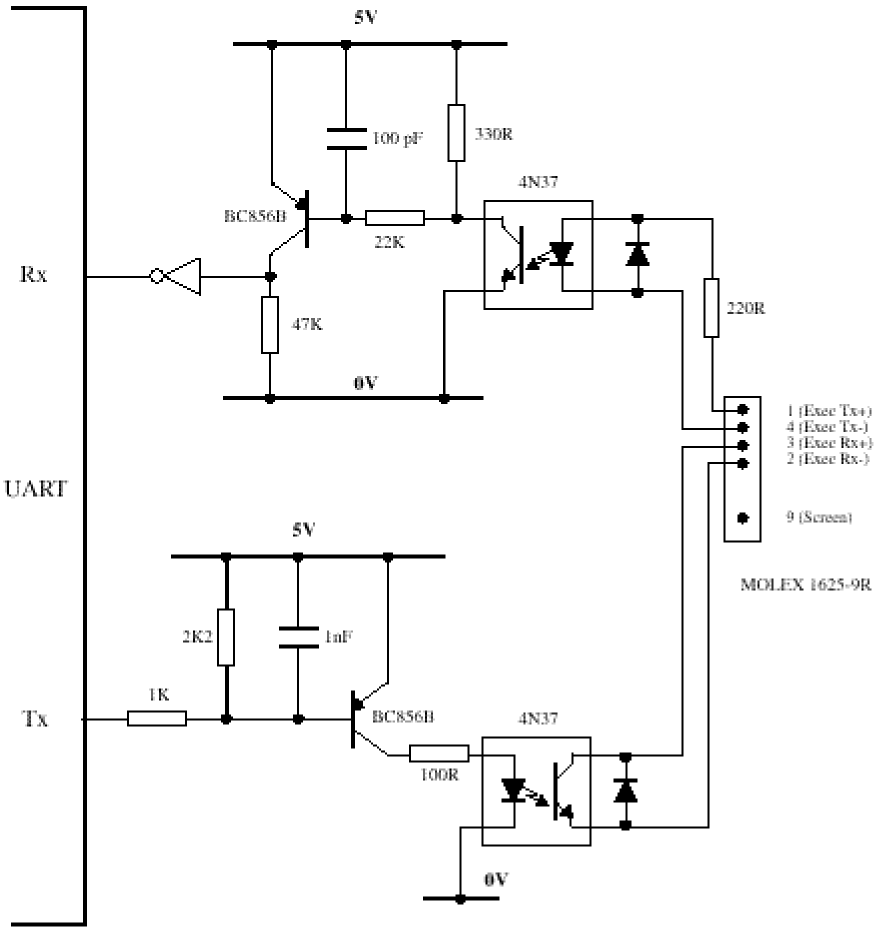
The following sequence illustrates the IDENTIFY command when the CPP has 1 audit byte pair to report.

Executive -> CPP	Meaning	Reply	Meaning
01111100b (7Ch)	IDENTIFY	00000001b	1 audit byte pair
01111010b (7Ah)	SEND AUDIT ADDRESS	10000000b	1st address = 128
01111011b (7Bh)	SEND AUDIT DATA	00101010b	Data for loc 128 = 42
01110001b (71h)	STATUS	00000000b	ACK (Nothing to report)
01110001b (71h)	STATUS	00000000b	ACK (Nothing to report)

