Communication and Data Retrieval Specification

Smart-Control Box v4.4C Firmware

Blackstone Global Inc.
10 Kidder Road, Unit 4
Chelmsford, MA. 01824
PH: 978-455-0672
www.blkcorp.com

1 Smart Box Communication Specification

1.1.1 Overview

On the hardware level the Smart Box (SMB) is connected to a host, typically a PC, using USB. SMB circuitry contains a FTDI USB to serial converter chip. When SMB is connected to PC and the proper drivers are installed a new virtual RS-232 port is created. Any software application can then open this serial port and communicate with SMB the same way as with any other serial device. Software application does not require any low level interaction on USB level.

1.1.2 Driver Installation

Connect smart box to computer with USB cable and power up the smart box from the tool or external source. If USB drivers are not present on the PC the first time the smart box is connected the Windows will display the prompt to install the drivers. Follow the prompt and install drivers from Drivers subfolder.

Once the driver is installed it is then required to determine which communication port number is assigned to the SMB. The best location to find the defined port number is in Device Manager. For Windows XP: click Start button, move cursor above My computer, right click and select Manage – this will open the Computer Management window. Click Device Manager and expand the ports section. Newly created serial port will be typically on the end of list, you can check its properties to confirm that this is an FTDI port.

For Windows 7, click Start button, then Devices and Printers. At the bottom of the list you should find the FT232R USB listed, right mouse click to select properties to find the assigned comm port.

1.1.3 Request Format

Host will need to open the serial port identified in previous step and send messages with following format to SMB. Serial port settings are 19200 bits per second, 8 data bits, 1 start bit, 1 stop bit, no parity, no handshaking.

Requests sent from the host to SMB board have following format:

length, sequence number, command, data (optional) …, checksum

length – 1 byte: number of all bytes sent from host including length byte and checksum

sequence number – 1 byte: number generated by host used to verify that the response received was the response expected for the last command

command – 1 byte: requested command

data – 0 to 251 bytes: optional request data

checksum – 1 byte: message checksum – see checksum calculation section

The shortest request message is 4 bytes long.

1.1.4 Response format

SMB will respond to requests with the response in format:

length, sequence number, status, data (optional) …, checksum

length – 1 byte: number of all bytes sent from the SMB including length byte and checksum

sequence number – 1 byte: the same number as the sequence number in command request

status – 1 byte: status of requested command – see status codes section

data – 0 to 251 bytes: optional response data

checksum – 1 byte: message checksum - see checksum calculation section

The SMB will typically respond in less than 50 milliseconds.

1.1.5 Checksum Calculation

Check sum is calculated as:

1. calculate arithmetic sum of all bytes including length byte and last byte in optional data
2. temp1 = sum & 0xff, where & is bitwise and operation
3. if temp1 = 0 then checksum = 0, else checksum = 256 – temp1

SMB uses the same checksum calculation for the response message. To confirm if response from SMB is valid, add all received bytes including checksum then perform bitwise AND operation with mask 0xff. Message is valid if result is 0.

1.1.6 Status Codes

Third byte in response is status code. Value 0 means success. Any non-zero value means an error.

1.1.7 Host Commands

Third byte in request sent by application is command. Following commands are supported:

```
CMD_GET_STATUS1 = 0x59, no data
```

If SMB receives valid request it will respond with 13 byte response. All multi-byte values are sent LSB first.

Field	Zero based byte index in response where field starts	Description
PX	3-4	X coordinate of current tool position in millimeters from Set-Origin point with offset 2048. Subtract 2048 from returned value to get position in mm.
PY	5-6	Y coordinate of current tool position in millimeters from Set-Origin point with offset 2048. Subtract 2048 from returned value to get position in mm.
Seq. Number	7	Used when sequence mode is active. Index of current point in sequence (zero based). Value 0xff means invalid value.
Active Point	8	Zero based index of recipe point if all conditions for that point are met. Returned value of 0xff if some conditions are not met (for example arm is above taught point but expected tool bit is not installed on tool)
Recipe	9-10	Current recipe index (0 based). Value 0 means recipe 1.
Torque	11	Zero based index of current torque selected by smart box. Value 0xfe and 0xff means that the torque controller is disabled by smart box (some condition is not met, e.g. arm is not on taught position or correct bit is not installed)