

# Media Technologies

Windows™ Brick-2 Videowall Control Software

## Serial Communications Protocol

### FRAME STRUCTURE

The Serial RS-232 communications frame structure is common to both directions of transmission. Messages are delimited by start and end of message flags. The body of the frame comprises the control and address bytes the message (which is optional) and a single byte checksum.

SOM	Control	Address	Message data...	Checksum	EOM
-----	---------	---------	-----------------	----------	-----

Frames not containing any message bytes are referred to as 'null' frames.

Characters received between EOM and SOM characters (i.e. un-framed) are ignored - unless the single character sequence file feature is invoked.

The start and end of messages are special character values that must not occur in the data bytes between them to ensure correct message framing. To achieve this a third value, the byte stuff character, (BSF) is used. This is also a unique character value that also must not appear as a data byte.

The byte values of these three special characters are:

SOM	0x7E
EOM	0x7D
BSF	0x7C

Byte transparency is achieved by replacing any of the three special characters that occur as data bytes with a two byte sequence headed by the BSF character. The second character is the character X-ORed with the 'smudge' character, 0x20, which prevents recognition by the receiver as a special byte. e.g. a data byte 0x7E is replaced with the byte pair: 0x7C,0x5E.

### CONTROL BYTE

The control byte comprises two fields. The upper 4 bits are mode bits controlling the use of the addressing information. The lower 4 bits are the Device Address. In the receive direction (Processor to PC) the mode bits are not used. (Unused bits are marked in the table below with a dash) As a convention, unused bits should be set to zero, but receiving software should not reject messages if they are not.

Transmit Bits							
7	6	5	4	3	2	1	0
M1	M0	-	0	A3	A2	A1	A0

Receive Bits							
7	6	5	4	3	2	1	0
-	-	-	1	A3	A2	A1	A0

Device Address [ A3 to A0 ] Device Type

Device Address	Device Type
0 - 6	Reserved
7	Brick Processor

The two mode bits M1 and M0 allow for 4 addressing modes:

M1	M0	Addressing Mode
0	0	Normal Address
0	1	Response (Poll)
1	0	Global of type (A3 to A0)
1	1	Global all units.

## ADDRESS BYTE

The address byte allows 256 devices of the type selected in the Control byte Device Address field to be individually addressed.

## NORMAL ADDRESS MODE

In this mode, the address byte is used to select a specific device (one of 256), of the type set by the device address field in the control byte. Only one unique device will therefore act on the message.

## RESPONSE (POLL) MODE

The message will be received by the addressed device in the same manner as the Normal Address mode, however the addressed device will respond with a return frame. If there is no message pending in the addressed device the response will be a null frame. Note this will be the reply from a previous message. If the device receives a POLL it will send the next message in the outgoing queue before processing the message. A null frame can be used to read the next message from the send queue without executing a command.

## GLOBAL TYPE ADDRESS MODE

All devices of the type defined in the control bytes unit field will act upon this message. It is not possible to globally poll devices since more than one would respond and the messages collide. In this mode the address byte is redundant, but it is still sent to maintain a simple frame structure. By convention the address byte should be zero in this case.

## GLOBAL ADDRESS MODE

This is similar to the previous case except that all devices of any type will act on the message. The usefulness of this mode is really restricted to generic commands such as 'RESET' and 'BAUDRATE'

## CHECKSUM

The checksum is calculated by summing modulo-256, all of the bytes between the start and end of message characters, prior to the byte stuffing process, and subtracting this figure from zero. The receiver checks the received data by summing all the bytes following the start of message (after expanding the 'stuffed' bytes). When the end of message character is received the sum should be zero.

## MESSAGE CONTENTS

The length of the message is not limited by the protocol. As described above, it can be of zero length for use in polling and null responses, where there is otherwise no data to transfer. The message format is the same for both directions of transmission, comprising a function code in the range 0 to 255 followed by an optional parameter list. The convention for parameters greater than 255 is least significant byte first.

Function codes are device specific and detailed in separate manuals for each device. The function codes from 0 to 15 are reserved for generic operations, such as returning software version numbers, which all devices support. In addition there are generic function codes to support extended addressing techniques for selecting multiple devices.

## PROTOCOL VIOLATIONS

Any protocol violations should cause the current message to be abandoned. The receiver will revert to a mode waiting for the start of message character. If the violation was an out of sequence start of message character, a new message will be assumed. Message frames with checksum errors are ignored.

Although the BSF character would normally only precede a 'smudged' version of one of the 3 special byte codes, (SOM, EOM, BSF) the presence of any other character is not considered a violation. The 'un-smudging' process should be applied regardless of the 'smudged' character code provided it is not either SOM, EOM or BSF.