APPENDIX: A Recommended Hardware Drive

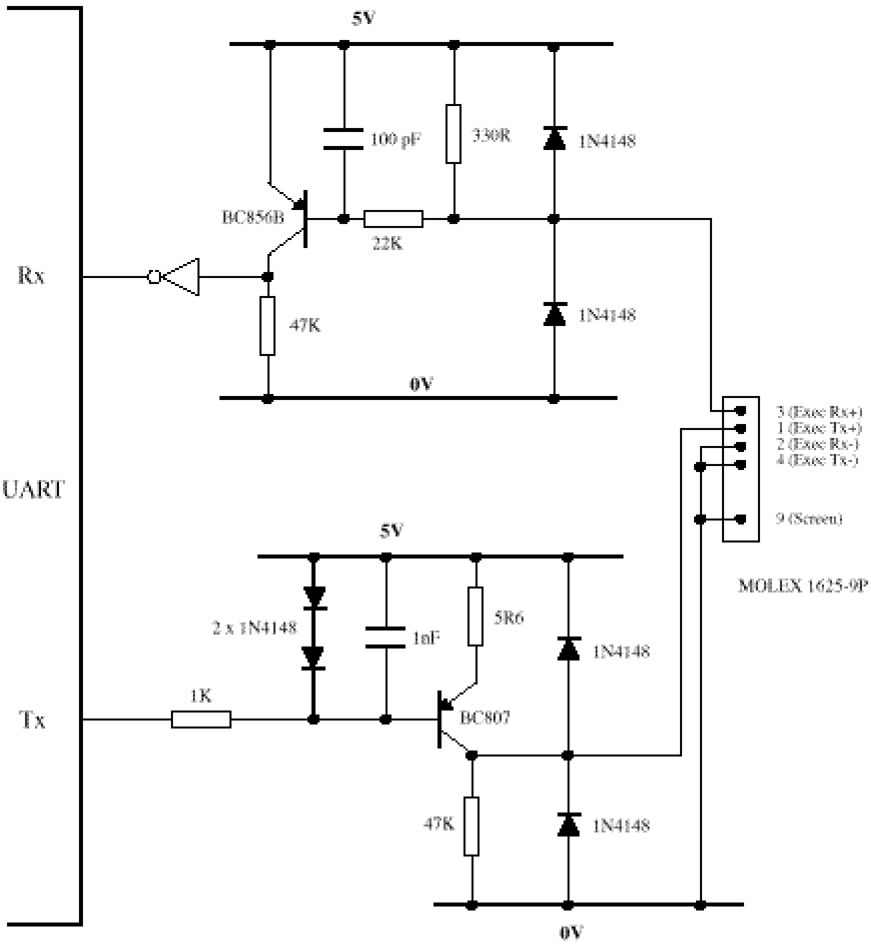
A.1 Peripheral Link Interface

The following circuit is recommended for all Protocol A peripherals:



A.2 Executive Link Interface

The following circuit is recommended for the Protocol A Executive:



APPENDIX: B CashFlow® 560 Implementation

General

This appendix details a number of aspects of the Protocol A implementation on the MEI CashFlow® 560 Executive Changer. These are given only to assist understanding of a Protocol A system based around the CashFlow® 560 Changer, and **DO NOT** form any part of the specification of the protocol.

MEI reserves the right to modify any of the following parameters without prior notice.

ASupported Peripherals

The CashFlow® 560 Executive Changer only supports the VMC peripheral. It does not offer Protocol A based audit or cashless.

The CashFlow® 560 BDV Changer supports VMC, audit and cashless peripherals. These peripherals can be either standard Protocol A devices or BDV001 devices.

Reply Timeouts

The CashFlow® 560 Executive Changer implements the following reply timeouts for each command sent to the VMC. Should a the reply not be received within this time, the Changer will inhibit coin acceptance and start its error handling actions.

Command	**Reply time-out**
STATUS	30 s
CREDIT	4 s
VEND	180 s
AUDIT †	-
SEND AUDIT ADDRESS	4 s
SEND AUDIT DATA	4 s
IDENTIFY †	-
ACCEPT DATA	4 s
DATA SYNC	4 s
NAK	Depends on the command previously transmitted.
Data Nibbles	4 s

The above times have a tolerance of -1.0s, +0.0s

†As the 560 Euro product does not support Protocol A audit, these commands are not used.

Initialisation

The first STATUS command will be sent to the VMC approximately 2.5 seconds after power up. Until Protocol A communications are successfully established, the Changer will treat the VMC as "failed". As such, coin acceptance will be inhibited and a host error will be indicated. Note that the VMC should not and need not assume STATUS command timing.

Polling

During normal operation, the Changer will poll the VMC with STATUS commands at approximately 80 ms intervals. This interval may be extended if other Changer operations are taking place (e.g. coin acceptance or dispensing).

Special Features

The CashFlow® 560 Executive Changer supports the following features :

- Price Holding
- Price Display
- Vend line tokens

The basic 560 Changer can hold up to 4 prices. If a CashFlow® Audit FEM is fitted, then this can be extended to 100 prices.

If Price Display operation is enabled, then the Changer will send over the price of the selected vend to the VMC's credit display if the vend cannot be granted (e.g. there is insufficient credit).

Vend tokens allow a free vend on either a specified price line or on any price line. For these to operate correctly, the Changer must be set to use Price Holding operation.

Audit Support

The CashFlow® 560 Executive Changer does not support a Protocol A ASU peripheral (audit is internal on an optional module). However, to ensure correct operation of the VMC, the Changer will "flush out" any audit data that the VMC indicates it has available. This data is discarded by the Changer. The AUDIT and IDENTIFY commands are not used on the 560 Executive Changer.

Error Handling

If a valid reply is not received to a command within the specified timeout period (see section B.3), the current command sequence will be aborted and the protocol A "reply time-out" error flag will be set. The Changer will inhibit coin acceptance and a host error will be indicated.

If the Changer receives a PNAK reply, or a reply byte with incorrect parity, then it will increment a "bad reply" counter and then re-send a command. The command sent depends on which invalid reply was received :

- PNAK - The Changer will re-transmit the command sent
- parity error - The Changer will send a NAK

Once the number of bad replies received for a command has reached 5, the transmission of that command (and / or command sequence) will be taken as failed and it will be aborted. The protocol A transmission retry error flag will be set. The Changer will inhibit coin acceptance and a host error will be indicated.

Note that should a reply time-out occur with the "bad reply" counter greater than 0 but less than 5, it will be the "reply time-out" error flag which will be set.

If either error flag is set, the Changer will inhibit all coin acceptance, and start to poll the VMC with the STATUS command. A shorter than normal timeout (2 seconds) will be used to allow the Changer to quickly detect if the failure has been rectified. While the error persists, the Changer will indicate this as a "host error" by turning off the red LED.

Once the VMC responds to the STATUS command, the Changer will re-enable coin acceptance and revert back to normal operation automatically.