SECTION 5

ACCESS.bus

Text Device Protocol Specification

TABLE OF CONTENTS

5.0 Introduction

5.0.1 Design Objectives

The text device protocol described in this specification is intended to provide a simple way to transmit character or binary data to and from stream oriented devices such as a bar code reader, or character display. The sequential character stream model also serves as a common denominator for connecting RS-232 interfaced devices.

5.0.2 Text Device Overview

A generic text device transmits a stream of 8-bit bytes from a character set. Simple control messages are defined to support flow control and to select communication parameters that might be used to interface with a modem. The capabilities string contains information that identifies the specific character set and communication parameters used.

5.1 Text Device Protocol

Text characters to be transmitted are sent using the Device Data Stream message (see ACCESS.bus, Description and Protocol Specification).

```
Format:

    ddddddd0     (destination address)
    sssssss0     (source address)
    0LLLLLLL     (P=0, LLLLLLL=body-length)
    |
    body         (0-127 bytes)
    |
    cccccccc     (checksum)
```

Characters are assumed to be from the coded character set specified in the Capabilities Information unless otherwise agreed upon between the application and device. This agreement might be established using ISO standardized control functions, but this is defined at a higher level than the text device protocol.

5.1.2 Device Number and Identification

A text device may be interactive like a bar code reader that transmits a stream of characters in response to user action, or non-interactive like an RS-232 converter attached to a printer. Non-interactive devices are required to provide a fixed device number since they cannot be distinguished by the order in which they are used.

A general purpose ACCESS.bus to RS-232 adaptor may have no way of reporting what is attached to its RS-232 port. In this case, the host system or application must rely on the RS-232 device for device identification. Information about what is attached to an RS-232 adaptor can be maintained on the host and associated by the device number if desired.

5.1.3 Message Length and Timing

Interactive text devices must not occupy the bus as bus master for more than eight milliseconds at a time, and must wait at least fifty microseconds (50µs) between message transmissions. Eight milliseconds (8ms) limits the maximum packet size to eighty characters or less at l00Kbps.

Non-interactive devices must not occupy the bus as bus master for more than five milliseconds at a time, and must wait at least twelve milliseconds (12ms) between message transmissions. Five milliseconds (5ms) limits the maximum packet size to fifty characters or less at l00Kbps.

The Text Device Protocol defines a Record Separator command to group message packets into identifiable records.

Application Record Separator ( )

Op-code: 30
Data: none

The Record Separator is sent by a device or the host computer to indicate the next data byte transmitted will be the first byte of a new record. Can also be used as a "end of record" command.

5.1.4 Flow Control

When a computer or device is capable of sending data faster than its bus partner can receive and process it, flow control may be needed to avoid losing data. In its simplest form, flow control allows the receiver to tell the sender to wait until it is ready before sending more data. While the I2C hardware provides low level flow control by stretching clock signals, this will not be appropriate for many applications because it blocks all traffic on the bus.

An example application requiring higher level flow control is a printer that cannot print data as fast as the host can send it. Depending on the application, the printer may need to instruct the computer to not transmit any more messages until it has room in its buffer to hold them.

Two application commands are defined for this purpose:

Application Hold ( )

Op-code: 13h Data: none.

Tell the sender to hold transmission until requested to resume.

Application Resume (count)

Op-code: llh Data: 16-bit unsigned integer count of number of bytes device is ready to accept.

Tell sender it may resume transmitting data up to 'count' characters. If 'count' is zero or omitted, the sender may continue transmitting until requested to hold.

By using a non-zero count in the AppResume command, a receiving device need not transmit an urgent message to the sender when its buffer is full.

Notes:

1. AppHold and AppResume only affect device data stream messages. All ControVStatus commands must continue to be processed regardless of the held state.

2. When a device is reset it has no way of knowing the previous flow control state. After being reset and the configuration process is complete, a device using flow control shall transmit a single AppResume command and clear its transmit held state (if any).

5.1.5 Serial Asynchronous Communication Parameters

The Text Device Protocol defines the following commands for interfacing with common UARTs.

Application Set Format (controlmask)

OpCode:	01
Data:	16-bit controlmask

The control mask specifies UART parameters as follows.
Choices listed in parenthesis are selected in order (0, 1, 2, . . .).

b0-b3	transmit speed (default, 19200, 9600, 4800, 2400, 1200, 600, 300)
b4-b7	receive speed (rx=tx, 19200, 9600, 4800, 2400, 1200, 600, 300)
b8	word size (8-bits, 7-bits)
b9	stop bits (one stop bit, two stop bits)
bl0-bll	parity (none, even, odd)
bl2	local echo (FDX, HDX)
bl3-bl4	flow control (none, XON/XOFF, DSR/DTR, RTS/CTS)

Application Request Format ( )

Op-code:	02
Data:	none

Application Request Format is sent from the host to a device and requests the device respond with an Application Report Format message.

Application Report Format (controlmask)

Op-code:	03
Data:	16-bit control mask, same as Application Set Format

5.1.6 Serial Asynchronous Control Signals

The Text Device Protocol defines the following commands for interfacing with common RS232 control signals.

Application Set Control (controlmask)

Op-code:	04
Data:	16-bit control mask

The control mask identifies the control signals to be set as follows (DCE side):

b0	DSR
b1	DTR
b2	RTS
b3	CTS
b4	RLSD (or CD)
b5	FE (framing error)
b6	OE (overrun error)
b7	PE (parity error)
b8	Break detected
b9	Send break
b10-11	reserved
b12-15	available for application use

Application Reset Control (controlmask)

Op-code:	05
Data:	16-bit control mask.

The control number identifies the control signals to be reset using the same defintions as Application Set Control.

Application Request Control ( )

Op-code:	06
Data:	none

Application Request Control is sent from the host to a device and requests the device respond with an Application Report Control message.

Application Report Control (controlmask)

Op-code:	07
Data:	16-bit control mask

Sent by a device to the host when an input control signal changes or in response to a request. The controlmask reports the status of UART control signals using the same definitions as Application Set Control.

5.1.7 Direction Control

The Text Device Protocol defines the following command for controlling the direction of bi-modal interfaces.

Application Set Direction (controlmask)

Op-code: 08
Data: 16-bit control mask

Sent by the host computer to a device to specify the direction of interface operation.

b0-b1 Direction (transmit, receive, full duplex)
b2-b3 Rank (Default, Master, Slave, Off-line)
  Examples of Master: RS-232 DTE
    Host centronics port
    Dialing FAX
  Examples of Slave: RS-232 DCE
    Printer centronics port
    Answering FAX
b4-b7 Reserved
b8-b15 Available for application use

5.2 Capabilities Information

The keywords defined in this section have standard meanings within the ACCESS.bus Generic Text Device Protocol.

Keyword	Meaning
Charset()	Tags the default character set used to encode text data. The following names are defined: ASCII, ISO:8859/1
Direction() Direction(input output HDX FDX)	Tags whether device can be used for data input, data output, or half-duplex, or full-duplex.
FlowControl() FlowControl(input output)	Tags whether low level flow control is used for input and/or output (AppHold and AppResume).

UARTformat()	Lists common UART format selections that are supported and can be modified or examined by Application Format commands.

UARTformat(speed, size, parity, stop, echo, flow(0 1 2 3))

The sublist of numbers, if present, indicates which selections are supported.

UARTcontrol()	Lists common modem control signals that are supported and can be modified or examined by Application Control commands.

UARTcontrol(DSR DTR RTS CTS RLSD FE OE PE BREAK)
